UNIVERSITÉ DE FRANCHE-COMTÉ

ECOLE DOCTORALE «LANGAGES, ESPACES, TEMPS, SOCIETES»

Thèse en vue de l'obtention du tire de docteur en

Sciences du langage

MODÉLISATION INFORMATIQUE DE STRUCTURES DYNAMIQUES DE SEGMENTS TEXTUELS POUR L'ANALYSE DE CORPUS

Présentée et soutenue publiquement par

François DAOUST

Le 10 janvier 2011

Sous la direction de M. le Professeur Jean-Marie VIPREY
et la codirection de M. le Professeur Yves MARCOUX

Membres du jury :

Lou BURNARD, université d'Oxford, Rapporteur
Jules DUCHASTEL, Professeur à l'université du Québec à Montréal
Yves MARCOUX, Professeur à l'université de Montréal
André SALEM, Professeur à l'université de Paris III, Rapporteur
Jean-Marie VIPREY, Professeur à l'université de Franche-Comté

Sommaire

Sommaire....2

Remerciements....3

1 Présentation....5

2 Quels modèles de calcul pour quelles analyses textuelles?....12

3 SATO : un modèle informatique pour la construction de dispositifs expérimentaux....54

4 Trente ans de développement et d'utilisation du logiciel SATO 114

5 Le modèle d'implantation de SATO....241

6 L'annotation structurelle....289

7 Modèles informatiques pour l'exploitation de la structure formelle et des segments dynamiques....348

8 Conclusion....386

Index....389

Table des matières....396

Remerciements

La rédaction d'une thèse en fin de carrière a l'avantage de pouvoir bénéficier de multiples collaborations de recherche sur des dizaines d'années. Le fruit de mes propres travaux et réflexions est donc en partie redevable à tous ces collaborateurs dont plusieurs se retrouvent cités dans la thèse à titre de cosignataires d'articles et de communications scientifiques.

Mais avant de signaler ces multiples contributions, je me dois de remercier ceux qui m'ont accompagné dans cette recherche doctorale. Mes remerciements iront donc d'abord à mon directeur de recherche, Jean-Marie Viprey, dont la confiance et la sollicitude ont été admirables tout au long de ces années. Sans son soutien, je n'aurais sans doute jamais mené mon projet à terme. Je remercie aussi Yves Marcoux dont l'apport à la thèse, particulièrement sur la problématique informatique, a été très précieux. Je veux aussi remercier André Salem dont l'appui intellectuel et amical, à l'occasion de nos multiples échanges au cours des années, a nourri mon travail de réflexion et de formalisation. C'est d'ailleurs André Salem qui m'a mis en contact avec Lou Burnard qui me fait l'honneur de participer au jury de la thèse. Pilier de la *Text Encoding Initiative*, Lou Burnard m'a apporté un point de vue éclairant sur l'édition numérique et sur l'annotation.

On comprendra, à la lecture de la thèse, que je puisse avoir des remerciements particuliers pour Jules Duchastel à qui je dois d'exister, en tant que chercheur en analyse de texte par ordinateur. Sans lui, en effet, mon travail de programmation et de réflexion serait resté une *activité de loisir* à la marge d'un travail de soutien au service informatique de notre université. Non seulement Jules Duchastel a reconnu l'intérêt de mes travaux, mais en m'offrant de participer au développement du Centre ATO de l'UQAM, il m'a permis d'en faire un métier. Plus encore, c'est à un véritable travail de collaboration qu'il me conviait. Cette collaboration, entamée dès le début des années 1980, se poursuit toujours après avoir marqué chacun des travaux dont je rends compte dans la thèse.

Je voudrais aussi remercier Dimitri Della Faille avec qui j'ai beaucoup collaboré ces dernières années, notamment pour la production des interfaces Web à SATO et pour la rédaction de documents de formation. Ses remarques et observations ont certainement contribué à faciliter l'utilisation des logiciels.

Je ne sais comment remercier toutes ces personnes avec qui j'ai travaillé au cours de ma carrière. Je ne peux les nommer toutes. Je me contenterai donc ici de quelques noms associés à diverses époques. Il y a eu l'équipe des pionniers, à laquelle j'ai participé alors que j'étudiais en mathématiques à l'UQAM. L'influence de professeurs comme Lorne Bouchard et Simon Curry en informatique, ainsi que Jean-Guy Meunier en philosophie, a été le point de départ de mes choix de recherche. Ce sera ensuite autour de l'équipe de recherche dirigée par Jules Duchastel et Gilles Bourque, avec Jacques Beauchemin et Victor Armony, que je développerai mes travaux. On doit aussi rappeler une période très importante de recherche-action avec des collaborateurs gouvernementaux. Ces recherches m'ont permis de prendre connaissance de problèmes concrets d'analyse textuelle dans les milieux de travail. Dans ces équipes, il y avait des universitaires : Louis-Claude Paquin, Luc Dupuy, Gracia Pagola, Suzanne Bertrand-Gastaldy, Claire Gélinas-Chebat, entre autres. Il y avait aussi des professionnels du gouvernement : Richard Parent, Léo Laroche, Lise Ouellette, Yves Rochon, Maurice Gingras...

J'ai pu profiter de l'apport d'équipes de recherche dans diverses disciplines, ce qui m'a permis de confronter mes modèles informatiques à des problématiques diverses. Parmi ces équipes, il faut mentionner une collaboration de longue date avec des linguistes qui s'intéressent tout particulièrement à l'analyse de corpus dans une perspective d'évolution de la syntaxe : Fernande Dupuis, Monique Dufresne et Monique Lemieux.

Beaucoup de personnes ont gravité autour des projets mentionnés et d'autres encore. C'est à toutes ces personnes que je voudrais adresser mes remerciements car elles ont contribué, chacune à leur manière, aux réalisations dont je fais part dans la thèse.

1 Présentation

Cette thèse a la particularité de s'inscrire au terme d'une trentaine d'années de développement d'outils et de méthodes en analyse de texte assistée par ordinateur. Elle sera l'occasion de tracer les grandes lignes de cette évolution afin d'inscrire nos choix théoriques et nos stratégies informatiques dans leur contexte historique. Mais, elle vise surtout à actualiser ces choix et ces stratégies dans la mouvance des dernières années qui tente d'exprimer en XML les formalismes de description et d'analyse du langage et du texte. Plus encore, elle vise à s'appuyer sur la continuité et l'actualisation de notre démarche pour évaluer la possibilité d'étendre nos modèles vers cet espace qui se déploie entre le lexique et les contextes, contextes le plus souvent limités aux frontières de la phrase. Cet espace, encore faiblement exploré du point de vue informatique, concerne les phénomènes de la *textualité*, et du discours matérialisé dans le corpus de recherche, et qui se manifestent à travers une multitude de structures embrassant de larges empans textuels, voire l'ensemble du corpus.



Résumé (1a, remarque)

La linguistique textuelle conçoit le texte comme un tout structuré et non comme un simple assemblage de propositions indépendantes. En plus des aspects de connexité intra et inter phrastique, le texte est le lieu de relations non linéaires de cohésion et de cohérence dont l'interprétation appartient au lecteur. L'analyse de discours propose une approche systématique et transparente de *reconstruction du sens* s'appuyant sur l'explicitation du fonctionnement textuel dans le contexte d'une production discursive inscrite dans l'histoire. Les procédures informatisées facilitent grandement cette explicitation basée sur l'analyse de grands corpus.

Interactivité, transparence, sensibilité aux dimensions linguistiques, déploiement de protocoles diversifiés en vue de découvrir les stratégies discursives, voilà quelques-unes des exigences posées au modèle informatique qui doit soutenir le travail de l'analyste. Les procédures informatiques les plus connues font surtout appel à l'analyse lexicale appuyée, dans certains cas, de procédures d'analyse syntaxique.

Cependant, peu de dispositifs permettent de manipuler les structures textuelles au-delà de la phrase.

L'objectif de la thèse est de proposer un modèle informatique pour représenter, construire et exploiter des structures textuelles. Le modèle proposé s'appuie sur une représentation du texte sous la forme d'un plan lexique/occurrences augmenté de systèmes d'annotations lexicales et contextuelles, modèle dont une implantation a été réalisée dans le logiciel SATO dont on présentera les fonctionnalités et l'organisation interne. La présentation d'un certain nombre de travaux rendront compte du développement et de l'utilisation du logiciel dans divers contextes.

La prise en charge formelle des structures textuelles et discursives trouve un allié dans le langage de balisage XML et dans les propositions de la *Text Encoding Initiative* (TEI). Formellement, les structures construites sur les segments textuels correspondent à des graphes. Dans le contexte d'une analyse textuelle en élaboration, ces graphes seront multiples et partiellement déployés. La *résolution* de ces graphes, au sens du rattachement des nœuds à des segments textuels ou à des nœuds d'autres graphes, est un processus dynamique qui pourra être soutenu par divers mécanismes informatiques. Des exemples tirés de la linguistique textuelle serviront à illustrer les principes de l'annotation structurelle. Des considérations prospectives sur une implantation informatique d'un système de gestion de l'annotation structurelle seront aussi exposées.

Notre projet de recherche s'appuie sur l'identification d'un besoin apparu au cours de nos travaux de développement d'outils informatiques pour l'analyse de texte dans une perspective d'analyse de discours. L'objectif du projet est de formuler une proposition de modèle informatique qui, s'appuyant sur le modèle implanté dans le logiciel SATO, fournirait un cadre fonctionnel pour représenter les relations qui structurent les divers segments d'un texte, d'un corpus de textes et du *genre* dans lequel il s'inscrit.

SATO (Système d'analyse de texte par ordinateur), dans ses versions 3 et 4 (Daoust, 1996, bibliographie du chapitre 2), est un logiciel destiné à supporter une variété de stratégies d'analyse textuelle. Il repose sur une reconfiguration du texte linéaire (chaine de caractères)

sous la forme d'un plan lexique/occurrences. Présenté en termes informels, l'axe lexical peut être vu comme la liste des mots utilisés dans le texte alors que l'axe des occurrences représente l'ordonnancement de ces unités lexicales suivant l'ordre naturel du texte (de gauche à droite et de bas en haut pour les langues latines).

L'émergence de la dimension lexicale du texte dans le plan lexique/occurrences permettra de distinguer la catégorisation hors contexte, qui appartient au lexique de la langue, du domaine et du corpus, de la dimension contextuelle, qui appartient davantage à l'énoncé.

Dans SATO, les systèmes de catégorisation sont appelés *propriétés*. Ces systèmes d'annotation peuvent être vus, d'un point de vue informatique, comme des champs venant décrire les formes lexicales ou les occurrences.

En termes généraux, on entend par *segment textuel* une suite continue de mots. Il peut s'agir d'un chapitre, d'une phrase, d'une tirade, d'un syntagme, etc. Dans le SATO actuel, il n'y a pas de modèle général de représentation et de traitement des segments. Les segments sont généralement construits à la volée pour les besoins du calcul, à partir des propriétés touchant les occurrences elles-mêmes.

En fait, on peut considérer qu'une occurrence (un mot dans le texte) correspond à un segment *dégénéré*. Aussi, lorsque plusieurs mots consécutifs partagent une même valeur de propriété, on peut procéder à une *mise en évidence*, au sens algébrique, faisant ainsi apparaitre une suite de mots comme une entité autonome. Ainsi, dans sa représentation du texte, SATO ne connait pas le segment en tant qu'objet primitif et pérenne. En ce sens, ce type de segment, que l'on pourrait qualifier de catégoriel, est un artifice d'annotation. C'est un segment virtuel en ce sens qu'il ne se matérialise pas dans une structure informatique explicite.

En plus des segments catégoriels, SATO manipule des segments calculés, c'est-à-dire qui résultent d'un dispositif (algorithme) de segmentation. Ces segments construits n'ont pas de pérennité dans la représentation SATO du texte. Ce sont davantage des dispositifs de lecture, par exemple une concordance ou une partition du texte en paragraphes. En fait, seule la référence de pagination (propriété *page*) possède une existence autonome et permanente en tant que segment. Mais cette structure est ad hoc et singulière.

L'absence d'un modèle général des segments textuels a pour conséquence qu'il est difficile de représenter la macro-structure d'un document au sens d'un modèle hiérarchique de type

SGML ou XML. Plus encore, la construction d'un texte fait appel à des relations sémantiques, stylistiques, narratives, argumentatives, etc. qui vont bien au-delà de la représentation hiérarchique propre aux documents structurés. Ces besoins variés de représentation des structures textuelles requièrent donc que le segment puisse avoir une existence autonome avec ses propres règles.

C'est sur cette base qu'en novembre 1999 nous présentions nos objectifs de recherche.

> En résumé, la recherche a trois objectifs:
>
> 1- Définir l'objet segment en rapport avec les autres objets du modèle SATO;
>
> 2- Définir la classe des fonctions applicables aux segments et en évaluer le potentiel pour représenter les structures définies dans les langages de balisage et les modèles syntaxiques;
>
> 3- Proposer des hypothèses de réalisation informatique du modèle dans la foulée du modèle SATO.
>
> (Daoust, 1999, cf. Bibliographie du chapitre 2).

Si, au départ, ce qui a motivé notre entreprise doctorale est la recherche d'un modèle informatique renouvelé, il est vite apparu que, dans sa rédaction, cette entreprise devait d'abord se fonder une présentation des travaux passés. Les dernières décennies ont été traversées par de multiples influences au gré des changements de paradigme dans le domaine des *sciences du langage*, pour reprendre ce terme somme toute assez récent. Il est rare que l'on prenne le temps de revoir une carrière de recherche en la replaçant dans ses contextes historiques. Or, l'intérêt particulier de cette démarche, c'est que le point de vue qui guide notre regard dans ce retour historique est celui de l'informaticien, informaticien qui propose un modèle formel de calcul et ses diverses implantations soumises aux aléas de l'évolution des contraintes informatiques. Il faut donc insister sur ce point. Notre objectif n'est pas de faire œuvre d'historien. Il n'est pas non plus de faire un essai sur l'évolution du point de vue des sciences humaines sur la langue et le texte. L'objectif est, essentiellement, d'évaluer un modèle informatique du point de vue de son insertion dans une pratique de recherche en sciences du langage. Quelle était la pertinence de ce modèle aux différentes époques de son histoire? Jusqu'à quel point ce modèle était-il, ou pas, avant-gardiste, dans le contexte historique, et précurseur de pratiques scientifiques devenues aujourd'hui courantes? Quelles sont les insuffisances du modèle par rapport aux nouvelles questions et aux nouvelles

possibilités de l'utilisation de l'ordinateur en sciences du langage? Dans ce contexte, le modèle existant garde-t-il sa pertinence? Et, si c'est le cas, est-il possible de faire évoluer le modèle pour qu'il prenne en compte les tendances nouvelles de recherche dans le domaine? Pour répondre à ces questions, on devra, nécessairement, aborder autant les questions de méthodologie de la recherche en analyse textuelle que les questions de modélisation informatique et d'algorithmie. On ne s'étonnera donc pas du caractère multidisciplinaire de nos propos, avec le paradoxe souvent lié à ce genre de démarche qui tente d'intégrer divers points de vue disciplinaires sans pouvoir se poser en spécialiste d'aucune de ces disciplines.

La présentation et la mise à jour de nos travaux s'articulera autour de trois pôles inscrits dans leur contexte historique :

- La présentation d'un modèle ayant conduit à la réalisation d'un logiciel et à des pratiques d'analyse ayant toujours cours;
- L'actualisation informatique de ce modèle dans le contexte d'un déploiement faisant appel à l'environnement Web et à la normalisation XML;
- Une proposition d'extension du modèle pour rendre compte de l'*annotation structurelle* visant, tout particulièrement, à représenter ce tissus de relations que le lecteur analyste établit entre segments textuels dynamiquement construits au cours du processus d'analyse. C'est d'ailleurs ce projet prospectif qui justifie le titre de la thèse, même si, dans le corps de l'exposé, la présentation des travaux antérieurs occupera une place majeure.

Plus précisément, le plan de l'exposé est le suivant.

D'entrée de jeu, nous plongeons dans l'examen des paradigmes théoriques auxquels nous avons été exposés. Et nous les abordons d'un point de vue précis, nommément désigné dans le titre de notre deuxième chapitre suivant immédiatement cette courte présentation : *Quels modèles de calcul pour quelles analyses textuelles?*

Bien sûr, la question posée au chapitre deux n'est pas que théorique puisque, au cours des dernières décennies, nous avons travaillé concrètement à une certaine réponse à la question sous la forme d'un logiciel que nous présenterons au chapitre trois intitulé *SATO : un modèle informatique pour la construction de dispositifs expérimentaux*. Outre la présentation du modèle logique à la base du logiciel, ce chapitre aborde l'ergonomie de SATO et illustre

l'utilisation de SATO sur un corpus d'entrevues de jeunes sur l'usage du tabac. Plusieurs autres analyses auraient pu être utilisées pour cette illustration. Mais celle-là a l'avantage de combiner l'utilisation de SATO à celle d'autres logiciels de *textométrie*, ce qui permet de saisir la spécificité de notre approche et sa complémentarité par rapport aux approches essentiellement statistiques de la lexicométrie française. De plus, cette analyse ayant fait l'objet de deux publications aux *Journées internationales d'analyse des données textuelles* (JADT), elle jouit déjà d'une exposition publique lui conférant un certain caractère exemplaire.

À cette étape de lecture de la thèse, le lecteur aura pris connaissances du modèle SATO et du contexte théorique dans lequel il se situe. Il devient alors possible d'aborder le contexte historique de développement du logiciel et de sa mise en œuvre dans une variété de projets. Notre quatrième chapitre, intitulé *Trente ans de développement et d'utilisation du logiciel SATO*, présente, par ordre chronologique, un certain nombre de projets dans lequel nous avons été directement impliqués. On y trouvera, bien sûr, la marque historique des paradigmes de recherche dominants selon les époques. Mais, et surtout, on verra comment le logiciel a évolué au travers du temps, tout en gardant et en approfondissant sa cohérence initiale. Au terme de ce parcours historique, on sera donc mieux placé pour se projeter vers le futur afin d'entrevoir une suite possible au modèle et à son implantation informatique.

Mais, avant d'arriver à l'aspect prospectif de notre recherche, il fallait présenter SATO du point de vue organique. Ce sera l'objet de notre cinquième chapitre intitulé *Le modèle d'implantation de SATO*. Ce chapitre, de nature plus technique, permet de saisir comment le programme est construit et comment on entend le faire évoluer pour en assurer l'entretien et l'évolution.

C'est finalement au sixième chapitre que nous aborderons ce qui nous a d'abord incité à entreprendre cette recherche doctorale, à savoir l'ajout d'un nouveau modèle d'annotation permettant de dépasser les limites actuelles pour mieux rendre compte des relations multiples qui structurent le texte et le discours. Ce chapitre, intitulé *L'annotation structurelle*, présente d'abord la problématique qui motive notre proposition pour procéder ensuite à une démarche exploratoire qui s'appuie sur des exemples de structures exposés dans le traité de linguistique textuelle de Jean-Michel Adam (1990, 2005). Notre objectif ici n'est pas de prendre une posture théorique de défense des modèles linguistiques abordés, mais d'utiliser ces exemples

et ces modèles pour exposer des syntaxes concrètes visant à représenter les diverses opérations d'annotation structurelle. C'est ainsi que nous soutiendrons qu'il est possible d'utiliser les recommandations de la *Text Encoding Initiative* (TEI) pour marquer de façon formelle, et relativement intuitive, les opérations d'annotation structurelle de corpus de textes afin d'en expliciter l'organisation et le fonctionnement.

C'est finalement au septième chapitre, intitulé *Modèles informatiques pour l'exploitation de la structure formelle et des segments dynamiques*, que nous reviendrons sur les aspects plus directement informatiques du problème en explorant diverses stratégies d'implantation dans le contexte général de l'analyse de texte assistée par ordinateur, et plus particulièrement du modèle SATO. Ce chapitre prendra notamment appui sur le principe de l'annotation en couches multiples, sur les recherches autour de l'exploitation informatisée des *arbres linguistiques*, de nature syntaxique principalement, et du monde en évolution des outils XML. L'annotation structurelle, surtout dans le contexte de la linguistique textuelle, n'est pas d'abord de nature syntaxique puisqu'elle vise l'au-delà de la proposition et de la phrase. Cependant, du point de vue de l'implantation informatique, la fouille et l'entretien de bases de données riches de millions d'arbres posent des défis suggérant des solutions qui pourraient nous inspirer pour la gestion de l'annotation structurelle dans un contexte d'analyse de discours.

Bien sûr, comme nous l'aborderons en conclusion, le défi de l'implantation de notre proposition d'annotation structurelle et sa prise en charge effective par les outils d'analyse textométrique, demeure entier au terme de cette recherche. Mais, nous pensons que nos travaux tracent une voie qu'il reste pertinent d'explorer dans un contexte de continuité avec la tradition d'analyse de texte assistée par ordinateur à laquelle nous avons, espère-t-on, contribué un peu au cours des années passées.

Sur le plan de la forme, nous avons choisi, dans la rédaction du présent document, d'éviter l'utilisation d'annexes et de notes en fin de chapitre. Notre objectif est de faciliter la lecture linéaire du document, tout en indiquant la nature complémentaire de certaines parties du texte. Autant que possible, donc, les notes sont directement intégrées au flux textuel alors que les annexes sont remplacées par des encadrés qui en identifient la nature par un support visuel, reprenant ainsi une approche suggérée par notre collègue Dimitri Della Faille pour l'écriture de documents de formation à SATO. Ainsi, on trouvera des encadrés ayant le statut de définitions, d'exemples, de remarques, de notices techniques et, surtout, de publications. En

effet, comme une grande partie de ce travail est consacré à rappeler des travaux déjà réalisés en les actualisant et en en faisant le bilan, il devenait nécessaire de reproduire, au moins partiellement, un certain nombre de publications concernant ces travaux. Inévitablement, on retrouvera des redites, d'une publication à l'autre. La suppression systématique de ces redites aurait eu pour effet de rompre la logique d'exposition des articles reproduits et d'en faire perdre le contexte historique. La nature et le contexte de ces publications étant brièvement présentées en entête de ces encadrés, le lecteur pourra choisir d'abréger la lecture de ces publications déposées en appui à l'historique de nos travaux. Dans la logique de cette présentation favorisant une lecture linéaire du document, nous avons aussi choisi de produire une table des matières unique qui intègre, en les identifiant clairement, les chapitres et leurs sections avec les encadrés, tableaux et figures afférents.

Signalons, avant d'entrer dans le vif du sujet que nous utilisons généralement l'orthographe nouvelle dans le corps de la thèse (cf. http://www.gqmnf.org/). Cependant, dans la reproduction de textes déjà publiés, nous respectons l'orthographe originale.

On verra aussi, en consultant la table des matières, que nous avons choisi de mettre une section bibliographie à la suite de chacun des chapitres. Là aussi, on a l'inconvénient de certaines redites. Cependant, comme les chapitres ont des portées distinctes, les sources citées relèvent également de disciplines distinctes. Nous avons jugé préférable de maintenir la cohérence des paradigmes référés selon les sujets plutôt que de fusionner l'ensemble des sources dans une bibliographie unique.

2 Quels modèles de calcul pour quelles analyses textuelles?

Si notre objectif, au cours de toutes ces années de recherche, a été de produire un modèle informatique, bien fondé du point de vue formel et performant au niveau de son implantation, il est clair que les spécifications à la base du modèle ont été motivées par la nécessité de répondre à des besoins spécifiques manifestés par une certaine pratique d'analyse de texte. Certes, il existe une certaine autonomie entre les modèles mathématiques ou informatiques et

les besoins qui ont inspiré l'effort de modélisation. Il est bien connu qu'un programme informatique, une fois remis entre les mains des usagers, sert souvent d'autres fins que celles initialement prévues. En ce sens, la référence à l'analyse de discours comme cadre conceptuel fondant une certaine pratique de l'analyse textuelle ne définit pas de façon exclusive la pertinence du modèle informatique qui a inspiré le développement de SATO, d'autant que nous avons toujours pris soin de dégager la logique formelle de l'outil informatique des méthodes d'analyse qu'il supporte. Il reste que, pour comprendre les choix privilégiés dans la conception de l'outil, on doit en référer aux principes méthodologiques guidant l'analyse de texte telle que nous l'avons pratiquée dans notre tradition d'analyse de texte assistée par ordinateur.

Dans ce chapitre, nous reprendrons quelques-unes des idées maitresses de l'analyse de discours pour en dégager les implications en termes de modélisation informatique. Sans aucune prétention d'exhaustivité, nous aborderons aussi quelques autres courants et paradigmes de recherche qui auront influencé nos modèles.

2.1 L'analyse de discours

Dans *Discours et archive*, Guilhaumou et coll. définissent l'analyse de discours comme la « *manifestation de la langue dans la communication vivante* » (Guilhaumou et coll., 1994: 194). L'analyse de discours, telle que développée par l'école française, se situe « *aux limites de la linguistique et de l'histoire* » (Guilhaumou et coll., 1994: 195). Même si elle s'appuie sur la linguistique, la démarche de l'analyse de discours se situe d'emblée au-delà de la phrase, dans le *transphrastique.*

Maingueneau, dans *Nouvelles tendances en analyse de discours*, explique l'essor de l'école française d'analyse de discours par « la rencontre à l'intérieur d'une certaine *tradition* d'une *conjoncture intellectuelle* et d'une *pratique scolaire* » (Maingueneau, 1987: 5). La tradition dont il parle relève de la philologie. De son côté, Van Dijk (Van Dijk, 1985a) parlera de tradition rhétorique. La conjoncture est celle d'une réflexion sur l'écriture mariant la linguistique, le marxisme et la psychanalyse. La pratique scolaire relève de *l'explication de textes*.

Des articles récents, sur la notion de *philologie numérique* actualisent ce débat sur les traditions fondatrices de l'analyse de discours, sur les rapports entre texte et discours et sur la philologie à l'ère du document numérique.

> À propos de la notion de texte, notons tout d'abord que l'ambition d'une *analyse textuelle des discours* souligne bien, et définitivement, non seulement que *texte* et *discours* ne sont pas des objets du même plan, mais encore quelle est leur relation: le *texte* est un mode opératoire sur le *discours*. Ne pas parler d'*analyse de textes*, mais d'*analyse textuelle*, indique que le texte n'est pas un objet en soi, mais une *phase* vers l'objet fondamental des sciences humaines qu'est le *discours*. Le syntagme *analyse de textes* n'est peut-être qu'un résidu, au fond, de l'académisme scolaire, résidu qui a connu un paradoxal mais révélateur succès aussi bien dans le champ du structuralisme que dans celui des applications informatisées. (Viprey, J.-M. 2005:52)

Précisant davantage, Viprey indique que le « texte est un artefact au sens plein du terme » (idem p. 54) . C'est d'ailleurs cet artefact qui, en tant qu'objet matériel constitué en corpus, collection raisonnée de textes établie à des fins d'analyse, sera scruté à la loupe des méthodes informatisées de l'analyse textuelle, ou *analyse de texte* (sans s !) *assistée par ordinateur* (ATO) pour reprendre une expression qui nous est familière. Si le texte, poursuit Viprey, est bien la *mise en ordre valorisante* d'un discours, la philologie numérique, qualifiée de *nouvelle philologie* « renouvelle les pratiques du texte après une phase de relatif mépris pour ce dernier, phase peut-être inévitable dans le développement de l'analyse de discours, mais aujourd'hui révolue » (idem, p.55).

L'analyse de discours renvoie à l'analyse du langage en ce qu'elle concerne des stratégies d'interlocution de sujets occupant des positions sociales inscrites dans des conjonctures historiques. Les points de vue qui traversent l'analyse de discours sont très influencés par les disciplines auxquelles elle réfère. Cela est particulièrement vrai quand on compare les écoles française et anglo-saxonne d'analyse de discours. Ainsi, pour l'école française, le sujet individuel disparait au profit des *formations discursives* (Foucault, 1969), lieu d'une *position* socio-historique dans laquelle les énonciateurs individuels sont substituables.

Parmi les divers apports de la linguistique, ce sont les problématiques de l'énonciation et de la pragmatique qui constituent nos paradigmes de référence. Ces problématiques s'opposent à la conception du langage comme simple support pour la transmission d'informations. La

communication linguistique est le lieu d'un rapport social qui permet de construire et de modifier les relations entre les interlocuteurs, leurs énoncés et leurs référents (Maingueneau, 1987).

S'éloignant du lien privilégié avec les archives historiques, les autres courants d'analyse de discours vont insister davantage sur le point de vue sociologique. Aux États-Unis surtout, on retrouve beaucoup plus de travaux impliquant les aspects communicationnels et anthropologiques. On y trouve une grande préoccupation pour le discours populaire, les mythes, etc.

Au-delà des spécificités de l'école française, de son lien privilégié avec l'histoire, le marxisme et la psychanalyse, on trouve beaucoup de similitudes entre les écoles française et anglo-saxonne quant à l'origine du domaine : le structuralisme des années 60, le courant sémiotique de Barthes, Greimas et Todorov, et la linguistique.

Dans l'introduction de son tome 1 du *Handbook*, (*Discourse Analysis as a New Cross Discipline* 1985a), Van Dijk constate, comme Maingueneau, une certaine institutionnalisation de l'analyse de discours. Il indique que le domaine serait rendu aux premiers stades d'une *science normale* (Van Dijk, 1985a). Le nouveau courant d'analyse de discours, dont il situe l'origine autour de 1972-1974, serait aux confins de plusieurs influences :

- une réfutation des grammaires formelles hors contexte : on oppose l'analyse fonctionnelle au générativisme ;
- la découverte des *actes de parole* qui inscrivent les actes élocutoires dans un contexte d'actions sociales ;
- l'évolution du cadre lui-même de la théorie grammaticale tendant à dépasser le syntagme isolé pour aborder les questions de cohérence et de macro-structure sémantique ;
- l'intelligence artificielle (« computer simulation of language understanding »), qui permet de voir combien la connaissance du monde est nécessaire pour la compréhension, même pour la compréhension des histoires les plus simples.

Si l'on pourrait percevoir la perspective sémiotique comme une métathéorie du texte et de la lecture, l'analyse de discours tend plutôt à se constituer en discipline dans le champ des sciences sociales. Michel Pêcheux en définit ainsi l'objectif.

> L'analyse de discours ne prétend pas s'instituer en spécialiste de l'interprétation maîtrisant « le » sens des textes, mais seulement construire des procédures exposant le regard-lecteur à des *niveaux opaques à l'action stratégique d'un sujet* [...]. L'enjeu crucial est de *construire des interprétations* sans jamais les neutraliser ni dans le « n'importe quoi » d'un discours sur le discours, ni dans un espace logique stabilisé à prétention universelle. (Pêcheux, 1984: 15,17, cité par Maingueneau 1987:6).

Réinscrit dans la sphère culturelle, le texte, comme matérialisation du discours social, n'est pas considéré comme un simple reflet. Il a un rôle actif, une fonction de régulation. Le texte a un fonctionnement objectif, avec ses procédés et ses stratégies. La *forme* n'est pas qu'un simple emballage. Elle fait partie du *message*, de l'action du texte. L'analyse textuelle a donc pour but de faire émerger les systèmes sémiotiques et discursifs qui concourent à l'action du texte.

Dans *Éléments de linguistique textuelle, Théorie et pratique de l'analyse textuelle,* (Adam 1990), Jean-Michel Adam souligne que la théorie du texte s'est beaucoup développée depuis le milieu des années 60.

S'appuyant sur de nombreuses références, Adam pose d'emblée que le texte, comme objet, doit être considéré en termes multidisciplinaires. Il indique aussi que le texte est un tout et non un simple assemblage de propositions indépendantes. En plus des aspects de connexité intra et inter phrastique, le texte est le lieu de relations non linéaires de *cohésion cohérence*, selon le terme employé par Adam dans le contexte de la *perception-construction* par l'interprétant d'une macrostructure sémantique.

La linguistique informatique, fortement influencée par les modèles syntaxiques génératives, fournit des modèles de représentation et de calcul pour les unités syntagmatiques et la morphologie lexicale. À une échelle de granularité plus *macroscopique* de l'information textuelle, on retrouve des modèles synthétiques de type lexicométrique et statistique qui s'avèrent particulièrement efficaces pour dépister les « tendances lourdes » au sein de larges corpus textuels (cf. Harman et coll. 1996).

Mais, le problème de cet *entre-deux* dans l'échelle de granularité de l'analyse textuelle, territoire de significations entre le lexique et le syntagme, est encore largement ouvert. La

linguistique textuelle, dans laquelle s'inscrit Jean-Michel Adam vise à combler ce vide théorique.

En accord avec Bakhtine (Bakhtine, 1978), il souligne que l'idée que nous avons de la forme de l'énoncé (genre de discours) structure le processus discursif. Plus encore, c'est parce que nous partageons socialement une intelligence des *genres du discours* que la communication est possible. Pour comprendre les énoncés, il faut dès le début être sensible au tout discursif. Cela fait partie intégrante de la *compétence linguistique*. Comprenons que la notion de *genre* est employée dans un sens beaucoup plus large que celle de genre littéraire plus ou moins codifié. Elle renvoie plutôt à la notion de *formation discursive* de Michel Foucault (Foucault, 1969). Cela rejoint aussi tout le courant d'analyse du discours (Maingueneau, 1991) qui considère le *genre* dans sa dimension fondamentalement sociale, la perception du genre s'opérant dans un espace social d'échange entre plusieurs discours socialement déterminés. François Rastier va exactement dans le même sens mais de façon plus précise en indiquant que le genre est ce qui rattache un texte à un discours. Il ajoute : « L'origine des genres se trouve donc dans la différenciation des pratiques sociales » (Rastier, 1989:40 cité par Adam 1990:22).

Le texte est un objet de communication. Comme l'indique Adam : « Le texte ne tire son identité sémiotique-sémantique que de son inscription dans un processus de lecture » (idem p.28). Ainsi, on parlera du *scripteur énonciateur* (l'auteur du texte) et du *lecteur co-énonciateur* puisque la lecture est une reconstruction du sens par le lecteur en fonction de ses connaissances préalables, de ses attentes, de ses intentions de lecture, etc. « Plutôt que le réceptacle dépositaire d'un sens plus ou moins profond, le texte apparait comme une série de contraintes qui dessinent des parcours interprétatifs. Chaque lecteur est libre de suivre un tracé personnel, de déformer ou de négliger à sa guise les parcours indiqués par le texte, en fonction de ses objectifs et de sa situation historique » (Rastier, 1989: 18, cité p.30).

2.2 La perspective sémiotique

Faisant le lien entre les théories cognitives et sémiotiques de la lecture, Bertrand-Gastaldy et coll. indiquent que le texte est un entrelacement de multiples systèmes sémiotiques. « La lecture est un acte d'interprétation sensible à certains de ces systèmes selon le projet ou le

point de vue » (Bertrand-Gastaldy et coll., 1995:3). Cette reconnaissance est donc à la fois un acte individuel et un acte social déterminé culturellement.

Dans un article publié en 1997, Bertrand-Gastaldy approfondit cette conception sémiotique du texte en examinant ses conséquences du point de vue du lecteur et des logiciels d'analyse textuelle. Rappelant le mot de McKenzie, qui indique que *texte* et *tissage* partagent la même origine étymologique *texere* (McKenzie, 1991:32), elle poursuit en citant Weaver :

> (...) each level of language and language processing (letters, words, etc.) is a system. With language and language process, then, we have systems within systems within systems, holarchically and thus multi-directionnal interrelated (Weaver, C. 1985:313).

L'image du texte, entrelacs de fibres tissées, est amplifiée par cette idée de systèmes de systèmes de systèmes. Les unités faisant objet de structurations signifiantes ne sont pas que les segments textuels de premier niveau. Ce sont des structures de segments déjà construites selon leur logique sémiotique interne. Dans une perspective fonctionnelle, cette structuration se retrouve aussi au niveau grammatical. Comme le signale Halliday, cité par Bertrand-Gastaldy,

> The key to a functional interpretation of grammatical structure is the principle that, in general, linguistic items are multifunctional. Most of the constituents in any construction higher than a word enter into more than one structural configuration (Halliday, 1985:32).

Bertrand-Gastaldy ajoute:

> In Maier's view (Maier, 1993:63), not enough attention has been paid to the many different relations structuring texts. Following a review of the classification of relations in several disciplines - in particular, linguistics, psychology and philosophy - and basing herself on Halliday (1985), she proposes to distinguish, in addition to ideational and interpersonnal relations referring to extra-textual knowledge, relations within the text : addition, comparison, consistency and temporality, which is subdivided into numerous ramifications (Bertrand-Gastaldy, 1997).

À ces relations à l'intérieur du texte se superposent les relations d'intertextualité. Ces marques d'intertextualité se retrouvent non seulement dans les références explicites de citation, mais aussi dans le partage de paradigmes lexicaux et de structures textuelles.

Il est important de souligner que ces diverses structurations se construisent à l'infini dans l'interaction entre le lecteur et le texte : « ... the choice of the granularity of the decomposition of the semiotic units and their properties depend on reading and analytic intentions » (Bertrand-Gastaldy, 1997).

D'un point de vue logiciel, Bertrand-Gastaldy en tire une conséquence précise, à savoir la nécessité de disposer d'outils informatiques de type *boite à outils* permettant des explorations multiples.

> Like a geographical map which can represent the economic, cultural, demographic or climatic aspects of a country by several sets of symbols, in isolation or in combination, each user can be offered a personal roadmap to explore the textual space. (Bertrand-Gastaldy, 1997)

On trouvera dans Bertrand-Gastaldy et coll., (1994a) diverses illustrations, appliquées au logiciel SATO, de l'utilisation d'une boite à outils d'analyse textuelle.

2.3 L'approche documentaire

Plusieurs approches d'analyse ne semblent conférer au calcul qu'une fonction de support documentaire permettant de gérer l'accès aux divers segments textuels et aux annotations générées au cours de la lecture humaine.

La fonction documentaire classique, qui consiste à classer et à indexer documents et artefacts, peut être réinvestie pour indexer et classer des parties de textes afin de les apparier, de les comparer et de les retrouver pour appuyer un raisonnement. Au-delà de la simple prise de notes, cette version élaborée du *clipping* se présente comme une véritable méthode d'analyse. Les chercheurs en sciences humaines, qui utilisent des approches dites d'*analyse qualitative* d'inspiration américaine, ont beaucoup raffiné ces méthodes d'analyse textuelle faisant appel à une technique dite de *codage ligne à ligne*.

Expliquant la méthode de l'*analyse de contenu*, Patton décrit ce qui ressemble à une indexation par sujet se traduisant par l'étiquetage de morceaux choisis ou de l'entièreté des paragraphes des textes collectés : « Simplifying the complexity of reality into some manageable classification scheme is the first step of analysis » (Patton, 1990, p. 382). Pour limiter l'arbitraire de ce processus de codage, la méthode prescrit une variété de techniques : manuel de codage, appel à plusieurs codeurs, tests d'accord pour vérifier la reproductibilité du codage par un même codeur ou entre codeurs.

Au-delà de l'aspect documentaire et classificatoire de la méthode, ce processus de codification, appliqué selon l'approche de la *théorisation ancrée* (Glaser et Strauss, 1967), est perçu comme une méthode pour construire des théories empiriquement fondées (voir Laperrière 1997 pour une présentation du point de vue et de la procédure de codage inductif et Charmaz 2000 pour une perspective constructiviste de la théorisation ancrée). On a donc ici, à la base, une lecture humaine qui vise à comparer, regrouper, classer des segments de texte selon un modèle classificatoire a priori ou selon une méthode inductive ou une combinaison des deux. L'opération va au-delà de la démarche exploratoire. Il s'agit de fait d'un processus systématique de codification qui n'est pas sans rappeler l'opération d'indexation documentaire. D'ailleurs, un des buts avoués de ce travail est de pouvoir maitriser une masse documentaire que l'on doit lire et relire dans un processus interprétatif. Le deuxième objectif de la méthode consiste à produire une hiérarchie de codes, ce qui peut nous rappeler certains aspects du thésaurus documentaire. On se sert alors des possibilités graphiques de l'ordinateur dans ses fonctions d'*idéation*, c'est-à-dire de support à la représentation schématique de concepts et de modèles. La principale critique apportée à cette méthodologie d'analyse textuelle est qu'elle ignore le mode de fonctionnement discursif et qu'elle est complètement indifférente aux articulations proprement textuelles et linguistiques (Maingueneau, 1991).

Le débat contrastant les dimensions sociales et individuelles de l'acte d'écriture et de lecture traverse aussi les grandes revues des sciences de l'information. Dans un article récent, Hjorland se démarque d'Ingwersen en présentant sa vision socio-cognitive des sciences de l'information.

> A central point in my approach is the claim that tools, concepts, meaning, information structures, information needs, and relevance criteria are shaped in discourse communities, for example, in scientific disciplines, which are parts of

> society's division of labor. A discourse community being a community in which an ordered and bounded communication process takes place. This communication is structured by a conceptual structure, by institutional enclosure, and by governance of discourse fora (see Wagner & Wittrock, 1991). This view changes the focus of IS from individuals (or computers) to the social, cultural, and scientific world. One important implication is that the relevant cognitive structures are of a historical rather than of a physiological nature. (Hjorland, 2002:258)

Hjorland fait d'ailleurs une référence explicite à l'influence du tournant pragmatique en linguistique qu'il perçoit comme une démarcation par rapport à la conception de Chomsky.

> A new pragmatic tradition turned the classic cognitive approach upside down. Human actions and activities are here seen as the most basic entities; pragmatics consists of the rules for linguistic actions; semantics is conventionalized pragmatics; and finally, syntax adds grammatical markers to help disambiguate the meaning when the context does not suffice to do so. (Hjorland, 2002:258)

Même si on y perçoit encore une forte influence positiviste, l'analyse de texte par ordinateur est aussi présente dans les grandes revues en sciences de l'information sous le vocable de *text mining*, que nous traduirons par *forage textuel*.

Dans l'édition 2002 de l'Annual Review of Information Science and Technology, Benoît situe le domaine du *forage textuel* par rapport au concept plus général de *data mining* : « Text data mining is closer to corpus-based computational linguistics and exploratory data analysis (EDA) » (Benoît, 2002:290).

En 1999, la même revue publiait un article de Walter Trybula qui présente un état de la question sur la recherche en *text mining*. Le processus de forage textuel y est résumé en 4 phases.

1. La **collecte de l'information** consistant à rassembler les textes, à les épurer et à les organiser en fonction d'objectifs spécifiques.

2. L'**extraction** d'informations dans les textes pour l'analyse de contenu. Diverses opérations sont impliquées : identification de la langue, extraction de dispositifs langagiers (noms, groupes nominaux, abréviations, vocabulaire spécialisé, et connaissances spécifiques au domaine); analyse lexicale, syntaxique, sémantique (par la

cooccurrence par exemple). L'analyse du langage naturel est utilisée dans le pré-traitement : lemmatisation, catégorisation grammaticale, identification des structures syntaxiques.

3. Le **forage** : « Mining is the process of analysing textbases and developing methods for presenting results to the user » (Trybula, 1999:400). On utilise l'analyse des métadonnées, la construction de systèmes catégoriels et de thésaurus. Mais, c'est l'analyse en clusters qui reste l'approche la plus populaire.

   > To determine document similarity, the vocabulary of the document is analyzed in a linguistic preprocessing step. The identified terms for a document are collected in term vectors. These vectors are compared to each other. The term vector of a cluster is a merge of the term vectors of its subclustors.
   >
   > A lexical affinity is the correlation of a group of words that appear frequently within short distances of each other throughout the given documents. Examples of lexical affinities are phrases like "online library" or "computer hardware". Lexical affinities are generated dynamically and are specific to each collection. (...)
   >
   > The extraction of lexical affinities is superior to a semantic analysis because it is a domain-independent solution. It can derive : set of semantically rich terms without requiring a hand-coded specialized lexicon or a domain-specific thesaurus. (Trybula, 1999:402)

4. La **présentation** des résultats. Trybula indique qu'il y a trois approches pour la présentation des résultats :

   > Les corrélations croisées. Les résultats sont présentés à la manière d'un moteur de recherche plein texte.
   >
   > Les résumés. Il s'agit en fait d'un mode KWIC à partir duquel on peut replonger dans le plein texte.
   >
   > Les visualisations. Il s'agit de présenter les résultats sous forme de cartes et de graphes avec des relations de proximité. On peut faire des zooms sur des parties de la carte pour voir les relations plus fines entre clusters ou documents ou groupes de mots.

En conclusion, Trybula indique « Researchers are realizing that construction of the textbase requires some understanding of the content » (p. 410). Il ajoute : « Text mining results in a collection of related documents from the textbase, but the assimilation of the information into knowledge requires the user to understand and absorb the information being presented. » (p.410).
Ces conclusions sont, pour le moins, évidentes. Aussi, si les procédures exposées dans le contexte du *forage textuel* font partie des méthodes de l'analyse de texte par ordinateur, le contexte théorique de l'analyse de discours, avec sa notion de stratégie discursive, est très différent. En particulier, il suppose un va et vient entre des approches inductives et déductives. Dans notre perspective, l'analyse de texte par ordinateur est un processus interactif et itératif par lequel on construit une lecture par boucles successives d'exploration, catégorisation et validation.

Ces énoncés sur la nature des textes dans un contexte de communication, et les théories analytiques qui en découlent, sont en toile de fond des modèles informatiques visant à soutenir des démarches analytiques sur le matériau textuel. Mais, avant de présenter nos modèles informatiques, il convient de voir à l'œuvre une pratique d'analyse s'appuyant sur les conceptions du texte exposées dans ce chapitre.

2.4 La lecture experte.

Le début des années 1990 a été traversé par une forte influence du courant cognitiviste qui pose d'emblée la question de l'analyse textuelle sous l'angle de la lecture en tant qu'activité cognitive du lecteur individuel. Mais, au-delà de l'individu, la lecture y est aussi abordée sous l'angle de la *lecture professionnelle*, c'est-à-dire une lecture inscrite dans un processus organisationnel de travail. Ce qui nous aura touché plus particulièrement, ce sont les implications de ce type de lecture appliqué à des textes en format électronique et appelant le support d'outils informatiques de lecture et d'annotation.

Ainsi, examinant ce processus de lecture-annotation-interprétation sous l'éclairage des sciences cognitives, des auteurs comme Hochon et Évrard (1994) abordent la question dans un contexte élargi, celui de la *gestion électronique de documents* (GED), et, plus spécifiquement, dans un contexte de gestion personnalisée destinée à soutenir une lecture

professionnelle. Ils avancent donc l'idée de la *lecture annotative* formalisable, à la limite, dans les termes d'un langage d'annotation qui reflèterait les fonctions cognitives de la lecture humaine. Il appert cependant que ce qui est visé, c'est davantage le processus cognitif de l'appropriation du texte qu'un formalisme permettant de marquer les structures textuelles elles-mêmes. Ici, c'est plus l'*analyseur analysant* que l'analyse textuelle elle-même qui est l'enjeu.

Il reste que cette approche, qu'il faudrait probablement coupler avec d'autres théories cognitives de la lecture (apprentissage, intelligibilité, lisibilité, etc.), nous permet d'aller au-delà des considérations documentaires pour aborder davantage l'acte de lire. En effet, contrairement à la vision de l'analyste qui extrait le *contenu objectif du texte*, Hochon et Évrard nous présentent la lecture comme un construit dont les opérations cognitives se déploient ici à travers le processus de l'annotation dynamique. « L'acte de lecture annotative est un acte de discours » (p. 15). Même s'il se situe d'abord sur un plan cognitif, on voit que le point de vue des auteurs est sensible au paradigme sémiotique et de l'analyse de discours.

Dans un article paru en 1989, Louis-Claude Paquin et Jacques Beauchemin, deux chercheurs associés au Centre ATO de l'UQAM, introduisent l'idée de la *lecture experte* dans le contexte où ils se font les porteurs d'une insatisfaction des *travailleurs du texte* par rapport aux outils et méthodes informatiques d'analyse textuelle. Plus précisément, la question est introduite dans le contexte de ce qui est perçu comme une opposition entre deux approches méthodologiques.

> Deux types d'outils d'analyse de textes se disputent la faveur des "travailleurs du texte". D'une part les analyseurs lexicographiques produisent des lexiques (listes de mots) et des concordances (liste de mots accompagnés d'un segment de leur contexte). D'autre part, les analyseurs morpho-syntaxiques associent aux phrases d'un texte les éléments d'une description structurale.

C'est deux types d'outils ont été plus ou moins associés à deux méthodologies d'analyse des données textuelles qui sont souvent tenues pour opposées : l'analyse quantitative où un maximum d'indices est pris en compte et l'analyse qualitative où seuls quelques indices jugés particulièrement significatifs sont considérés. Cette opposition méthodologique a été transposée sur le plan des familles d'outils informatiques. Les analyseurs lexicographiques sont utilisés pour produire des analyses quantitatives basées sur des calculs statistiques, alors qu'on attend des *parseurs* une description exhaustive permettant des analyses qualitatives.

> La pauvreté de certains résultats obtenus par des analyses lexicales imputable à une formalisation insuffisante des données textuelles a fait croire en la primauté du second type d'outil sur le premier. Un tel raisonnement repose sur une définition implicite suivant laquelle la langue naturelle correspond à un ensemble fini de règles circonscrivant un univers de "possibles". Or la supériorité présumée du "parsage" en analyse de texte est discutable pour peu que le texte soit considéré dans toutes ses dimensions et dans toutes ses manifestations. En effet, la description attendue des parseurs, bien qu'exhaustive, ne recouvre qu'un système du texte, celui qui régit l'enchaînement et la hiérarchisation des mots. Il en résulte que les autres dimensions (la référenciation, la thématisation, l'actantialité, l'intertextualité, etc.) restent à couvrir et que l'analyse doit être produite par d'autres moyens.» (Paquin et Beauchemin, 1989)

Même si les auteurs font référence à une opposition entre analyse qualitative et quantitative, l'enjeu est probablement plus spécifique. Il concerne l'apport de la syntaxe générativiste dans le contexte de l'intelligence artificielle de l'époque qui a entretenu l'illusion de l'automate lecteur.

> Face à la complexité de l'analyse des données textuelles, nous proposons de troquer l'automatisation de la lecture experte pour l'assistance à la lecture experte. (Paquin et Beauchemin, 1989).

Les auteurs redonnent sa place au lecteur dans sa relation à la communauté à travers le discours.

> Le texte, comme discours, déborde largement l'univers clos de la rationalité de son objet ou des catégories qu'il met en œuvre. Il s'organise dans une économie de l'énonciation tout aussi porteuse de sens que les objets de la réalité qu'il désigne nommément au lecteur. Le texte connote ainsi les objets qu'il aborde tout autant qu'il les désigne. L'ironie, l'humour grinçant, la déférence, le discours d'autorité et combien d'autres dispositifs sont autant de procédés discursifs que le lecteur expert doit reconnaître et intégrer à son analyse globale du texte. Cette dimension constitutive du texte le pose en objet à "décoder" au-delà des règles proprement linguistiques qui le structurent.

> Mais il y a plus. Le texte doit également être situé dans l'espace social qui le porte et dans les rapports de forces dans lesquels il s'insère. Le texte est toujours tissé de procédés et de stratégies. (...) Dans un mouvement le plus souvent imperceptible à l'œil nu, il converse avec quelque invisible interlocuteur, répond implicitement à ses détracteurs et appelle à sa rescousse ses alliés du moment.» (Paquin et Beauchemin, 1989).

Puis, enchainant sur la lecture, les auteurs introduisent l'idée de *lecture experte*.

> Mais qu'en est-il de la lecture?
>
> Nous avons affirmé que le texte est polyphonique, traversé par les contraintes auxquelles le soumet l'espace pluraliste du discours dans lequel il se déplace et soumis à des modalités d'énonciation définies en société. Nous avons avancé qu'il est en cela déploiement de stratégies discursives. Le décodage des stratégies mises en œuvre dans les textes - menées sur ses multiples registres (morphologique, syntaxique, rhétorique, etc.) - mobilise une expertise aussi vaste que variée. Or, malgré la complexité du processus discursif, un lecteur humain est en mesure, à un degré ou à un autre, de faire une lecture experte des textes qu'il aborde. (Paquin et Beauchemin, 1989).

Dans les années qui suivront, Louis-Claude Paquin développera cette idée de lecture experte qu'il associera à un projet informatique couplant SATO et un moteur d'inférences.

> Plutôt que chercher à mettre au point un analyseur général et exhaustif de l'ensemble des structures des textes, nous proposons d'implanter un modèle de la lecture particulière telle qu'effectuée par des experts sur un ensemble particulier de textes. Cette approche, désignée par l'appellation de lecture experte, s'est imposée à l'occasion de projets de recherches, mais surtout de projets pilotes dans les organisations. Elle justifie pour une large part la commandite du développement d'un Atelier cognitif et Textuel (ACTE) (Paquin, 1992).

Il poursuit:

> La lecture nous apparaît relever plus d'un savoir-faire plus ou moins implicite, que d'un savoir exact formalisé dans une théorie. Il s'agit d'une expertise, acquise non pas tant par apprentissage mais au fil d'une pratique. De plus, cette pratique constitue

> rarement une fin en elle-même, elle est partie intégrante d'une activité professionnelle, comme par exemple, déterminer l'admissibilité d'un dossier ou encore à analyser des récits de vie, etc. Nous explorons un modèle empirique de la lecture pour en dégager les composantes opérationnelles mises en œuvre par les experts de domaine. Dans cette perspective, le lecteur expert effectue sur les textes quatre opérations fondamentales dont la complexité est croissante : segmenter, filtrer, déchiffrer et interpréter (Paquin, 1992).

Bien que fidèle à l'approche développée dans la tradition du Centre ATO de l'UQAM, on assiste ici de la part des auteurs à un certain glissement du focus. On passe de la modélisation par le lecteur des structures textuelles construites sur le texte au processus cognitif et au savoir-faire heuristique du lecteur expert. On donne la primauté au modèle empirique de la lecture traduit sous forme de règles de production et de systèmes à base de connaissances (SBC). Une large place sera donnée aux protocoles de verbalisation destinés à *faire sortir* l'expertise, souvent implicite, du lecteur expert.

> Parallèlement à l'analyse des verbalisations, les concepteurs du système de lecture experte tireront profit d'une analyse de contenu d'un sous-corpus de textes représentatif du domaine de référence. En plus de leur fournir des questions pertinentes à poser aux experts lors des entrevues, cette démarche leur permettra d'évaluer la faisabilité d'une modélisation et de valider les différents aspects du modèle envisagé (Paquin, 1992).

Cette approche se veut aussi une alternative à la *stratégie étapiste* visant à concevoir l'apport de l'ordinateur en ATO comme un processus de calcul épuisant successivement les structures du texte en commençant par la morphologie. En pratique, cette stratégie étapiste a rarement dépassé le niveau du syntagme.

> La description linguistique des unités et de leurs relations à l'intérieur de la phrase est elle-même l'objet d'une stratégie étapiste dont chacune des étapes constitue en elle-même un vaste champ de recherches.
>
> (…)
>
> Une telle stratégie repose sur les postulats qu'un texte est composé d'un ensemble de structures dont les principes d'organisation peuvent être explicités et que, quoique

> intereliées, ces structures sont suffisamment distinctes pour être décrites les unes après les autres. Or rien n'est moins sûr (Paquin, 1993).

Le rejet de la vision étapiste de la modélisation informatique des diverses couches de structure textuelle a pu, dans une certaine mesure, justifier le fait que l'on renonce aux formalismes au profit d'un modèle empirique. Les systèmes à base de connaissance visent d'abord à simuler le raisonnement humain dont la validité repose essentiellement sur la réputation de l'expert. L'étape intermédiaire, qui aurait visé à modéliser une *macro-structure sémantique* en repérant sur le texte des structures d'énoncés, aura alors tendance à être sacrifiée au profit d'un enchainement ad hoc de règles d'inférence permettant difficilement de concevoir des procédures de lecture formalisables et transférables. Telle que décrite, la lecture experte tend à simuler directement les heuristiques de la lecture humaine plutôt que de s'inspirer des indices fournis par les experts pour construire des modèles de *lecture électronique* (Daoust 2002) des structures textuelles.

2.5 L'analyse de texte par ordinateur.

Ce qui caractérise les modèles de calcul inspirés de l'analyse de discours et de la sémiotique, c'est qu'ils postulent la possibilité de modéliser certaines lectures, c'est-à-dire de construire des procédures permettant de dépister et de marquer des traits lexicaux et contextuels, de construire des parcours s'appuyant sur la découverte de relations à l'œuvre dans le texte et que l'on sait reconnaître et interpréter. Ce sont ces lectures et ces reconnaissances-explicitations que l'outil informatique veut appuyer par des algorithmes de découverte et par des procédés de marquage des unités et des structures linguistiques et discursives. L'analyse de texte assistée par ordinateur s'inscrit dans un processus interprétatif qui vise à expliciter ses procédures par la construction de dispositifs expérimentaux qui n'autorisent un certain niveau d'automatisation que dans la mesure où ils traduisent le fonctionnement spécifique d'un genre discursif et de sa mise en texte. Ainsi, la catégorisation peut faire l'objet d'une certaine automatisation dans la mesure où les grilles catégorielles s'inscrivent dans des paradigmes spécifiques. De même on peut envisager une variété de calculs s'appuyant sur les systèmes de traits déployés sous la gouverne de l'analyste.



L'analyse de texte assistée par ordinateur : lunettes de lecture des textes électroniques (2.5a, publication)

Se démarquant du paradigme de l'intelligence artificielle, le paradigme cognitif a voulu remettre le lecteur au centre de l'analyse textuelle en faisant la promotion de la lecture experte. Mais, comme nous l'affirmions en conclusion de la section précédente, ce terme n'est pas sans ambigüité en ce qu'il partage avec son prédécesseur la connotation d'un mimétisme de l'intelligence humaine. Le premier paradigme se place dans un contexte de simulation de l'intelligence humaine alors que le second se situe davantage du côté de la reproduction d'un savoir circonstancié et limité à un domaine d'expertise.

Certes, on est ici dans le domaine de la métaphore. Mais, ces métaphores ne sont pas sans laisser un certain malaise en amplifiant la portée réelle de nos techniques d'analyse. Voilà pourquoi, nous avons profité d'un colloque sur l'édition électronique en 2001 pour présenter l'analyse de texte assistée par ordinateur, également de façon métaphorique, sous l'angle de la lecture électronique des textes. Cette communication, destinée à un public des sciences de l'information peu familier avec l'analyse de discours et l'analyse de texte par ordinateur, a l'avantage d'introduire de façon vulgarisée notre pratique scientifique. Voilà pourquoi nous la reproduisons ici.

Parcours d'une métaphore…

La métaphore de la lunette de lecture peut être prise dans son sens médical de béquille visuelle. Cependant, ce n'est pas dans ce sens-là qu'on l'aborde ici.

La lunette, en sciences naturelles, c'est ce qui a permis de dépasser les limites de notre vision biologique pour atteindre l'infiniment petit et l'infiniment grand. On fait donc référence au microscope et au télescope, ou même à la vision à travers des spectres qu'on ne perçoit pas directement comme l'infrarouge ou les ondes radio.

En passant du support papier au support électronique, le texte devient perceptible par des outils artificiels, c'est-à-dire construits par la technologie sur la base de théories scientifiques. On songe principalement ici à ces outils de calcul logique

que sont les ordinateurs.

On connaît les conséquences évidentes du passage du support papier des textes au support électronique : plus grandes capacités de stockage et diminution des coûts du stockage. Mais, c'est surtout au niveau de la diffusion des textes qu'on mesure l'importance de l'édition électronique couplée à sa distribution à travers le réseau Internet. Aussi, même si l'édifice social et économique de l'édition conventionnelle résiste encore au changement, il reste qu'on a maintenant accès à une quantité astronomique de productions écrites sous forme électronique.

Ces changements posent, de façon dramatique, la question de nos capacités de lecture. Tant qu'on en était à l'édition papier, il était difficile d'envisager des solutions en dehors de la sélection miraculeuse, parmi la masse des documents, de ceux qui répondraient à nos attentes. Mais, le miracle, on le sait, ne fait pas vraiment partie de la démarche scientifique…

Cependant, avec l'édition électronique couplée aux capacités de calcul de l'ordinateur moderne, la quantité peut commencer à être perçue comme un allié plutôt que comme un obstacle. C'est-à-dire que la mesure de la masse documentaire peut elle-même devenir source d'information. Ça, c'est pour le côté “macro” ou “astronomique” pour reprendre notre métaphore télescopique…

Mais, l'ordinateur peut aussi nous fournir un accès nouveau au côté “micro” en sachant reconnaître à la surface des textes des régularités difficilement saisissables dans le parcours d'une lecture conventionnelle. On peut donc envisager la conception d'agents, sortes de lunettes de lecture électroniques, capables d'élargir nos capacités de lecture. Ces lunettes, est-il nécessaire de le préciser, doivent être conçues comme des dispositifs au service de nos finalités interprétatives. En d'autres mots, elles doivent être taillées selon notre prescription.

Stratégies de lecture, stratégies d'énonciation.

Comme lecteur, quand on aborde des textes, on a des attentes. On lit avec nos questions, nos connaissances et nos objectifs du moment. Pour répondre à nos attentes, on développe des stratégies de lecture. Ces stratégies de lecture vont devoir se

confronter à la stratégie d'écriture du texte par son auteur. Le processus de lecture est donc essentiellement une (re)construction du sens qui suit les prescriptions de lectures suggérées par le texte lui-même mais aussi, beaucoup, nos propres besoins et dispositions.

Plus encore, la stratégie d'écriture de l'auteur est elle-même largement redevable au contexte général du discours dans lequel elle s'inscrit! Au-delà du lecteur réel, il y a un lecteur fictif auquel l'auteur s'adresse dans un contexte historique précis tel que perçu par l'auteur. On est donc très loin de la transparence du texte comme simple support de contenus qu'il suffirait de cueillir comme fleurs de printemps!

Examinons une démarche typique d'un lecteur qui recherche des textes dans une démarche informative et analytique.

Dans le contexte de l'édition électronique son premier niveau de stratégie de lecture consistera probablement à rechercher des textes en utilisant, par exemple, les moteurs de recherche sur le Web. Mettons que notre lecteur s'intéresse à la question de la lecture électronique des textes. Donc, il sait que le sujet existe ou il vient de le découvrir. Dépendant de son niveau de connaissance préalable, il devra d'abord trouver les termes utilisés pour aborder le sujet. C'est le problème classique de la recherche documentaire.

Une recherche simple dans les moteurs de recherche risque de nous donner des milliers de références. Notre lecteur devra donc préciser de quel point de vue il veut aborder la question. Recherche-t-il une nouvelle occasion d'affaire? Veut-il savoir comment réagissent les éditeurs à cette nouvelle réalité? Se pose-t-il des question sur l'impact social de l'édition électronique. Est-ce que ça va inciter à lire davantage, par exemple? Est-ce que ça changera nos habitudes de lecture?

Au terme de cette première étape de repérage, notre lecteur, qui veut s'adonner à une lecture de nature professionnelle ou analytique, aura sélectionné son corpus de référence rassemblant les textes jugés pertinents. Maintenant, il va vouloir répondre de façon plus précise à nos questions.

• “Quels sont les différents points de vue qui s'expriment?

• Quels sont les termes du débat?

• Comment les divers acteurs sociaux se positionnent-ils?

• Est-ce que l'âge, le sexe ou l'origine sociale ou géographique distinguent les diverses positions?

• Est-ce qu'il y a évolution des points de vue dans le temps?

• Si je reprends ma recherche sur Internet dans trois mois, est-ce que je vais pouvoir vérifier facilement si le débat a évolué?"

Pour répondre à ces questions, il faut aussi s'interroger sur la nature des textes, sur leur genre. Le texte, on le sait, s'inscrit dans un processus de communication. Il participe à un genre qui en définit la structure générale et que le lecteur se doit de reconnaître pour développer sa stratégie de lecture. La notion de genre dépasse ici l'idée classique de genre littéraire et renvoie plutôt à des conventions sociales plus ou moins explicites. Ces conventions peuvent d'ailleurs appartenir à des groupes sociaux spécifiques. Elles peuvent, ou pas, être sanctionnées par des réseaux institutionnels comme des revues scientifiques par exemple.

D'après François Rastier,

> Un discours s'articule en divers genres, qui correspondent à autant de pratiques sociales différenciées à l'intérieur d'un même champ. Si bien qu'un genre est ce qui rattache un texte à un discours. (...) L'origine des genres se trouve donc dans la différenciation des pratiques sociales (Rastier, 1989:40 cité par Adam 1990:22).

On n'abordera donc pas de la même façon un forum de discussion grand public, des écrits dans une revue informatique donnée, un compte-rendu de colloque à l'ACFAS ou un essai philosophique sur l'écrit à l'ère de l'édition électronique!

L'analyse de texte assistée par ordinateur : dispositifs de lecture électronique.

Cela nous amène donc au deuxième temps de la lecture : l'analyse des textes jugés pertinents. On conçoit aisément que la lecture séquentielle des textes à l'écran, ou des textes transférés dans leur format imprimé traditionnel, est le goulot

d'étranglement dans notre stratégie de lecture.

C'est là qu'entre en jeu l'analyse de texte assistée par ordinateur.

L'analyse de texte vise à faire ressortir les multiples procédés qui structurent les énoncés et positionnent le texte dans le contexte auquel il participe. Ce n'est pas très différent de la pratique scolaire de l'analyse de texte.

Dans cette tâche, on n'utilise pas l'ordinateur pour mimer la lecture humaine. Le radiotélescope ou le microscope électronique ne ressemblent guère à un œil humain pas plus que l'ordinateur ne ressemble à un cerveau humain. Mais, la construction du télescope, son utilisation dans un cadre expérimental et l'interprétation des lectures qu'il nous donne sont l'extension technologique d'une démarche scientifique dirigée par l'humain.

On peut distinguer trois phases dans notre stratégie d'analyse de texte par ordinateur. La première phase consiste à faire parler les données. On a notre corpus électronique et on veut en révéler les caractéristiques générales avant de déployer des stratégies de lecture qui vont dépendre du genre des textes, de leur homogénéité et de leur hétérogénéité et du type de questions posées au texte.

Cette première phase inductive n'est pourtant pas empirique. Elle s'appuie sur des hypothèses générales sur la statistique lexicale, sur les divers procédés de la langue et de la pragmatique textuelle. On tiendra compte également de ce que l'on connaît au préalable du processus d'énonciation et de la structure du corpus. On procède aussi dans un va-et-vient entre le quantitatif et le qualitatif, c'est-à-dire qu'on a toujours le texte au bout des doigts pour vérifier la pertinence des régularités et singularités qu'on dépiste.

Cela nous conduit, de proche en proche, à une deuxième phase d'analyse qui est davantage de nature hypothético-déductive. Il s'agit de construire un dispositif expérimental, une lunette de lecture composée de scénarios de commandes et, possiblement d'opérations de catégorisation ou validation manuelles bien définies.

Par exemple, si on pense que les points de vue exprimés sur la lecture électronique diffèrent qu'ils s'expriment à partir du milieu universitaire ou à partir du milieu de

l'édition ou des vendeurs de produits, on segmentera le corpus en fonction de leur d'origine. On calculera un lexique pour chacun des segments et on déploiera un analyseur lexicométrique sur certaines catégories de mots qualifiés grammaticalement ou sémantiquement. Voilà comment on peut se tailler sur mesure une petite lunette de lecture électronique susceptible de valider ou d'invalider notre hypothèse. On verra peut-être que le lexique de départ est insuffisant et qu'il faut aussi s'intéresser aux mots qui sont cooccurrents avec les termes centraux du débat.

Enfin, on peut imaginer une troisième phase qui consisterait à appliquer nos dispositifs de lecture sur de nouvelles données. La réutilisation et l'adaptation de nos dispositifs de lecture ont alors une double fonction : gagner de la puissance en termes de capacité de lecture, en termes quantitatifs, et aussi mesurer l'évolution du discours afin d'ajuster nos modèles de lecture.

C'est ainsi qu'on se constitue nos propres outils de lecture analytique. Ces outils exploitent des mécanismes généraux de la langue et de la pragmatique textuelle. Mais aussi, ils font appel à notre propre connaissance du monde et matérialisent nos théories explicatives. Ils permettent une certaine reproductibilité de nos lectures ou, à tout le moins, une explicitation de nos procédures analytiques.

De grands défis.

Manipuler un microscope ou un télescope, ce n'est pas vraiment un jeu d'enfant. L'instrument lui-même doit être compris pour qu'on puisse le lire, c'est-à-dire interpréter sa “vision” intimement reliée à la théorie scientifique qu'il matérialise. L'analyse de texte par ordinateur, en nous obligeant à rompre avec une vision naïve de la lecture, requiert donc un élargissement de notre culture scientifique. La lecture devient objet de science comme le montage d'un dispositif scientifique dans un laboratoire de sciences naturelles. Plus encore, l'analyse de discours qui est un peu la base théorique de l'analyse de texte par ordinateur, est essentiellement pluridisciplinaire.

Notre premier défi en est donc un de formation. De la même façon qu'on peut difficilement aujourd'hui ignorer l'intérêt des modèles mathématiques en sciences

humaines, sera-t-il possible demain d'ignorer demain l'apport de l'analyse de texte par ordinateur?

Il faut bien constater aussi les limites de nos théories sur le texte et le discours. Ce que l'on maîtrise le mieux, c'est la dimension lexicale et la lexicométrie de même que l'analyse syntaxique. Mais, au-delà de la phrase ou même de la proposition, on manque de modèles. Le deuxième défi est donc théorique.

Également, on affronte les limites de nos outils de calcul. On investit encore très peu dans ce domaine probablement en conséquence de la faiblesse de nos théories et de la formation dans ce domaine. Il faut aussi noter les limites au niveau des formats du texte électronique lui-même qui est encore le plus souvent une simple image de son équivalent papier ou graphique. La production de textes sur la base de leur marquage logique plutôt qu'éditique est encore à généraliser. C'est le troisième défi.

Mais, à court terme, c'est vraiment le problème de la formation qui bloque l'utilisation à plus large échelle de l'analyse de texte par ordinateur et de la lecture électronique des textes.

Ce qui distingue l'analyse textuelle assistée par ordinateur du simple commentaire interprétatif, c'est la construction de dispositifs expérimentaux qui visent à construire des faits qui soutiennent l'interprétation. Selon Benoît Habert, le dispositif expérimental est un « montage d'instruments, d'outils et de ressources destinés à produire des « faits » dont la reproductivité et le statut (l'interprétation) font l'objet de controverses » (Habert 2005). Dans sa définition du dispositif expérimental, Habert indique qu'un instrument, c'est « un dispositif expérimental qui a réussi ». L'instrument est donc un dispositif stabilisé dont le mode d'emploi et l'interprétation des résultats produits font l'objet d'un certain consensus. Quand on procède à une nouvelle analyse mobilisant de nouvelles questions de recherche ou visant un discours dont le fonctionnement reste à expliciter, on doit élaborer un dispositif expérimental original adapté à des hypothèses nouvelles, ou à tout le moins, à des hypothèses qui se déploient dans des fonctionnements discursifs spécifiques.

D’un point de vue technique, le dispositif expérimental se matérialise par des procédures de calcul transparentes et reproductibles, et par des procédures assistées de catégorisation dont la

trace doit être explicite. Cela signifie que les critères de cette catégorisation sont clairement exprimés et qu'il est possible de retourner au corpus pour repérer les balises qui sont la marque physique de la codification. Ainsi, la controverse de l'interprétation pourra s'appuyer sur la discussion serrée des procédures de constitution des faits sur lesquels elle s'appuie.

L'utilisation des procédures informatisées a aussi pour objectif de permettre la coexistence de plusieurs dispositifs expérimentaux construits sur un même corpus. Ces dispositifs, matérialisant divers points de vue et perspectives théoriques, peuvent soutenir à la fois la complémentarité des points de vue et la multiplicité des parcours interprétatifs correspondant à la nature plurielle intrinsèque de la lecture. C'est à cette approche que s'identifient des logiciels comme SATO (Daoust, 1996a). On en trouve une illustration dans une analyse, à l'aide de SATO, du recueil poétique *Regards et jeux dans l'espace* de Saint-Denys Garneau (Bertrand-Gastaldy et Marchand, 1999). SATO, « permet d'enrichir et de modifier les analyses au fur et à mesure que naissent certaines intuitions, que s'infirment ou se vérifient certaines hypothèses » (p. 64). Ce type de logiciel « offre non seulement des lectures supplémentaires du vocabulaire sur l'axe vertical qui s'ajoutent à la lecture linéaire traditionnelle, mais aussi des déconstructions et des reconstructions d'unités textuelles variées » (pp. 64-65).



L'informaticien, le lecteur et le texte, l'approche SATO (2.5b, publication)

Notre réflexion sur l'apport de l'informatique et de l'informaticien à l'analyse textuelle ne date pas d'hier. Ainsi, il est intéressant de constater qu'un article, publié il y vingt ans (Daoust, 1990), contient plusieurs considérations sur le texte, comme objet informatique, qui demeurent encore très actuelles. Il n'est pas inutile d'en rappeler quelques extraits.

L'informatique et le texte : rappel historique.

Très tôt, l'informatique a dû faire appel au texte. Les codes binaires, en effet, sont indigestes même pour l'informaticien. Voilà pourquoi la programmation a très rapidement fait appel à des systèmes symboliques, les langages de programmation. Et ces langages, tout artificiels fussent-ils, firent appel aux caractères, aux mots et à

la syntaxe. Les caractères, bien entendu, "parlaient anglais" et n'avaient pas d'accent. Les mots appartenaient à un lexique pré-défini, par exemple des codes d'instruction, ou au lexique du programmeur, par exemple des noms de variables. La syntaxe se devait d'être non-ambigüe. Elle permettait de définir la micro-structure, l'énoncé, et la macro-structure, le programme. On a là en petit tous les ingrédients d'un texte. Les informaticiens ont aussi développé des analyseurs permettant de compiler le texte programme et de le traduire dans la langue des codes binaires de l'ordinateur.

Certains langages de programmation ont d'ailleurs été développés spécifiquement pour manipuler l'information textuelle. Ce sont les langages de traitement de chaîne, par exemple SNOBOL et son successeur ICON, ou, de façon plus limitée, les programmes d'édition et de traitement de textes.

La linguistique computationnelle s'est largement inspirée de ces traitements lorsqu'elle a défini les langues naturelles comme son objet d'étude. Les préoccupations de l'informatique pour le **langage** ne sont donc pas nouvelles. C'est cependant la question du **texte** qui a été davantage négligée. Comment? Ne s'agit-il donc pas là de la même question? Pas tout-à-fait, justement.

La problématique du texte origine davantage des sciences humaines ou de ce que les anglophones appellent les "humanities". Les pionniers étaient des théologiens, des latinistes, des médiévistes, des littéraires qui réalisaient des concordances sur ordinateur central. Ces concordances étaient ensuite éditées et permettaient de retrouver tous les contextes d'apparition des formes lexicales triées. Ce sont les littéraires qui développèrent des techniques d'authentification des textes. Ce sont aussi des chercheurs en sciences humaines qui développèrent des méthodes d'analyse de contenu et d'analyse de discours.

Ces recherches s'effectuaient sur ordinateur central, dans les universités et touchaient très peu la "grande informatique". Certes, les informaticiens manipulaient des caractères mais le texte comme objet informatisable n'avait pas encore véritablement émergé.

Le projet SATO lancé au début de 1970 par Jean-Guy Meunier, professeur de philosophie à l'UQAM, est un de ceux qui a posé le plus clairement la question du rapport au texte informatisé. Le choix qui démarquait déjà SATO des concordanciers de la première génération s'appelait "interactivité". Il ne s'agissait donc plus seulement de produire des résultats mais de placer le lecteur, chercheur ou étudiant, devant son texte en lui fournissant des outils pour l'interroger. (Daoust, 1990)

Ces considérations sur la contribution de l'informaticien à la position des sciences humaines sur l'objet textuel n'est pas sans nous rappeler la réflexion sur la philologie numérique (Viprey 2005) dont on a fait état précédemment. L'article de 1990 poursuit en expliquant l'avantage, de ce point de vue, du modèle SATO en montrant la dialectique entre l'apport technique de l'outil et la démarche méthodologique de l'analyste.

Le lecteur et le texte informatisé.

Si, pour l'ordinateur, le texte se présente d'abord comme une suite de caractères, pour le lecteur, le texte est appréhendé d'abord comme une suite de mots, c'est-à-dire d'unités langagières. Bien sûr, le lecteur perçoit les mots écrits à travers un système de transcription alphabétique. Cependant, les caractères et graphèmes agissent ici comme simple support matériel pour les unités langagières. Cette familiarité préalable du lecteur avec l'univers des mots (univers lexical) implique que le texte, du point de vue du lecteur, comporte dès le départ une double dimension.

La première dimension est explicite. C'est l'axe "horizontal" ou syntagmatique qui voit le texte se dérouler comme une séquence linéaire correspondant à l'ordre conventionnel de la lecture. Le mot se perçoit dans un voisinage, un contexte où il participe à une intention de communication.

Cet axe cependant n'est compréhensible, ne fait sens, que parce que les mots ainsi ordonnés participent à un autre système, un autre axe que nous appellerons "vertical" ou paradigmatique. Cet axe fonctionne dans l'univers de la langue d'usage

au sens socio-linguistique, c'est-à-dire dans des systèmes langagiers et sémantiques partagés par les locuteurs, plus précisément ici le rédacteur du texte et la communauté des lecteurs.

L'objectif premier réalisé par SATO est de rendre explicite cette double dimension du texte. Partant de la surface du texte, il s'agit de transformer la "perception machine" du texte pour l'inscrire dans un modèle informatique qui corresponde davantage à la façon dont les humains appréhendent le texte.

Le modèle opérationnel qui permet à l'ordinateur de manipuler un texte, c'est, nous l'avons vu, de le considérer comme une **suite de caractères** inscrits sur un fichier de données. Le modèle que nous voulons construire représente le texte comme une **suite de mots** correspondant à autant d'occurrences d'unités langagières.

Si on en restait à ce niveau de généralité, on pourrait présumer qu'un tel modèle du texte suppose la connaissance préalable des unités langagières, ce qui est déjà un problème immense. Plus encore, la représentation du texte serait alors essentiellement multiple puisque les unités langagières participent à des ensembles sémantiques qui les définissent différemment. C'est le cas en particulier des termes complexes dont le degré de figement est très variable. Tantôt, ces termes pourront être perçus comme des séquences d'unités simples sur l'axe syntagmatique. Tantôt, ils seront vus sur l'axe paradigmatique comme autant d'unités complexes dépassant les règles normales de transcription des mots.

Ainsi, par exemple, dans un texte technique sur les ordinateurs, l'expression "mémoire vive" agit à la manière d'un mot composé (unité complexe) désignant un objet unique que la langue anglaise traduit par l'acronyme RAM (Random Access Memory). Dans un autre contexte de communication, par exemple un texte dressant un portrait psychologique, l'expression "mémoire vive" pourra être perçue comme un état de vivacité qualifiant la mémoire entendue ici dans le sens d'une fonctionnalité associée au cerveau : "Diable qu'il a la mémoire vive...". On aurait donc ici une séquence formée de deux unités simples, "mémoire" et "vive".

En fait, si on ne veut pas quitter la structure de surface du texte, et si on ne veut pas

partir du résultat de l'analyse avant même de l'avoir entamée, il faut s'en tenir à une définition stricte des unités langagières. Cette définition nous est donnée par les règles de transcription alphabétique associée à chaque langue. Elle nous est donnée aussi par les règles d'édition permettant d'inscrire des références de pagination, de mise en page, de titrage, etc. Elle nous est donnée finalement par des règles explicites d'annotation permettant, par exemple, de distinguer les locuteurs quand on rapporte un dialogue.

Dans SATO, les unités langagières que permettent d'identifier ces règles se nomment formes lexicales, ou lexique du texte. Ce **lexique du texte** est à distinguer clairement des **lexiques de domaine**, ou glossaires, qui rassemblent le vocabulaire d'un domaine de spécialisation. Le lexique du texte dressé par SATO contient la liste des unités produites par l'application des règles de transcription alphabétiques ou éditiques données en paramètres. Pour un texte donné et pour un ensemble donné de règles de transcription, on obtient donc un lexique unique. Sur la base de ce lexique, on est en mesure de représenter le texte comme une suite de mots définis comme autant d'occurrences des unités lexicales. Dans SATO, l'axe syntagmatique représente les mots en contexte alors que l'axe paradigmatique ou lexical représente les mots hors contexte.

Les règles de codification ne sont pas toujours suffisantes pour distinguer ce que le lecteur appréhende comme unité langagière. Outre la question des termes complexes qui relèvent de l'analyse, on retrouve des ambiguïtés dans l'utilisation de certains marqueurs. Par exemple, le trait d'union relie les termes d'un mot composé, ou marque une inversion syntaxique, ou introduit un nombre négatif, ou sert de marque de coupure de mot à la fin d'une ligne... En fait, plusieurs de ces ambiguïtés ne peuvent être levées que parce que le lecteur connaît la langue et ses unités lexicales. (Daoust, 1990)

Du problème de la représentation, à la fois linguistique et informatique, on passe ensuite à la question des procédures informatiques dont le déploiement, sous la gouverne de l'analyste, permettra de rapprocher ces deux représentations par une explicitation passant ici par une *distanciation*.

Les avantages méthodologiques de la représentation lexicale.

Les avantages de cette transformation du texte par SATO sont-ils simplement informatiques? Certainement pas. Déjà, nous l'avons vu, le rapport au texte est modifié parce qu'il permet au lecteur d'effectuer des lectures thématiques de son texte, des lectures instantanées qui sont autant de coupes transversales dans le matériau textuel. En ce sens, SATO agit à la manière d'un hypertexte et permet de réaliser le caractère pluriel de l'acte de lecture.

Mais il y a plus. Car, en rendant explicite l'axe paradigmatique, SATO fournit au lecteur un nouvel outil de distanciation. En effet, le lexique apparaît un peu comme un rayon X qui révèle de quoi le texte est fait. Sa nature matérielle, en tant qu'objet d'étude, dépasse donc le papier (ou le fichier) sur lequel il est inscrit. En effet, au-delà de sa singularité, le texte s'inscrit comme une intervention discursive dans un champ sémantique dont la trace est clairement marquée par la "signature lexicale" du texte.

Parce que l'instrument informatique facilite la distanciation, la lecture devient alors plus facilement analyse. Le clavier de l'ordinateur devient le tableau de bord permettant de cheminer dans le texte, de le décrire, de l'exploiter, tel un gisement dont on veut extirper le minerai. On saisit mieux dans ce contexte l'ergonomie de SATO basée sur l'idée de la boîte à outils.

Si l'instrument veut permettre un nouveau rapport au texte, il ne doit pas obscurcir l'objet qu'il veut servir. Bien loin de se substituer à la lecture, qui est essentiellement acte de compréhension au sens étymologique du mot, SATO se veut un instrument pour décupler nos capacités de lecture. Voilà pourquoi SATO, dès le départ, a été conçu comme un programme interactif permettant de maintenir ce rapport intime avec le texte. La nature des outils d'analyse que fournit SATO et le soin apporté à la performance du système correspondent à cet objectif.

(…)

La nature de SATO comme outil d'accès au texte en fait un instrument privilégié de modélisation. A défaut de théories unifiées et de modèles formels, l'analyse de texte

se traduit le plus souvent par des heuristiques permettant de modéliser des fonctionnements localisés du discours. (Daoust, 1990)

SATO, du point de vue informatique, se situe dans le prolongement du postulat méthodologique de l'analyse de discours. En analyse de discours, indique Maingueneau (1987), on recourt constamment à des entrées diversifiées : l'approche par le vocabulaire n'est pas « dépassée » par une prise en compte des modalités appréciatives ou des schémas argumentatifs. Tout est affaire de stratégie de recherche : ces différentes entrées doivent s'éclairer et se corriger l'une l'autre, chacune produisant un certain nombre d'effets qu'il convient de contrôler. Étant donné le statut de l'analyse de discours, on ne peut pas se contenter d'« appliquer » de manière aveugle des protocoles méthodologiques à des corpus. À chaque fois, il faut mener une réflexion spécifique pour construire, de manière interactive, le corpus et son mode d'investigation.



À propos de la variation linguistique (2.5c, remarque)

L'analyse de texte, telle que nous l'entendons dans la perspective de l'analyse de discours, construit l'interprétation au cours de la mise en œuvre de procédures constamment soumises à l'évaluation critique. Un situation qui illustre bien cette exigence, c'est celle de l'analyse de corpus traduits qui subissent l'effet combiné de systèmes linguistiques différents et de l'action d'un rédacteur-traducteur qui doit interpréter le texte dans une langue pour l'adapter dans l'autre. Comment nos procédures réagissent-elles devant ces *corpus parallèles* et comment peuvent influencent-elles nos interprétations et notre herméneutique?

Ce sont ces questions que nous avons abordées à partir d'un cas précis de deux corpus monolingues correspondant à la transcription, en anglais et en français respectivement, d'un débat au parlement canadien portant sur le *Québec comme société distincte* du reste du Canada. Dans chacun des corpus, on retrouve à la fois une transcription dans la langue native des locuteurs, s'ils s'expriment dans la langue du texte produit, et une traduction, dans le cas où l'homme politique s'est exprimé dans une langue différente de celle de la reproduction écrite.

Notre réflexion porte avant tout sur le problème de l’interprétation. Dans

> un chapitre récent (Daoust et Duchastel, 2007), nous distinguons *pluralisme* des interprétations comme « multiplicité des points de vue possibles dans l'observation et l'analyse d'un même objet » et *pluralité* des interprétations en tant que « pluralité des choix herméneutiques qui s'imposent à toutes les étapes de la recherche ». C'est dans ce second sens que nous allons nous intéresser à la variation possible des interprétations des mêmes données discursives selon deux axes : linguistique et logiciel. Après avoir présenté le contexte historique et le contenu des trois discours, nous examinerons l'effet de l'application de diverses techniques lexicométriques proposées par les logiciels SATO, Lexico et ALCESTE dans les deux langues. Notre objectif est de montrer non pas la fragilité des interprétations, mais leur nécessaire complémentarité dans un processus d'analyse incrémentiel. (Duchastel, Daoust, Della Faille 2008)

2.6 Quels modèles de calcul?

Quelle est la place du calcul, et donc de la modélisation informatique, dans les diverses approches d'analyse de texte? Au cœur du débat, on retrouve la question de la frontière entre l'interprétation (au sens d'herméneutique) et l'analyse du matériau textuel.

L'intention de ce chapitre introductif était de montrer que nos modèles informatiques sont guidés par des choix qui visent à répondre à des points de vue théoriques sur le texte et à des approches méthodologiques en analyse textuelle.

Pour certains, le rapport au texte est immédiatement interprétatif. La modélisation informatique n'est là que pour gérer des segments textuels considérés comme des données directement soumises à l'interprétation par la lecture humaine. Dans ce contexte, la codification nécessaire au traitement informatique fait déjà appel à un niveau élevé d'interprétation qui échappe à toute velléité d'automatisation. En amont de la *codification*, les traitements informatiques relèvent surtout de la schématisation conceptuelle. La gestion des segments textuels pourra donc être relayée par une *gestion* des interprétations de l'analyste humain.

Au niveau informatique, les logiciels les plus connus dans le domaine (NUD*IST4 – QSR 1997, InVivo - Fraser, 1999, ATLAS.ti 1998) se présentent donc comme des systèmes de gestion documentaire couplés d'*idéateurs* graphiques. L'idéateur permet de faire des schémas conceptuels dont les nœuds et les feuilles peuvent être des documents mais surtout des index rassemblant un ensemble d'extraits dont l'étiquette (code) sert à nommer un sujet, un phénomène ou un concept à l'œuvre dans les segments codés.

Ce système de liens nommés est vu comme une façon d'asseoir l'interprétation sur les données, les données étant des citations perçues comme immédiatement interprétables par le lecteur analyste. L'impératif de calcul est donc, ici, essentiellement utilitaire. C'est un système de gestion informatisé des documents numériques découpés en énoncés.

Tout à l'opposé, on retrouve l'approche de « compréhension automatique de textes » (Sabah et Grau 2000). Ces courants, associés au traitement automatique de la langue naturelle (TALN) et de l'intelligence artificielle, maximisent la place du calcul. Les succès de ces approches se limitent généralement à des micros mondes caractérisés par des contextes de communication extrêmement contraints. En effet, l'intelligence artificielle et l'automatisation intégrale impliquent qu'on modélise l'acte de lecture lui-même et l'appropriation du texte en termes de connaissance du monde. Il faut dire que le domaine du traitement automatique de la langue naturelle a beaucoup évolué récemment.

Dans *Les linguistiques de corpus*, Habert et coll. (1997) indiquent qu'on assiste présentement à un véritable changement de cap en TALN. Les auteurs n'hésiteront pas à écrire qu'il s'agit d'un *profond changement de paradigme*. Auparavant, la primauté était donnée à la modélisation destinée à formaliser le savoir humain. Le courant dominant était résolument *anti-empirique, anti-numérique et pro-symbolique*. Aujourd'hui, au contraire, indiquent les auteurs de l'ouvrage, suite au peu de résultats opérationnels de l'ancien courant de pensée, on voit de plus en plus la constitution de corpus annotés et de ressources langagières comme une condition au développement de la recherche. Ce nouveau courant est aussi très inspiré par une tradition anglo-saxonne de linguistique descriptive qui s'appuie sur les corpus électroniques. Enfin, il faut inscrire dans cette nouvelle conjoncture l'arrivée massive de textes électroniques et la disponibilité accrue, mais encore insuffisante, d'outils et de ressources langagières informatisées.

Dans sa thèse de doctorat, *Sémantique Interprétative Intertextuelle : assistance informatique anthropocentrée à la compréhension des textes*, Thlivitis (1998) propose un modèle informatique d'assistance à la compréhension des textes qui diffère du paradigme d'un traitement automatique dans la mesure où il reconnait le rôle fondamental de l'humain dans la production du sens en l'intégrant au sein de l'application. En même temps, il tente d'intégrer des éléments de formalisation qui visent à dépasser la subjectivité de l'approche dite qualitative en tenant compte formellement l'interdiscours.

Le modèle applicatif de Thlivitis est essentiellement fondé sur la *Sémantique Interprétative* (SI) de F. Rastier (1989). Cependant, dans le modèle informatique de Thlivitis, l'interdiscours est essentiellement réduit à une intertextualité explicite qui postule une *clôture sémiotique au sein d'un quasi-monde de textes, créé et géré par le(s) lecteur(s)* (extrait de l'introduction). Même si Thlivitis se démarque de l'idée d'*immanence* du sens, même s'il recourt aux dispositifs de l'analyse sémique (*sèmes, sémèmes, taxèmes, isotopies* etc.), en pratique son modèle ne vise qu'à fournir un encadrement méthodologique de manière à forcer le lecteur à expliciter les sources sémantiques à l'origine d'une interprétation dans un texte. Son modèle est donc davantage de l'ordre d'un système de gestion documentaire que de celui d'un analyseur des procédés discursifs.

Bertrand-Gastaldy, qui s'inspire aussi de Rastier, et qui a utilisé dans ses recherches les outils informatiques du Centre ATO dans divers projets de modélisation de *lecture documentaire*, est davantage influencée par les pratiques d'*analyse différentielle* qui caractérisent les méthodes de l'ATO.

Dans *La modélisation de l'analyse documentaire : à la convergence de la sémiotique, de la psychologie cognitive et de l'intelligence artificielle*, Bertrand-Gastaldy et coll. (1995) illustrent comment opère la perspective sémiotique du texte dans une tâche de lecture professionnelle à des fins d'indexation. Les auteurs signalent que toute lecture combine une dimension perceptive et une dimension conceptuelle. Le texte est un entrelacement de multiples systèmes sémiotiques. « La lecture est un acte d'interprétation sensible à certains de ces systèmes selon le projet ou le point de vue » (p.3).

Cette idée d'*analyse différentielle* et de multiplicité des réseaux sémiotiques a des conséquences décisives en termes de modélisation informatique. D'abord, cela veut dire que les systèmes sémiotiques, s'ils reposent sur un ensemble de traits, sont aussi le lieu de leur

reconnaissance. Il n'y a pas de découpage unique des unités textuelles. Non seulement les contours des unités dépendent-ils de la logique de chaque système sémiotique mais, une fois déclenché, le processus sémiotique devient lui-même un processus actif de reconnaissance qui va découper et structurer le texte de son point de vue. En d'autres mots, distinguer et annoter participent à un même mouvement, même si ces opérations peuvent être techniquement distinctes. Il n'y a pas d'unités informationnelles sans système et le processus de reconnaissance et découverte de ces systèmes est à la base d'une méthode qui vise à favoriser le va-et-vient entre le matériau textuel, sa structuration en cours d'analyse et l'acte interprétatif constamment présent dans la démarche.

Si la lecture est multiple, elle n'en est pas moins déterminée socialement et certains de ces processus de reconnaissance peuvent donner lieu à des processus de calcul. L'outil informatique est donc plus qu'un simple support à l'annotation manuelle. On peut s'en servir pour déclencher des processus de dépistage, des lectures machines qui peuvent s'écarter de la lecture linéaire. Même si ces *lectures machines* relèvent de modèles logiques de calcul et ne sont pas de simples calques de la lecture humaine, elles reproduisent, à leur manière, des processus mis en œuvre au cours de la lecture analytique.

La grande mouvance de l'analyse de texte assistée par ordinateur se situe nettement dans une position intermédiaire entre les hypothèses minimalistes et maximalistes, représentées respectivement par les modèles de gestion textuelle et ceux de la compréhension automatique. Ce n'est pas la compréhension au sens fort, l'interprétation herméneutique, qui est l'objet de modélisation, mais l'explicitation sur le texte même des processus discursifs et des réseaux de tous ordres mis en œuvre dans le texte. En contrepartie, les modèles d'analyse assistée, s'ils ne visent pas une compréhension automatique, cherchent néanmoins à révéler des fonctionnements objectifs, calculables, qui sont essentiels à une interprétation raisonnée des procédés discursifs et à leur appréciation herméneutique.

Il faut noter que le courant d'analyse assistée par ordinateur inclut des concepts de gestion électronique des documents (GED) et d'annotation dynamique. Cependant, par rapport au courant cognitif, la perspective est très différente. Car ici, c'est bien le texte lui-même que l'on veut modéliser et non pas les processus cognitifs de l'annotation et de l'idéation.

Finalement, pour compléter ce tableau, il faut introduire un mouvement de pensée inspiré de l'informatique et dont l'objectif est moins de proposer une méthode d'analyse de texte que de refléter la composante structurale des textes.

L'émergence d'une nouvelle préoccupation en informatique portant sur la représentation des structures explicites des documents (documents structurés) laisse entrevoir un nouveau lieu de maillage entre les préoccupations analytiques des sciences humaines et les préoccupations de l'informatique documentaire. La reconnaissance qu'un document est une donnée structurée, semi-structurée pour être plus juste, plutôt qu'un amas plus ou moins informe de chaines de caractères, est susceptible de fournir un nouveau point d'ancrage pour penser la structuration analytique du texte.

Même si c'est la structure explicite des textes, en particulier celle des textes techniques et administratifs, qui est à l'origine de cet intérêt pour les documents structurés, le secteur des sciences humaines et des lettres a vite saisi le potentiel des langages de balisage SGML et XML pour représenter des niveaux de structuration sémantique. Les travaux de la *Text Encoding Initiative* (Ide, 1995) en témoignent. Ces travaux ont d'ailleurs influencé la norme elle-même. Les pointeurs XML, par exemple, sont directement inspirés des travaux du TEI (Michard 1999:100; Bonhomme 2000)

Les langages de balisage ne constituent pas un modèle de traitement, ni même un modèle du texte. Ils fournissent plutôt un formalisme pour marquer l'effet du traitement sur les données brutes. Certes, à l'origine, ces langages ont été conçus moins dans l'esprit d'un traitement à postériori des données textuelles que dans celui d'un traitement à priori destiné à inscrire le flux textuel dans son cadre structurel dès le moment de sa création en tant que document numérique. On est donc dans le paradigme *modèle de document* versus *instance du modèle*, réalisée dans un document concret lors de sa production.

Il appert cependant que ce formalisme de balisage peut aussi être utilisé pour rendre explicite des structures révélées par l'analyse. Cela implique que l'on puisse contourner le modèle hiérarchique unique dicté par la syntaxe XML. Diverses techniques peuvent être utilisées. Par exemple, on peut définir des *éléments vides*, c'est-à-dire sans contenu, qui seront marqués par des *balises frontières* typées, on dit aussi *auto-fermantes*, avec des règles d'interprétation permettant de calculer des empans entre balises frontières (Barnard, 1995). Également, des produits dérivés de la norme XML, comme les liens, illustrent la possibilité de faire du

balisage débarqué, dont les contenus sont référés par pointeur même dans un document XML totalement séparé. Ces techniques seront examinées en détail au chapitre sur l'annotation structurelle.

Ces techniques de balisage s'intègrent parfaitement aux préoccupations de la *philologie numérique*.

> (…) toutes ces composantes entrent technologiquement sous une bannière unique, celle de l'enrichissement par balisage. (…) dès lors que des liens permanents et explicites (les hyperliens, de façon générique) sont marqués, apparaît clairement une plus complète définition du texte, qui n'est donc pas indépendante de la question du déficit ou du comblement philologiques : loin d'être réductible à l'énoncé du discours, le texte, qui en est l'institution, est un ensemble multicouche et multipolaire, extensible et réductible, selon les besoins de toute remise ultérieure en jeu, en discours. (Viprey, J.-M 2005:58-59).

Au niveau de l'exploitation des documents structurés, on trouve des modèles informatiques qui pourraient être élargis pour traiter des structures plus analytiques. Ainsi, Burkowski (1992a), utilise un modèle dit de « contiguous extent », pour représenter diverses structures hiérarchiques. Il s'agit ici d'un modèle de données et de traitement faisant appel à une algèbre dont on peut décrire les propriétés formelles. L'auteur cherche en effet à proposer un modèle qui repose sur une fondation solide, à la manière des SGBD relationnelles. Il applique son modèle au cadre traditionnel de la recherche documentaire en montrant que la représentation des segments emboités permet une navigation qui nous situe constamment dans le contexte logique du document.

Burkowski rapporte qu'on lui a signalé l'applicabilité de ce modèle pour la représentation de couches sémantiques. Cela correspond à l'hypothèse de segments dynamiques que nous avons nous-mêmes développée dans le projet AlexATO (Daoust 1993).

En fait, plusieurs approches formelles peuvent être utilisées pour traiter ces annotations externes. Ce qu'il faut retenir ici, c'est que des formalismes normés et des outils logiciels existent et sont susceptibles d'influencer largement nos modèles informatiques, à tout le moins en ce qui concerne la syntaxe externe des données.

La nécessité de dépasser les frontières phrastiques semble faire consensus en analyse textuelle. L'intérêt pour le développement d'outils de calcul est aussi de plus en plus grand. La définition de la portée des modèles de calcul à la base de ces outils dépend directement de la conception, implicite ou explicite, qu'on se fait du texte.

L'hypothèse minimaliste esquive la nécessité d'une analyse proprement textuelle et linguistique pour passer à l'interprétation directe de segments textuels. Le calcul est alors vu comme une fonction de soutien à la gestion des données textuelles, gestion qui aura son pendant dans le soutien à la gestion et à la représentation de schémas contextuels prenant appui sur les segments textuels *codés* par l'interprétant.

S'inspirant de la psychologie cognitive, d'autres vont assigner au modèle de calcul une fonction de représentation et de support à l'acte de lecture en tant que tel. L'annotation, dans le cadre d'une lecture professionnelle, est donc perçue comme quelque chose de *modélisable*. L'emphase, cependant, est mise sur le lecteur comme agent cognitif plutôt que sur le texte dans ses conditions de production et de diffusion.

Le courant de l'analyse de texte par ordinateur, inspiré entre autres par les modèles de l'analyse de discours et de la sémiologie, insiste davantage sur la notion de stratégie discursive déployée au sein du texte, lui-même artefact d'un discours social qui en détermine les conditions d'existence. Les modèles de calcul visent donc la découverte des processus discursifs et de la stratégie discursive.

Finalement, inspiré par les courants de l'intelligence artificielle, on retrouve chez certains chercheurs, une hypothèse de modélisation maximaliste qui débouche vers une calculabilité de l'interprétation et de la représentation des connaissances acquises par un processus de lecture automatique. On notera cependant un nouvel intérêt, au sein de ce courant de pensée, pour une analyse de corpus plus descriptive.

En pratique, il n'y a pas de frontières étanches entre tous ces courants de pensée. C'est surtout la perspective qui varie. Francis Jacques (Jacques, 1987:76 cité dans Adam, 1990:9), écrivait en 1987 : « L'heure de la mise à feu successive des grandes hypothèses de travail est passée. Celle de la réintégration a sonné ».

Cette *humilité* retrouvée est de plus en plus partagée et productive. Les défis posés par l'analyse textuelle restent toujours aussi impressionnants mais, comme cette courte analyse

exploratoire l'illustre, les ressources théoriques pour en définir les contours sont de plus en plus acérées.

Bibliographie du chapitre 2

Adam, 1990. Adam, Jean-Michel. *Éléments de linguistique textuelle, Théorie et pratique de l'analyse textuelle*. Mardaga, Liège 1990, ISBN 2-87009-440-X.

ATLAS.ti, 1998. Scientific Software Development ATLAS.ti -- *The Knowledge Workbench*. http://www.atlasti.de/ 1998.

Bakhtine, 1978. *Esthétique et théorie du roman*. Paris, Gallimard, 1978.

Barnard et coll., 1995. Barnard, David T.; Burnard, Lou; Gaspart, Jean-Pierre; Price, Lynne A.; Sperberg-McQueen, C.M.; Varile, Giovanni Battista Hierarchical Encoding of Text : Technical Problems and SGML Solutions. *Computers and the Humanities* 29:211-231, 1995.

Benoît, 2002. Benoît, G. Data Mining. *Annual Review of Information Science and Technology*, ARIST, vol 36, 2002, pp. 265-310

Bertrand-Gastaldy et Marchand, 1999. Bertrand-Gastaldy, Suzanne; Marchand, Paul. L'analyse du texte littéraire assistée par ordinateur : essai d'illustration avec Regards et jeux dans l'espace de Saint-Denys Garneau, traité avec le logiciel Sato. *Documentation et bibliothèques*; 45(2); avril-juin 1999 : p. 55-66.

Bertrand-Gastaldy, 1997. Bertrand-Gastaldy, Suzanne. Text Semiotics and Computer-assisted analysis : a multiplicity of points of view for multiple types of users. *Proceedings of the First International Workshop on Computational Semiotics*, May 26-27 1997, Paris.

Bertrand-Gastaldy et coll., 1995. Bertrand-Gastaldy, Suzanne; Giroux, Luc; Lanteigne, Diane; David, Claire. La modélisation de l'analyse documentaire : à la convergence de la sémiotique, de la psychologie cognitive et de l'intelligence artificielle. In *Connectedness: Information*, *Systems, People, Organizations*, Travaux du 23e congrès de l'Association canadienne des sciences de l'information, Olson, H. A.; Ward, D. B. (editors), School of Library and Information Studies, University of Alberta, 1995.

Bonhomme, 2000. Bonhomme, Patrice. Codage et normalisation de resources linguistiques. In *Ingénierie des langues*, chapitre 7, sous la direction de jean-Marie Pierrel, Hermes Science europe, Paris 2000. ISBN 2-7462-0113-5.

Burkowski, 1992a. Burkowski, F. J. Retrieval Activities in a Database Consisting of Heterogeneous Collections of Structured Text. *Proc. 15th Annual International ACM/SIGIR*, ACM, Denmark, 112-125.

Burkowski, 1992b. Burkowski, F. J. An algebra for hierarchically organized text-dominated databases. *Information Processing and Management*, Pergamon Press, New York.

Charmaz, 2000. Charmaz, K. Grounded Theory, Objectivist and Constructivist Methods. **In** Handbook of qualitative research (2nd. ed.), Edited by Denzin. N. K., Lincoln, Y. S. Thousand Oaks : Sage; 2000, pp. 509-535.

Daoust, 2002. Daoust, F. L'analyse de texte assistée par ordinateur, lunette de lecture des textes électroniques. *Communication présentée au colloque Publications et lectures numériques : problématiques et enjeux, 70ième congrès de l'ACFAS*, EBSI, Montréal. http://www.ebsi.umontreal.ca/rech/acfas2002/daoust.pdf

Daoust, 1999. Esquisse du projet de recherche doctorale. Document privé, EBSI, Montréal 1999.

Daoust, 1996. *SATO 4, Manuel de référence*, Centre d'analyse de texte par ordinateur, UQAM, Montréal, 1996. Modifié en ligne de façon régulière : http://www.ling.uqam.ca/sato/satoman-fr.html

Daoust, 1993. Daoust, François. *Une formalisation du modèle des segments textuels en SATO*. In *Analyse de textes et parallélisme (ALEXATO), Projet Alex, Rapport final.* Annexe D, sous la responsabilité de Jules Duchastel; Rapport privé, Centre ATO-CI, 1993.

Duchastel; Daoust et Della Faille (2008). Duchastel J.; Daoust F.; Della Faille, D. Le problème de l'interprétation des données à partir d'un corpus bilingue. L'exemple du discours des trois chefs de parti sur la motion de reconnaissance du « Québec comme société distincte au sein du Canada » , in *Actes des JADT-2008, vol. 1*, pp- 421-431, Presses universitaires de Lyon, 2008. ISBN 978-2-7297-0810-8. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2008/pdf/duchastel-daoust-faille.pdf

Duchastel et Daoust, 2007. Daoust F. et Duchastel J. Pluralisme et pluralité des interprétations. In *Humanités numériques 1, nouvelles technologies cognitives et épistémologie*, 257-268, Lavoisier, Paris 2007. ISBN 978-2-7462-1661-7

Foucault, 1969. Foucault, M. *L'Archéologie du savoir*, Gallimard, 1969.

Fraser, 1999. Fraser, Donald. *NVivo, Reference Guide*. QSR, Melbourne, Australie, 1999.

Glaser et Strauss, 1967. Glaser, B.G.; Strauss, A. L. The Discovery of Grounded Theory. Strategies for Qualitative Research, Aldine, Chicago, 1967.

Guilhaumou et coll., 1994. Guilhaumou, Jacques; Maldidier, Denise; Robin, Régine. *Discours et archive*, Mariga, Liège, 1994. ISBN 2-87009-520-1.

Habert, 2005. Habert, B. *Instruments et ressources électroniques pour le français* Ophrys Paris ISBN 2-7080-1119-7 p.2., 2005.

Habert et coll., 1997. Habert, Benoît; Nazarenko, Adeline; Salem, André. *Les linguistiques de corpus*. Armand Colin/Masson, Paris 1997, Collection U, série «Linguistique», ISBN 2-200-01775-8, 240p.

Halliday, 1985. Halliday, M.A.K. *An introduction to functional grammar*. London ; Baltimore, Md., USA : Edward Arnold, 1985.

Harman et coll., 1996. Harman, Donna; Schäuble, Peter; Smeaton. Document Retrieval. Document Retrieval. In *Survey of the State of the Art in Human Language Technology*, chapitre 3, http://cslu.cse.ogi.edu/HLTsurvey .

Harris, 1951. Harris, Z. H. *Methods in structural linguistics*, University of Chicago Press .

Hochon; Evrard, 1994. Hochon, J.C.; Evrard, F. Lecture professionnelle et gestion personnalisée de documents textuels. *ICO Québec*, vol 6, no 1-2, p. 9-18.

Ide et Sperberg-McQueen, 1995. Ide, Nancy M.; Sperberg-McQueen, C. M. The TEI : history, goals, and future. In *Text encoding Initiative : Background and Context*, Edited by Nancy Ide and Jean Véronis. Dordrecht/Boston, Kluwer Academic Publishers; 1995 : 5-15.

Jacob, 1996. Jacobs, Paul. Text Interpretation : Extracting Information. In *Survey of the State of the Art in Human Language Technology*, chapitre 3, http://cslu.cse.ogi.edu/HLTsurvey .

Laperrière, 1997. Laperrière, Anne. La théorisation ancrée (grounded theory) : démarche analytique et comparaison avec d'autres approches apparentées. In *La recherche qualitative : enjeux épistémologiques et méthodologiques*. Gaétan Morin, Boucherville 1977.

Maier, 1993. Maier, Elizabeth. Textual relations as part of multiple links between text segments. **In** Adorni, Givanni; Zock, Michael, eds. *Trends in Natural Language Generation; an Artificial Perspective*. Fourth European Workshop, EWLNLG '93, Pisa Italy, April 1993. Selected Papers: 68-87.

Maingueneau, 1997. Maingueneau, Dominique. *L'Analyse du Discours, Nouvelle édition*. Hachette, Paris 1997, ISBN 2.01.016907.7.

Maingueneau, 1991. Maingueneau, Dominique. *L'Analyse du discours, Introduction aux lectures de l'archive*. Hachette, Paris 1991, ISBN 2-01-0169-077.

Maingueneau, 1987. Maingueneau, Dominique. *Nouvelles tendances en analyse du discours*. Hachette, Paris 1987, ISBN 2-01-012116-3.

McKenzie, 1991. McKenzie, D. F. *La bibliographie et la sociologie des textes*. Paris. Éditions du Cercle de la Librairie; 1991.

Michard, 1999. Michard, Alain. XML, *Langage et applications*. Eyrolles, Paris, 1999. ISBN 2-212-09052-8.

Patton, 1990. Patton, M. Q. *Qualitative evaluation and research methods (2nd)*, Sage, Newbury Park, 1990.

Pêcheux, 1984. Pêcheux, Michel. Sur les contextes épistémologiques de l'analyse du discours. In *Mots*, Presses de la Fondation nationale des sciences politiques, no. 9, oct 1984.

NUD*IST4. QSR International. *NUD*IST4*. QSR, Melbourne, Australie, 1997.

Rastier, 1989. Rastier, François. *Sens et textualité*, Paris, Hachette, 1989.

Richards, 1999. Richards, Lyn *Using NVivo in Qualitative Research*. QSR, Melbourne, Australie, 1999.

Sabah et Grau, 2000. Sabah, Gérard; Grau, Brigitte. Compréhension automatique de textes In *Ingénierie des langues*, chapitre 13, sous la direction de jean-Marie Pierrel, Hermes Science europe, Paris 2000, ISBN 2-7462-0113-5.

Thlivitis, 1998. Thlivitis, Théodore. *Sémantique Interprétative Intertextuelle : assistance informatique anthropocentrée à la compréhension des textes*. Université de Rennes, Thèse de doctorat, Rennes, 1998, http://www-iasc.enst-bretagne.fr/~thliviti/th/th.html.

Van Dijk, 1985a. Van Dijk, T. A. Introduction : Discourse Analysis as a New Cross Discipline In *Handbook of discourse analysis*, vol. 1, Discourse Analysis in Society, édité par Teun A. van Dijk, Academic Press, London 1985.

Van Dijk, 1985b. Van Dijk, T. A. Introduction : Levels and Dimensions of Discourse Analysis In *Handbook of discourse analysis*, vol. 4, Discourse Analysis in Society, édité par Teun A. van Dijk, Academic Press, London 1985. ISBN 0-12-712-004-1.

Viprey, 2005. Viprey, J.-M. Philologie numérique et herméneutique intégrative. In *Sciences du texte et analyse de discours : enjeux d'une interdisciplinarité* dir. Jean-Michel Adam & Ute . Slatkine (pp. 51-68).

Weaver, 1985. Weaver, C. Parallels between new paradigms in science and in reading and literary theories : an essay review. *Research in the Teaching of English*; 19(3); 1985 : 298-316.

3 SATO : un modèle informatique pour la construction de dispositifs expérimentaux.

3.1 Introduction.

Avant d'aborder l'histoire du logiciel SATO et de ses diverses utilisations, il convient d'en présenter dès maintenant les caractéristiques formelles répondant aux exigences d'un modèle de calcul appuyant l'analyse textuelle inspirée de l'analyse de discours. Comme nous l'indiquions dans le chapitre précédent, ce qui distingue l'analyse textuelle assistée par ordinateur du simple commentaire interprétatif, c'est la construction de dispositifs expérimentaux qui visent à construire des faits qui soutiennent l'interprétation.

Le logiciel SATO est conçu comme une plateforme permettant la la mise en place de ces dispositifs en s'appuyant sur deux grands principes méthodologiques de manière totalement transparente. Cette transparence s'applique, bien sûr, aux procédures mises en place et à leur déploiement spécifique, mais aussi aux données textuelles elles-mêmes qui seront enrichies par de multiples couches d'annotation marquant les étapes et les résultats de l'analyse. La nature essentiellement itérative du processus d'analyse-interprétation prend donc la forme d'un dialogue avec l'artefact textuel qui en gardera une double trace : procédurale et déclarative. L'annotation, se manifestant par diverses formes de balisage, transforme donc le matériau textuel par un processus de qualification non destructive alors que le cahier de procédures qui accompagne cette qualification en fournit la clé de lecture. Ce chapitre exposant le *modèle SATO* abordera donc donc tour à tour la question du modèle de données et celle du modèle de traitement accompagné, à titre illustratif, de la présentation d'un certain nombre d'instruments de mesure inclus dans le logiciel.

3.2 Le modèle lexique / occurrences.

Si, d'un point de vue matériel, le texte se donne d'abord comme une suite de caractères, du point de vue du lecteur, le texte se présente d'emblée comme une suite de *mots qui font sens*. Pourquoi ces mots font-ils sens? D'abord parce qu'ils sont perçus comme des unités, c'est-à-dire des séquences de caractères délimitées explicitement ou reconnues implicitement par le lecteur comme des unités lexicales appartenant à une langue et référant à un monde réel ou imaginaire dont on peut se faire une représentation.

Donc, au-delà de sa définition purement matérielle en tant que feuille de papier imprimée ou page électronique, le texte met en œuvre une dimension implicite, la dimension lexicale, et une dimension explicite qui correspond à l'ordre séquentiel de la lecture.

En conséquence, SATO situe le texte dans un espace, un plan composé de deux axes. On a d'abord un axe lexical qui dresse la liste du vocabulaire utilisé dans le texte. Ce vocabulaire (les formes lexicales) a un sens dans l'univers de la langue et du discours dans lequel s'inscrit le texte. Le deuxième axe représente la linéarité du texte qui se donne comme une suite d'occurrences des formes lexicales. De façon formelle, on peut donc voir un texte donné comme un série de points tracés sur ce plan.



Texte dans le plan lexique/occurrence (3.2a, exemple)

donc	**|**			**x**		
je	**|**	**x**			**x**	
pense	**|**		**x**			
suis	**|**					**x**
	|	‗‗	‗‗	‗‗	‗‗	‗‗
		1	**2**	**3**	**4**	**5**

Représentation linéaire dans la syntaxe de SATO

je pense donc je suis



Un même lexique... deux textes (3.2b, exemple)

donc	**|**			**X**		
je	**|**	**X**			**X**	
pense	**|**					**X**
suis	**|**		**X**			
	|	— —	— —	— —	— —	— —
		1	**2**	**3**	**4**	**5**

Représentation linéaire dans la syntaxe de SATO

je suis donc je pense

Cette représentation du texte dans sa double dimension, lexicale et séquentielle est un choix stratégique du modèle de données de SATO qui aura des conséquences sur le type d'opérations logiques mises en œuvre dans les stratégies d'analyse de texte supportées par le système.



Génération d'un corpus en SATO (3.2c, définition)

La génération d'un corpus par SATO est la transformation de celui-ci en un format que le logiciel peut traiter. En particulier, cette transformation permet de générer le lexique des mots du corpus et d'affecter des valeurs de propriétés aux formes lexicales ou à leurs occurrences dans le corpus. Ce découpage du flux de caractères en formes lexicales est dirigé par des règles associées à des *alphabets* correspondant aux diverses langues utilisées dans le corpus. SATO traitant l'Unicode, tous les caractères sont admissibles. Mais, à part des caractères de contrôle et des espaces, les divers types de séparateurs devront être spécifiés : ponctuations, caractère initiant ou terminant un mot.

Le fait qu'un caractère, au sens de SATO, peut correspondre à une suite de caractères Unicode permet d'indiquer qu'un caractère simple, s'il fait partie d'une chaine plus longue, aura le statut de cette chaine. Par exemple, si on définit la séquence «...» comme caractère séparateur de SATO, elle sera reconnue comme une forme lexicale composée de trois points et non pas comme trois occurrences de l'unité lexicale «.». C'est donc dire que SATO reconnait d'abord

les séquences les plus longues. Un autre exemple de chaines longues concerne l'utilisation des points décimaux (ou virgule selon le standard) : «3.1416» pourra être reconnu comme une unité lexicale si «.1» apparait dans la liste des caractères standards, même si «.» est aussi défini comme séparateur simple. Outre les alphabets, l'entête d'un corpus soumis à SATO peut redéfinir les métacaractères de propriété, de citation et de césure de mot.

Du point de vue linguistique, cette distinction entre *forme lexicale* et *occurrence* en contexte de la forme lexicale est assez classique. Dans SATO, nous utilisons le terme *forme lexicale* pour désigner les entrées normalisées du *lexique du corpus*. Le lexique de SATO est un catalogue rassemblant un exemplaire de tous les mots, grammaticaux ou non, présents dans le corpus soumis au logiciel. Il s'agit, à proprement parler du vocabulaire employé dans le corpus. Par exemple, le mot *il* ou le nombre *3.1416* pourront apparaitre 12, 20 ou 100 fois dans un corpus, mais une seule fois chacun dans le lexique du corpus. On a donc plusieurs occurrences de la forme lexicale dans autant de contextes particuliers. Voilà pourquoi on utilise quelques fois les expressions *mot en contexte* pour désigner l'occurrence (*token* en anglais) et *mot hors contexte* pour désigner la forme lexicale (*type* en anglais).

La notion de *forme lexicale* utilisée dans SATO s'apparente à celle de forme graphique définie par Lebart et Salem (1994) à ceci près que la chaine de caractères extraite du fichier source subit une forme de normalisation consistant, en particulier, à convertir les majuscules en minuscules. Cette normalisation implicite peut être neutralisée en faisant précéder la majuscule d'un méta-caractère, dit *caractère de citation*. Ainsi, dans l'exemple suivant, SATO produira deux entrées lexicales *Pierre* et *pierre.* Il est à noter que le pronom *Il*, débutant la seconde phrase, sera inscrit en minuscules (*il*) dans le lexique du corpus.



Texte avec figement de majuscule (3.2d, exemple)

```
\Pierre possède une pierre précieuse. Il la conserve à l'abri dans
un coffre-fort.
```

Comme on le verra plus loin, la trace de cette normalisation est notée comme valeur de la propriété *Édition* de l'occurrence : *maj*, pour une majuscule limitée au premier caractère et *cap* pour une forme lexicale entièrement en majuscules. La propriété *Édition* agit comme variable d'annotation indiquant le rendu physique de chaque forme lexicale dans son contexte d'occurrence. Ainsi, la notion de forme lexicale se rapproche davantage, linguistiquement, de

la notion de forme fléchie du lemme, incluant le lemme lui-même s'il apparait explicitement dans le texte, que de celle de forme graphique.

Le *lexique du corpus* compilé par SATO est un répertoire des mots présents dans un corpus de textes particulier. En linguistique, le terme *lexique* est davantage compris comme une somme d'unités lexicales disponibles pour un locuteur, un groupe ou une communauté linguistique donnés, voire tous les locuteurs de la langue. On utilise aussi le terme lexique dans le sens d'une compilation du vocabulaire d'un domaine. Par exemple, on pourrait avoir le *lexique de l'Internet* qui rassemblerait les termes (sous la forme normalisée du *lemme*) utilisés dans le domaine de l'Internet. Contrairement au dictionnaire de langue ou au lexique de spécialité, le *lexique du corpus*, compilé par SATO, contient toutes les formes linguistiques du mot (singulier/pluriel, masculin/féminin, conjugaisons du verbe) présentes dans le corpus. Il contient aussi des unités graphiques que l'on ne retrouve pas généralement dans un dictionnaire ou un lexique, par exemple les ponctuations, les nombres, les noms propres et acronymes de toute sorte.

Dans l'idéal, on aimerait que les unités répertoriées dans le *lexique du corpus* se rapprochent le plus possible d'une vision consensuelle des locuteurs de la langue sur ce qu'est un mot appartenant au lexique de la communauté linguistique à laquelle renvoie le corpus. En somme, on aimerait que la compilation du vocabulaire du texte ressemble à une véritable opération de *lexicalisation* au sens linguistique. Comme l'indique Benoît Habert citant Corbin, le lexique est vu « comme un réceptacle ouvert et accueillant des expressions linguistiques lexicalisables, quelle que soit leur origine, qui se lexicalisent parce qu'elles conviennent, à un moment donné et dans une culture donnée, à la dénomination de catégories référentielles » (Corbin, D. (1997b). cité par Habert 1998). L'établissement d'un tel lexique est donc une opération complexe qui relève de l'analyse plutôt que d'en être le préalable. Le lexique, au sens linguistique du terme, implique non seulement des critères formels, mais aussi une évaluation de l'usage inscrit dans le discours et attesté dans le corpus. Une telle évaluation est difficilement réalisable actuellement par un automate à portée universelle.

Du point de vue opérationnel, ce que font généralement les outils informatiques, c'est une segmentation du flux de caractères en suites de caractères adjacents découpées selon des règles que l'on peut généralement formaliser sous forme d'expressions rationnelles. Il peut être éclairant de comparer ce que l'on fait en analyse de texte aux processus de compilation des

langages de programmation en informatique. Dans ce cas, on aura souvent deux opérations qui tendent à se confondre : lexicalisation et, selon le terme anglais, *tokenization*. Les langages de programmation sont constitués de classes lexicales définies en extension ou en intention. Ainsi, on a des mots réservés appartenant au lexique du langage et des classes qui se définissent par leur morphologie, par exemple les identificateurs et les nombres. Dans ce processus de compilation, l'automate repère les lexèmes du langage et attribue des catégories aux occurrences en fonction de règles lexicales et syntaxiques. Ces techniques peuvent être utilisées pour les langues naturelles, mais elles n'ont pas du tout la même portée puisque la langue et le discours sont un ensemble ouvert et constamment modulé par l'intention de communication.

Certains algorithmes de calcul peuvent se contenter d'un découpage en unités très primitif. À la limite, ce découpage peut ne pas aller au-delà du caractère lui-même et de ses classes Unicode. Le *lexique du corpus* pourrait n'être que l'inventaire des caractères Unicode utilisés dans le corpus. La fréquence des différents caractères et de leur classe nous donnera déjà des informations interprétables sur la nature du corpus en termes de langue utilisée ou de genre par la fréquence des ponctuations, des nombres et autres caractères non alphabétiques. Certains outils vont plutôt s'appuyer sur un découpage en séquences de caractères de longueur fixe, du moins pour les chaines alphabétiques. Ces suites de *n-grammes* sont souvent utilisées dans le domaine du traitement automatique de la langue (TAL) pour la correction des fautes, le classement de textes, le passage d'une langue à l'autre dans des textes multilingues, etc. Les *n-grammes* sont généralement utilisés avec des des méthodes statistiques comportant une phase d'apprentissage sur corpus.

Dans la tradition lexicométrique, on utilise surtout un découpage en séquences de longueur variable dirigé par des classes de caractères, en particulier la classe de caractères dits séparateurs. Heureusement, les langues occidentales utilisent ce type de caractères pour identifier les mots simples de la langue, ce qui laisse quand même ouvert le problème des termes complexes et des formes composées, pour ne rien dire de l'ambigüité des règles d'écriture utilisant des caractères pouvant appartenir à plusieurs classes fonctionnelles, par exemple le trait d'union simple et l'apostrophe.

À l'autre extrémité de la complexité, on pourrait disposer d'un dispositif de segmentation qui ferait appel à un ensemble de ressources linguistiques et terminologiques afin de segmenter le

texte en unités lexicales plus fondées du point de vue linguistique. C'est l'approche privilégiée par les logiciels d'analyse automatique qui incorporent des procédures le plus souvent opaques pour l'utilisateur.

Dans une perspective d'analyse de discours, nous avons opté avec SATO pour une position intermédiaire. Nous considérons en effet qu'au-delà de règles simples de découpage, le repérage d'unités requérant un savoir complexe est déjà une opération d'analyse qui trouvera avantage à s'inscrire dans un dispositif contrôlé et transparent exposant clairement les décisions interprétatives qu'implique un processus pleinement qualifié de lexicalisation. Ce dispositif, appliqué sous contrôle de l'analyste, pourra produire un état du corpus gardant la trace de l'analyse sous forme de balises dirigeant une nouvelle lexicalisation jugée plus fondée pour la suite de l'analyse.

SATO se base sur les règles d'écriture alphabétique de la langue pour découper le texte en mots. Ces règles peuvent être insuffisantes parce que le code alphabétique est souvent ambigu. Comme nous l'indiquions précédemment, c'est le cas du trait d'union simple qui est utilisé à la fois pour les mots composés et l'inversion du pronom et du verbe dans la forme interrogative comme dans « *aimes-tu le chocolat? ».* C'est le cas aussi de certains noms propres, dénominations, sigles et expressions composés de mots séparés par des des points d'abréviation ou des espaces, comme dans *assemblée nationale*. Pour forcer SATO à considérer ces chaines de caractères comme des entrées dans le lexique du corpus, on peut les encadrer par les balises **(* et **)* d'ouverture et de fermeture de mot. Voici des exemples.



Texte avec figement de locution (3.2e, exemple)

```
La question nationale a été soulevée à l'*(assemblée nationale*).
As-tu vu ce *(m'as-tu vu*)?
```

Les expressions *assemblée nationale* et *m'as-tu vu,* ainsi balisées, seront inscrites telles quelles dans le lexique du corpus. On verra plus loin que des fonctionnalités de catégorisation de SATO permettent d'effectuer automatiquement ce marquage des termes complexes.

Destiné à soutenir des activités d'analyse, SATO offre la possibilité d'annoter le texte. Le travail d'annotation sur le texte est cette opération matérielle qui permet de marquer par un symbole le dépistage d'une unité cognitive ou sémiotique. Cette unité peut s'établir sur l'axe lexical. Par exemple, on peut reconnaitre que telle forme lexicale appartient à un vocabulaire

familier pour un domaine de connaissances. On peut constater qu'il s'agit d'un adverbe, d'un marqueur d'argumentation, etc. L'unité dépistée peut également se définir sur le plan textuel (occurrence). Par exemple, le lexème *le* qui précède le mot *lexème* agit ici comme article. Ou bien, l'annotation peut consister à identifier en contexte des termes complexes de telle sorte que puisse être produit un nouvel état du corpus qui, soumis de nouveau à SATO, produira un *lexique du corpus* davantage fondé linguistiquement.



À propos de la consolidation terminologique (3.2f, remarque)

Comme dans le cas des *n-grammes* utilisés en TAL, plusieurs analyseurs statistiques utilisés en lexicométrie, par exemple l'analyse factorielle des correspondances (cf. Lebart et Salem 1994), sont peu sensibles au découpage approximatif en *mots*. Ces méthodes, qui proposent une vue synthétique des données, s'appuient sur une analyse multidimensionnelle de l'espace lexique/textes qui permet de relativiser l'apport d'entrées lexicales peu significatives parce qu'ambigües.

Le problème se pose davantage lors de l'interprétation et de la catégorisation directes de ces entrées lexicales. Par exemple, une catégorisation socio-sémantique de la forme lexicale *nationale* peut devenir très difficile si elle renvoie à la fois à la *question nationale* comme enjeu politique particulier et à l'*assemblée nationale* comme dénomination référant à l'espace politique en général. La pertinence d'effectuer une consolidation terminologique relève donc essentiellement de l'analyste en fonction des procédures envisagées et des objectifs de l'analyse.

Dans SATO, on utilise le terme *propriété* pour désigner un système d'annotation permettant de marquer des formes lexicales ou des occurrences. Par exemple, une propriété *connu* et ses valeurs *oui*, *p6*, etc. pourrait servir à identifier les lexèmes connus de tous (comme les nombres), et ceux connus par les élèves de sixième année du primaire dans le système québécois d'éducation. Une propriété *syntaxe* pourrait permettre d'identifier la fonction grammaticale précise de l'occurrence d'un lexème alors que la propriété *gramr* pourrait servir à définir l'ensemble des fonctions grammaticales possibles du lexème. Il s'agit ici d'exemples de propriétés dites *symboliques* qui renvoient à des systèmes de catégories.

En recoupant les systèmes des propriétés avec la représentation du texte en deux dimensions, on obtient donc le modèle suivant.



Texte augmenté de propriétés (3.2g, exemple)

Fréqtot	Gramr	donc	|			x		
1	Con	je	|	x			x	
2	Proper	pense	|		x			
1	Vconj	suis	|					x
1	Vconj		|	— —	— —	— —	— —	— —
				1	2	3	4	5
		Édition		maj	nil	cap	nil	nil
		Partie		prém	prém	conn	conc	conc

Dans cet exemple, nous avons deux propriétés sur l'axe lexical.

- La propriété *fréqtot* est une propriété entière, ce qui signifie qu'elle prend comme valeur des nombre entiers positifs ou zéro. Cette propriété contient le nombre total d'occurrences du lexème dans le corpus de textes.
- La propriété *gramr* est une propriété symbolique dont les valeurs possibles sont des symboles qui désignent des catégories grammaticales. La propriété *fréqtot* est une propriété prédéfinie de SATO alors que *gramr* est une propriété ajoutée.

Sur l'axe textuel, nous avons deux propriétés.

- La propriété *édition* est une propriété prédéfinie de SATO dont les valeurs sont des symboles qui définissent des attributs de mise en page de l'occurrence. Par exemple, le symbole *maj* indique que la première occurrence du lexème *je* débute par un *J* majuscule.
- La propriété *partie* est une propriété symbolique définie par l'analyste pour classer des occurrences d'après leur fonction argumentative : prémisse, connecteur logique, conclusion.

Représentation linéaire dans la syntaxe de SATO

```
*partie=prém Je pense *partie=conn DONC *partie=conc je suis
```

On remarquera que la linéarisation du texte représenté dans le plan SATO peut factoriser les valeurs de propriétés de telle sorte que la valeur commune à une suite continue d'occurrences ne soit pas répétée pour chaque occurrence. Il s'agit là d'un paramètre de l'opération de linéarisation du texte qui peut être modifié au besoin. On remarquera aussi que le texte, une fois *linéarisé*, respecte la syntaxe du corpus en format texte. C'est ainsi que des annotations ajoutées en cours d'analyse avec SATO pourront donner lieu à un nouvel état du corpus qui pourra être soumis de nouveau à SATO produisant une configuration possiblement différente du plan lexique/occurrence.

SATO dispose d'une propriété prédéfinie destinée à contenir la référence de pagination de chaque mot du corpus. Cette propriété contient une valeur structurée en quatre champs :

1) nom du document;
2) numéro de la page dans le document;
3) numéro de la ligne dans la page;
4) numéro du mot dans la ligne.

Cette propriété est gérée automatiquement lors de la soumission d'un corpus.

La référence de pagination en SATO correspond à la valeur de la propriété page. Voici quelques exemples.



Référence de pagination (3.2h, exemple)

```
*page=ton_livre Texte complet du document ton_livre
```

La première référence (***page=ton_livre**) se limite au nom du document, sous la forme d'un identificateur standard. SATO assumera que le document commence au premier mot de la première ligne de la première page du document. À partir de là, la pagination automatique sera utilisée à l'intérieur du document.



Référence de pagination (3.2i, exemple)

```
*page=mon_livre/5/2/3
Extrait du document de mon_livre en page 5, à partir du troisième
mot de la ligne 2...
```

La deuxième référence (***page=mon_livre/5/2/3**) est très précise et cite un extrait de *mon_livre* commençant au troisième mot de la deuxième ligne de la cinquième page du document *mon_livre!* À partir de là, la pagination automatique sera utilisée à l'intérieur du document.



Référence de pagination (3.2j, exemple)

```
*page=/7
Suite de mon_livre au début de la page 7
```

La troisième référence (***page=/7**) indique que l'extrait de *mon_livre* se poursuit au début de la septième page (on présume ici que cette balise suit celle de l'exemple 3.2i). À partir de là, la pagination automatique sera utilisée à l'intérieur du document.



Référence de pagination (3.2k, exemple)

```
*page=@critique_de_paul.txt
```

Enfin, la quatrième référence (***page=@critique_de_paul.txt**) introduit un nouveau document dont le contenu se trouve sur un fichier séparé envoyé sur le serveur. Le nom du fichier est *critique_de_paul.txt* et le nom du document dans SATO sera *critique_de_paul*. Il est aussi possible d'avoir un nom de document différent du nom du fichier, par exemple *paul*, en utilisant la syntaxe suivante : ***page=paul@critique_de_paul.txt**. À partir de là, la pagination automatique sera utilisée à l'intérieur du document.

Puisque la référence de pagination se présente sous la forme d'une propriété SATO, il sera possible, lors de l'analyse du corpus, d'inclure cette propriété dans un filtre pour sélectionner une partie du corpus en fonction de sa pagination. Ainsi, un choix judicieux des noms de documents permettrait de coder simplement des informations concernant l'ensemble du document. Dans un corpus médiatique, par exemple, on pourrait identifier les articles par des noms de document débutant par identifiant du journal, suivi d'une date normalisée, de la page, etc.



Notion de propriété en SATO (3.2l, définition)

Les propriétés dans SATO sont similaires à des variables dans un tableau de données. Ainsi, chaque ligne du lexique correspond à une forme lexicale, et chaque colonne ajoutée correspond à une variable. Le contenu des cases du tableau lexical correspond à la valeur d'une des variables annotant la forme lexicale de la ligne. Une annotation des mots hors contexte est désignée dans SATO par le terme propriété lexicale.

De façon analogue, SATO utilisera le terme propriété textuelle pour désigner une annotation associée à chacun des mots en contexte (occurrences). En termes imagés, on peut voir les propriétés textuelles comme des interlignes ajoutées au texte original pour permettre d'annoter chacun des mots de chacune des lignes du corpus de texte. On peut aussi exporter ces annotations sous forme de tableau : chaque ligne correspond à une occurrence et chaque colonne à une variable.

Si, dans SATO, on peut qualifier une propriété selon sa portée, c'est-à-dire le texte ou le lexique, on peut aussi la distinguer selon le type de ses valeurs. Une propriété sera dite numérique entière si l'on peut lui associer des valeurs entières : 0, 5, 33, etc. Elle sera dite symbolique si elle peut prendre comme valeur un ou plusieurs symboles faisant partie d'une liste fermée de catégories : anglais, français, article, nom, etc. C'est l'utilisateur qui, normalement, définit cette liste d'éléments. Les propriétés symboliques sont dites *ensemblistes* parce qu'un ensemble de symboles peut être employé pour catégoriser un item. Ainsi, le filtrage sur les valeurs des propriétés symboliques pourra faire appel aux opérateurs habituels en théorie des ensembles. Enfin, une propriété sera dite en format libre si l'on peut associer un texte quelconque à une occurrence ou à un lexème.

Le modèle de données de SATO repose sur le concept de classe lexicale et d'héritage. Une occurrence, c'est-à-dire le *mot en contexte* est une instance, une réalisation en contexte de la classe. Toutes les instances partagent, par héritage dynamique, les propriétés de la classe. À ce titre, la chaine de caractères normalisée qui permet de visualiser la forme lexicale, peut-être aussi considérée comme une propriété de la classe. Il sera d'ailleurs possible, au moment de l'affichage de la forme, de substituer à cette propriété implicite une autre chaine contenue dans une propriété explicite. On pourrait, par exemple, afficher la forme avec une orthographe

nouvelle plutôt qu'avec l'orthographe ancienne qui aurait été utilisée dans le corpus original. Aussi, le rendu de l'affichage en contexte de la forme est modulé par une propriété contextuelle (*Édition*) qui retient les attributs de présentation spécifiques à l'occurrence de la forme, par exemple, le fait qu'elle débute par une lettre majuscule. L'héritage dynamique, c'est-à-dire le fait que les propriétés lexicales de l'occurrence existent de façon unique dans la classe, permet un *savoir partagé* entre toutes les instances de la classe, et cela à tous les moments de l'analyse. Ainsi, par exemple, si une propriété lexicale retient la fréquence des mots dans une partie du texte, cette information sera disponible pour toutes les instances de la classe.

À cet héritage dynamique découlant du modèle de données de SATO s'ajoute un dispositif d'héritage statique entre les propriétés, quelles soient lexicales ou textuelles. L'héritage statique signifie qu'on peut créer une nouvelle propriété dont les valeurs de départ seront héritées d'une *propriété mère*. Par la suite, toute modification de la *propriété fille* pour un objet donnée sera spécifique à cette propriété et ne modifiera pas la propriété mère. Cet héritage peut aller dans toutes les directions.

- De lexique à lexique : il s'agit d'une simple copie de la valeur pour chaque forme lexicale ;
- De lexique à texte : il s'agit d'une distribution de la valeur sur chacune des instances de la forme lexicale ;
- De texte à texte : il s'agit alors d'une simple copie de la valeur pour chaque occurrence ;
- De texte à lexique : il s'agit alors d'une opération de synthèse sur l'entrée lexicale des valeurs de propriété affectant les occurrences de la forme en contexte. Cette synthèse prend la forme d'une union ou d'une sommation sur la propriété lexicale héritée des valeurs de propriété affectant les instances de chacune des formes lexicales. Dans le cas de propriétés entières, on additionne simplement la valeur de la propriété de l'occurrence à celle de la propriété lexicale ; pour une propriété symbolique, on procède à l'union ensembliste des valeurs de l'occurrence avec les valeurs de la propriété équivalente de la forme lexicale ; pour une propriété libre, on concatène la chaine de caractères correspondant à la valeur de propriété de l'occurrence à celle de la propriété héritée sur le lexique en autant que la chaine à ajouter ne constitue pas déjà une sous-chaine de la propriété mère.

Ce dispositif d'héritage peut permettre d'affiner en contexte une valeur héritée de la classe lexicale. Il peut aussi permettre de résumer au lexique les valeurs associées aux instances de la classe.

Signalons une autre caractéristique du système de propriétés de SATO. En cours d'analyse, l'ajout d'une propriété lexicale aura pour effet d'enrichir la description des classes lexicales existantes. Cependant, si, au moment de la soumission d'un corpus à SATO, il existe des propriétés à portée lexicale, des valeurs différentes de ces propriétés auront pour effet de produire plusieurs classes lexicales pour une même chaine de caractères comme dans les exemples suivants.



Utilisation de propriété lexicale dans le texte (3.2m, exemple)

Quand je vais à la pêche, je n'oublie jamais d'emporter mes vers***sens="animal"**. Lorsque je les accroche à l'hameçon, je tâche de les installer vers***sens="direction"** le centre.
Si on écrit ***alphabet=en** on the floor ***alphabet=fr** en anglais, on aura une entrée lexicale pour *on* en français et une autre pour l'anglais de la même façon qu'on aura deux entrées pour *vers*, l'une sans le sens d'*animal* et l'autre dans le sens de *direction*.

Ainsi, on pourrait en analyse avec SATO, avoir une propriété textuelle *sens* dont les valeurs seraient ajustées pour chaque occurrence de la forme lexicale. Ensuite, si on exporte le corpus en format texte et que l'on modifie la déclaration de la propriété *sens* pour indiquer qu'il s'agit maintenant d'une propriété lexicale, ce deuxième état du corpus, une fois soumis à SATO, aura pour effet de produire plusieurs entrées lexicales pour *vers* alors qu'il 'y en avait qu'une au départ. C'est un des procédés que l'on peut utiliser pour enrichir le *lexique du texte* suite à un travail d'analyse.

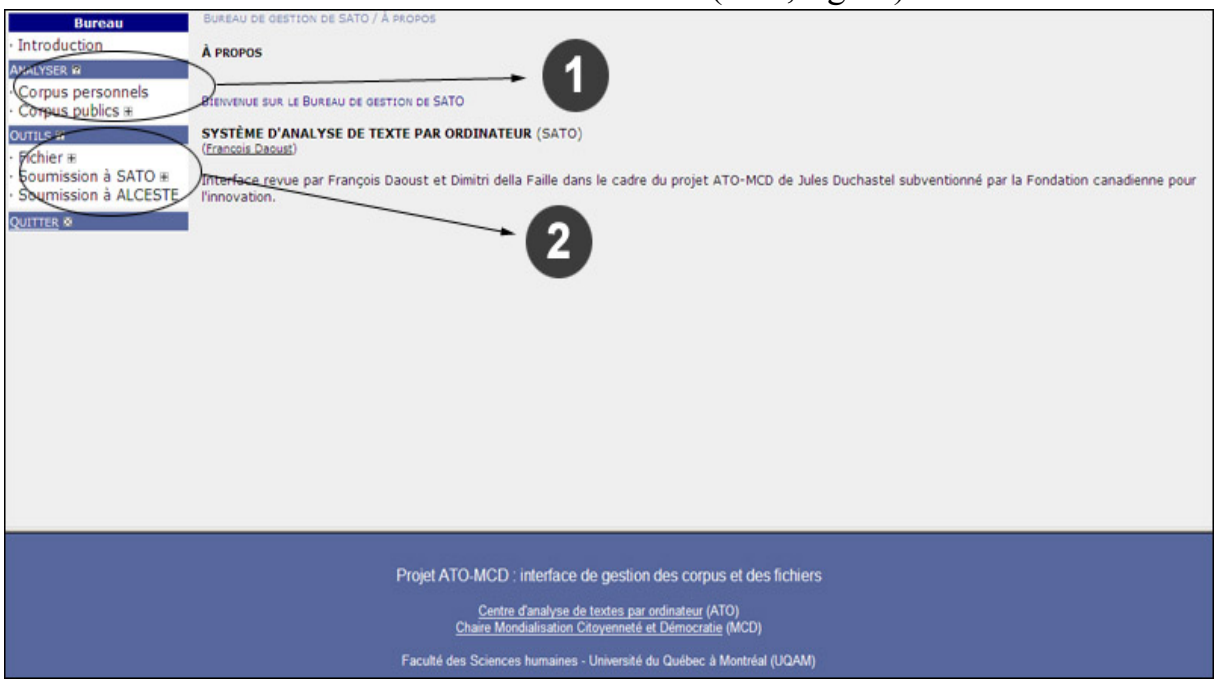
Une fois soumis à SATO, le corpus, mis en forme dans le plan lexique/occurrences, pourra être interrogé et annoté dynamiquement. SATO se présente alors comme un outil de dialogue interactif entre le corpus et le lecteur-analyste.

3.3 L'ergonomie de SATO.

Dans sa configuration actuelle, l'accès au logiciel SATO s'effectue au moyen d'un navigateur Web standard.

Le niveau d'accueil de l'interface est le bureau Web permettant à l'utilisateur de gérer ses fichiers, de les éditer et de faire appel à des outils de mise en forme. Cet interface est accessible à partir du moment où l'usager ouvre une session qui lui donnera accès à ses données personnelles, à des données partagées avec un autre utilisateur ou à des données publiques sur le serveur. C'est à partir de cette première interface que l'utilisateur peut activer des environnements de traitement spécifiques, par exemple SATO, des progiciels statistiques ou tout autre logiciel accessible via un protocole de type *service Web*. Voici une illustration de l'*interface bureau* utilisée par SATO.

Le bureau de SATO 4.3 (3.3a, figure)



Le bureau de SATO est divisé en deux parties principales : *Analyser* et *Outils (*champs 1 et 2 dans l'illustration). La partie 1 permet d'appeler les fonctions d'analyse de SATO sur un corpus existant. La partie 2 donne accès à des fonctions de gestion des fichiers, permet de

soumettre un corpus à SATO et donne accès à d'autres logiciels, Par exemple ici, le logiciel ALCESTE conçu par Max Reinert (Reinert 2002).



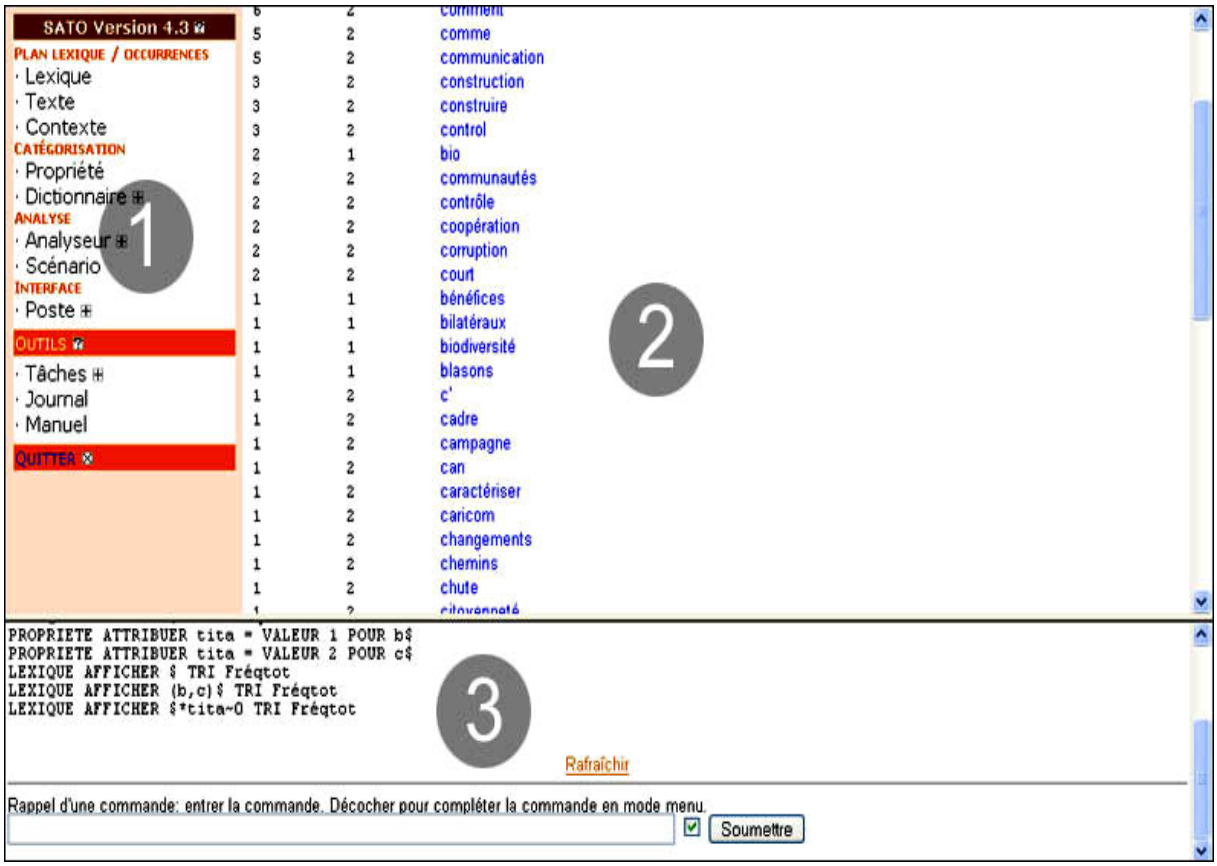
L'onglet *Fichier* de l'interface bureau de SATO 4.3 (3.3b, notice technique)

L'onglet *Fichier* de l'interface bureau donne accès aux opérations usuelles de gestion de fichier : afficher, envoyer, modifier, renommer, copier ou supprimer. Mais il donne aussi accès à des outils plus avancés comme l'utilisation de fichiers de recherche et remplacement de chaines pouvant comprendre des expressions rationnelles (*regular expression* en anglais), des procédures de transformation d'encodage des caractères, ainsi que des procédures de conversion de format vers XML-TEI et vers des formats particuliers de logiciels textométriques. On peut aussi éditer des entêtes déclaratifs de type SATO ou TEI.

Lorsque, à partir de cette *interface bureau*, on active l'analyse avec SATO sur un corpus *généré* lors de la phase de soumission, on entre dans l'interface d'analyse de SATO.

Visuellement, l'écran se présente alors sous la forme d'une page HTML divisée en trois fenêtres (voir figure en 3.3c). La fenêtre de gauche est le menu des commandes (point 1 de la figure 3.3c). La fenêtre centrale est utilisée pour afficher les formulaires de requête et les résultats des commandes (point 2 de la figure 3.3c). Enfin, la fenêtre inférieure servira à diverses fins : affichage des chapitres du Manuel et des bulles d'aide, journal des commandes de la session courante et menu de catégorisation (point 3 de la figure 3.3c). Il est à noter que chaque mot du corpus dans les résultats affichés est un lien cliquable donnant accès au menu de catégorisation. Il pourra s'agir d'une *forme lexicale* ou d'une *occurrence* selon la situation.

L'interface d'analyse de SATO 4.3 (3.3c, figure)



L'environnement de travail livré avec le logiciel SATO repose sur trois dispositifs qui interagissent.

Il y a d'abord une interface interactive qui utilise des formulaires HTML pour guider l'usager dans l'élaboration de ses commandes. Les pages HTML contiennent des liens vers les sections pertinentes du Manuel de référence. Aussi, les formes lexicales et les occurrences affichées à l'écran suite à l'exécution d'une commande SATO seront des hyperliens donnant accès à un menu de catégorisation.

Items du menu de catégorisation de SATO 4.3 (3.3d, définition)

. catégorisation : pour ajouter (+), enlever (-) ou remplacer (=) une valeur de propriété pour le mot pointé ;

_ kwic : pour afficher les contextes courts (key word in context) pour le mot pointé ;

! sauvegarde : pour sauvegarder les modifications aux propriétés ;

[bloc-début : pour marquer l'occurrence pointée comme début d'un bloc de mots ;

] bloc-fin : pour marquer l'occurrence pointée comme fin d'un bloc de mots ;

: bloc-catégorisation : pour ajouter (+), enlever (-) ou remplacer (=) une valeur de propriété pour l'ensemble des mots du bloc courant ;

*** bloc-extrait** : pour extraire le bloc courant et lui donner un numéro ;

- information : pour afficher l'information connue par SATO sur l'item pointé ;

= : applique la dernière catégorisation sur l'item pointé ;

X : représente ici une touche quelconque définie par l'usager. Ces touches permettent de définir des raccourcis pour la catégorisation manuelle. Ces touches peuvent aussi apparaitre dans la fenêtre principale à coté du mot à catégoriser.

Le deuxième dispositif de l'interface SATO est le langage de commandes qui fait en sorte que toute manipulation interactive produit une commande explicite que l'utilisateur pourra saisir et copier dans des fichiers pour constituer des scénarios de commandes. C'est ainsi qu'un utilisateur expérimenté pourra développer des environnements de travail taillés sur mesure et des protocoles d'analyse réutilisables.

Le troisième dispositif caractérisant l'ergonomie de SATO est la production automatique d'un journal dressant un historique complet des interventions sur le corpus. Ce journal vise plusieurs objectifs. D'abord, il fournit un contexte élargi permettant à l'utilisateur d'avoir une mémoire de sa démarche exploratoire. Ensuite, le journal fournit un dispositif de sécurité permettant de reprendre des opérations ou de les corriger le cas échéant. L'examen du journal permet également un retour critique sur une démarche analytique effectuée dans les conditions plus spontanées de la phase exploratoire. C'est souvent à partir de l'examen du journal que l'on composera les scénarios qui vont permettre de cristalliser les stratégies de recherche les plus productives. Le scénario est donc issu généralement d'opérations de repiquage (couper-coller) à partir du journal.

Les scénarios de commandes ont donc un double statut. D'un point de vue technique d'abord, ce sont des programmes permettant de reproduire des stratégies d'analyse et de les appliquer dans un cadre de production. Mais ce sont aussi des objets scientifiques qui sont la matérialisation d'un savoir descriptif ou analytique.

En conclusion, ce qui signe l'originalité d'une analyse de texte avec SATO, c'est en bonne partie cette démarche itérative qui consiste à explorer le corpus en mode interactif, à revenir sur la démarche suivie à travers l'examen du journal pour finalement cristalliser les stratégies fécondes sous la forme de scénarios.

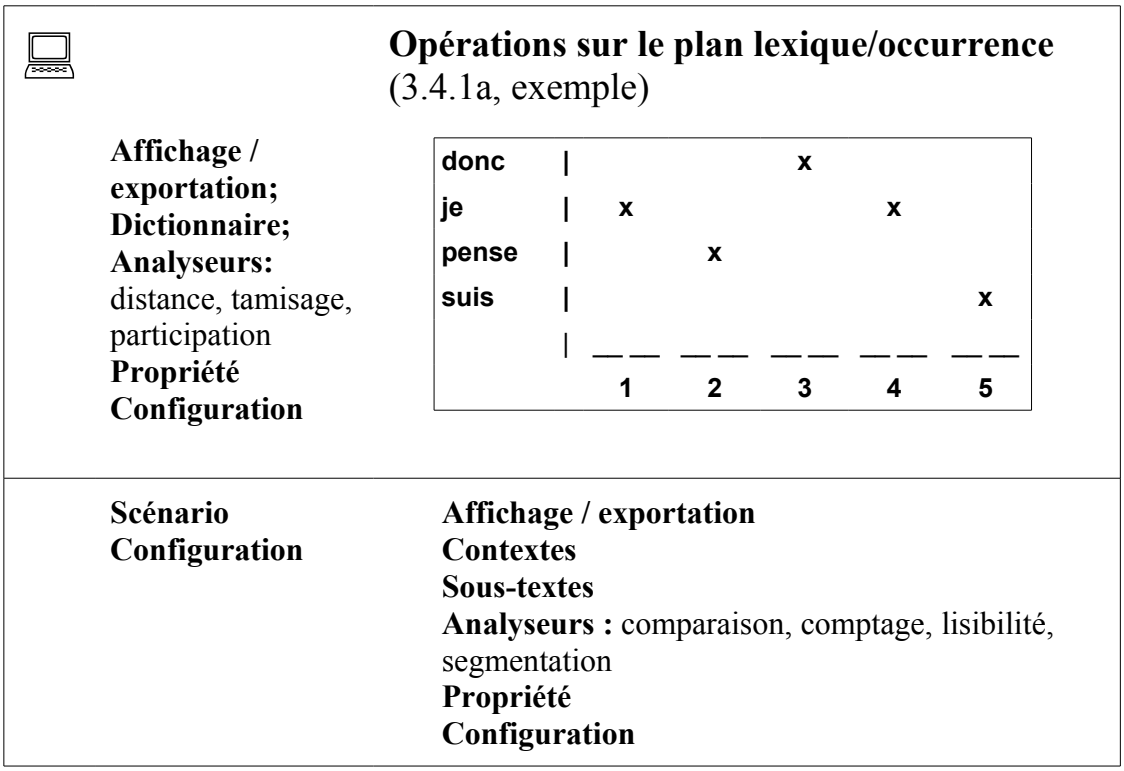
Ce cadre de travail est complété par diverses possibilités d'intervention sur le poste de travail SATO. Par exemple, il est possible d'exporter ses résultats sur un fichier externe et d'agir très finement sur les divers formats de présentation. Cette exportation peut être commandée au besoin ou peut s'effectuer automatiquement afin de conserver une copie des résultats affichés à l'écran. Il est aussi possible d'extraire divers résultats sur un fichier qui pourra être traité par un logiciel externe. Plusieurs modes d'importation des données sont aussi prévus.

Voilà pourquoi le logiciel SATO constitue un outil central pour la construction de dispositifs expérimentaux appliqués à des corpus textuels. Il correspond tout à fait à l'idée que l'on se fait d'un laboratoire, à savoir un ensemble d'outils que l'on peut combiner à loisir. SATO allie les avantages d'un système interactif et d'un système à base de commandes. D'un côté, le mode interactif permet de naviguer rapidement à l'intérieur du matériau textuel. En cela, c'est un outil d'exploration et de découverte. De l'autre côté, le *mode commande* permet de construire des dispositifs reproductibles qui pourront prendre place à l'intérieur de protocoles expérimentaux bien ficelés. Au-delà du cadre expérimental, SATO se présente comme un logiciel générique permettant de bâtir une variété d'applications spécialisées en ATO, applications destinées à des publics particuliers ou plus novices.

3.4 Les opérations dans le plan lexique/occurrences.

3.4.1 Opérations communes

Rapportées sur le plan lexique/occurrences de SATO, les opérations que permet d'effectuer le logiciel se distribuent selon le schéma suivant.



On remarquera que certaines opérations sont disponibles tant sur l'axe du lexique que sur celui des occurrences.

Il s'agit en particulier des opérations d'**affichage** et d'**exportation**, qui sont aussi des opérations de sélection par l'utilisation du mécanisme des patrons de fouille, appelé aussi *filtrage*. Le système SATO, en effet, utilise un langage de commandes dont la syntaxe permet de décrire avec beaucoup de flexibilité les objets primitifs du texte. Les patrons de fouille

permettent de désigner des formes lexicales ou des occurrences par la concaténation dans le patron de filtres portant sur leurs caractères ou sur leurs valeurs de propriété.

Le filtre admet, comme élément de recherche, soit l'expression littérale des caractères ou soit une combinaison de caractères simples et de caractères spéciaux permettant notamment des jeux de troncature des parties gauche ou droite des chaines de caractères. Il est aussi possible de filtrer les caractères ou les valeurs de propriété en utilisant une expression rationnelle de type Perl.



Filtrage sur les caractères (3.4.1b, exemple)

`parle` : le mot *parle* ;

`parle$` : tous les mots débutant par *parle*; $ est un opérateur désignant une chaine quelconque de caractères ;

`p$ent` : tous les mots débutant par *p* et se terminant par *ent* ; l'opérateur $ indique qu'on peut avoir un nombre quelconque de caractères (ou aucun) entre les deux ;

`p_rle` : tous les mots débutant par p suivi d'un caractère quelconque (opérateur _) et se terminant par rle comme parle ou perle.

`parl(e,ent,ure)` : *parle*, *parlent*, *parlure*; les parenthèses introduisent des chaines alternatives séparées par des virgules;

`$ent*fréqtot=5,>5` : tous les mots se terminant par *ent* et dont la propriété fréqtot (introduite par *) est égale à 5 ou est supérieure à 5.



Filtrage sur les valeurs de propriété (3.4.1c, exemple)

`$*gramr=(Nomcom,Adjqua)` : tous les mots dont la propriété gramr possède les valeurs Nomcom OU Adjqua ET n'importe quoi d'autre (un élément parmi l'ensemble) ;

`$*gramr==(Nomcom,Adjqua)` : tous les mots dont la propriété gramr possède les valeurs Nomcom ET Adjqua ET rien d'autre (égalité d'ensembles) ;

`$*gramr~~(Nomcom,Adjqua)` : tous les mots dont la propriété gramr

possède possède autre chose que Nomcom ET Adjqua (inégalité d'ensembles) ;

$*gramr<(Nomcom,Adjqua) : tous les mots dont la propriété gramr possède possède les valeurs Nomcom ET/OU Adjqua ET rien d'autre (sous-ensemble) ;

$*gramr>(Nomcom,Adjqua) : tous les mots dont la propriété gramr possède possède les valeurs Nomcom et Adjqua avec possiblement d'autres symboles (sur-ensemble) ;

$*fréqtot>5 : tous les mots dont la fréquence totale (propriété fréqtot introduite par l'astérisque) est plus grande que 5 ;

$*fréqtot~1 : tous les mots dont la propriété fréqtot est différente (opérateur ~) de 1 ;

$ent*fréqtot=5,>5 : tous les mots se terminant par ent et dont la propriété fréqtot est égale à 5 ou est plus grande que 5 ;

a$*alp=fr : tous les mots débutant par a dans l'alphabet français ;

ab$*gramr=(Nomcom,Adjqua) : tous les mots débutant par ab ayant été étiquetés Nomcom (nom commun) ou Adjqua (adjectif qualificatif) ;

$ing*gramr==Nomcom : tous les mots qui ne sont que des noms communs et qui se terminent par ing ;

$*alphabet~fr*fréqtot=1 : tous les mots qui ne sont pas en français et dont la fréquence totale est 1.

Cette syntaxe de description des mots, combinée à une structure de commandes également très générale, donne à SATO une grande souplesse. C'est ainsi que l'on dispose d'une base solide pour implanter des analyseurs et assurer une communication efficace entre l'utilisateur et le texte informatisé.



Opération d'exportation en SATO (3.4.1d, définition)

L'exportation consiste à écrire sur un fichier en format texte une version linéaire du corpus selon une variété de format : format SATO pour une soumission ultérieure à SATO, format tabulaire, format LISP et format XML TEI selon diverses variantes. On peut inclure les propriétés sous forme de *balises frontières* (*milestone* TEI), structures de traits ou segments (span TEI) dans un fichier séparé. On reviendra sur le format TEI dans le chapitre sur l'annotation structurelle.

Qu'il s'agisse de l'affichage et de l'exportation, on peut sélectionner les propriétés à présenter ainsi que leur format. On peut aussi afficher le texte en couleur en fonction des valeurs d'une propriété symbolique.

Le deuxième ensemble d'opérations, disponible sur les deux axes de notre plan, concerne la définition et l'exploitation des systèmes de **propriétés**.

On peut définir ou redéfinir une propriété, en appliquant, s'il y a lieu, le mécanisme d'héritage déjà décrit. On peut aussi supprimer une propriété existante. On peut attribuer des valeurs de propriétés aux formes lexicales ou aux occurrences. Cette opération se réalise par manipulation directe : on pointe l'objet (forme lexicale ou occurrence) avec la souris et on assigne des valeurs à l'une ou l'autre de ses propriétés. On peut aussi utiliser une commande d'affectation qui permet d'attribuer des valeurs à un ensemble de lexèmes ou d'occurrences décrits par un filtre. On pourra utiliser le mécanisme des concordances sur l'axe textuel pour catégoriser des occurrences en fonction de leur position en contexte. On peut faire appel aux techniques de la statistique descriptive pour dresser un portrait de l'utilisation des valeurs de la propriété sur un ensemble données de formes lexicales ou d'occurrences. Et, on peut modifier un très grand nombre de paramètres de visualisation des propriétés et de leurs valeurs. Par exemple, on peut substituer les valeurs au mot lui-même pour produire des textes transformés.

3.4.2 Opérations sur l'axe des occurrences

La première opération qui concerne spécifiquement l'**axe des occurrences** a trait au repérage de segments textuels. C'est le cas en particulier de la concordance (commande **CONTEXTE** APPLIQUER) qui permet de repérer des passages, c'est-à-dire des contextes qui contiennent

un ou plusieurs mots avec divers types de contraintes de cooccurrence ou de position dans la séquences des mots. La dimension des contextes est paramétrable selon une variété de critères qui peuvent se combiner.



Bornes de contexte dans les concordances en SATO (3.4.2a, définition)

On distingue trois types de bornes pour les contextes :

1. Les bornes DÉLIMITÉES définissent des contextes qui se terminent par un délimiteur à gauche et à droite du mot pôle autour duquel se construit le contexte. Le délimiteur peut être inclus ou exclu du contexte ;
2. Les bornes HOMOGÈNES définissent des contextes constitués des mots adjacents possédant une même valeur de propriété que le pôle de la concordance, par exemple une même valeur pour la propriété locuteur ;
3. Les bornes NUMÉRIQUES définissent des contextes constitués d'un nombre maximal (entier positif ou nul) de mots à gauche et à droite du mot pôle qui est au centre du contexte.

Les trois types de contextes peuvent être combinés. Dans ce cas, le contexte produit sera borné par la première limite valide parmi l'ensemble des bornes spécifiées.

On peut aussi se servir de la concordance pour réaliser une catégorisation automatique des mots repérés. Bien sûr, on pourra afficher ou exporter les passages repérés, en soulignant les mots dépistés en position de contrainte. Cette édition des contextes est accompagnée de références de pagination aussi précises que l'on désire.



Filtre contextuel dans les concordances en SATO (3.4.2b, définition)

Dans un patron de concordance, chacun des filtres désigne un mot (occurrence) satisfaisant aux contraintes exprimées par le filtre. Des opérateurs spéciaux apposés directement à la droite des filtres permettent de définir des contraintes supplémentaires sur le statut et la position des occurrences cherchées dans le contexte. En l'absence d'opérateurs de positionnement, SATO va repérer tous les

contextes qui possèdent au moins un mot répondant à chacun des filtres. L'option implicite est donc une recherche booléenne (logique) basée sur la co-présence (conjonction logique) de mots satisfaisant à chacun des filtres.

Les opérateurs de positionnement sont les suivants :

*. est un opérateur d'ancrage qui indique le passage à une recherche avec contrainte de position; le mot ancré doit donc précéder l'occurrence définie par le prochain filtre; le point peut être suivi d'un nombre qui indique la distance maximale entre les occurrences dépistées; l'absence du nombre indique une distance maximale de 1 (mots adjacents) ; tous les filtres apparaissant à la droite d'un filtre positionnel sont considérés comme positionnels même en l'absence de l'opérateur d'ancrage «.»; donc, tous les filtres agissant de manière booléenne doivent précéder le premier filtre positionnel ;

*~ si le mot DOIT être absent ;

*- si le mot PEUT être absent ;

*+ si le mot DOIT être présent et PEUT se répéter ;

*& si le mot doit être pris comme pôle de la concordance ;

*@ si les contextes doivent être triés selon l'ordre alphabétique du mot pris comme pôle de la concordance plutôt que dans l'ordre de leur apparition dans le texte.

Il est à remarquer que la combinaison *-*+ indique que le mot est à la fois facultatif et répétable.

Dans un patron de concordance, chacun des filtres peut être complété par une attribution de valeur de propriété qui prendra une des formes suivantes :

*propriété:=valeur *(pour remplacer la valeur)*

*propriété:+valeur *(pour ajouter la valeur)*

*propriété:-valeur *(pour retrancher la valeur)*

Pour une propriété numérique, l'ajout et le retrait correspondent aux opérations

arithmétiques d'addition et de soustraction. Pour une propriété symbolique, l'ajout et le retrait correspondent aux opérations ensemblistes de l'union et de la différence. Pour une propriété en format libre, l'ajout est une opération de concaténation de la nouvelle chaine à la fin de la chaine préexistante. La concaténation ne sera réalisée que si la chaine à concaténer ne constitue pas déjà une sous-chaine de la chaine existante. L'opération de retrait correspond à la suppression de la nouvelle valeur si elle existe comme sous-chaine de la valeur existante.

Outre les concordances, il existe en SATO un deuxième mécanisme de repérage de contextes. Il s'agit de l'analyseur **SEGMENTATION** qui a pour fonction de partitionner le texte en segments. Par exemple, on pourrait découper le texte en documents, en paragraphes, en phrases possédant une certaine longueur, ou en segments de longueur fixe. On peut afficher ou exporter un ou plusieurs des segments ainsi repérés. Mais, le plus souvent, on voudra compter des objets dans chacun des segments ou contextes (voir, plus loin, la présentation de l'analyseur COMPTAGE).

La **définition de sous-textes** est une opération très importante du point de vue de l'analyse. En faisant appel aux filtres ou au repérage de segments et contextes, on peut définir un sous-texte. Le sous-texte est une restriction sur l'axe des occurrences de SATO. La commande permet aussi de dresser le lexique du sous-texte dans une propriété lexicale associée. En termes algébriques, cette propriété est en fait le nom d'un vecteur qui représente le sous-texte dans l'espace lexical. On verra plus loin comment l'analyseur lexical DISTANCE permet de mesurer la position relative des sous-textes dans l'espace lexical.



Sous-textes en SATO (3.4.2c, définition)

Le sous-texte est une restriction de l'étendue du corpus à un sous-ensemble quelconque d'occurrences. Par exemple, il peut s'agir d'un chapitre ou d'un document particulier ou de l'ensemble des interventions d'un locuteur ou de l'ensemble des noms, etc.

Lorsqu'un sous-texte est défini, toutes les commandes subséquentes portant sur le

texte s'appliquent à ce sous-texte uniquement plutôt qu'à l'ensemble du texte. Il est possible, au moment de la création d'un sous-texte, de dénombrer les fréquences des formes lexicales présentes dans le sous-texte dans une propriété lexicale. Il est entendu qu'un seul sous-texte peut être actif à un moment donné. Il est aussi possible de rappeler un sous-texte ayant déjà été actif.

Il y a plusieurs façons de définir un sous-texte dans SATO.

- L'option CONTEXTE permet de créer un sous-texte composé des occurrences faisant partie des contextes courants, par exemple toutes les phrases qui contiennent certains mots ou suites de mots.
- L'option DOCUMENT permet de créer un sous-texte constitué des documents qui contiennent, en sur-ensemble, les contextes courants.
- L'option PARAGRAPHE permet de créer un sous-texte constitué des paragraphes qui contiennent les contextes courants.
- L'option PHRASE permet de créer un sous-texte constitué des phrases qui contiennent les contextes courants.
- L'option RAPPEL permet de réactiver par son nom un sous-texte qui a déjà été défini. Il faut noter que le sous-texte rappelé sera strictement identique à l'ancien sous-texte, même si les valeurs de propriété qui auraient pu servir à le construire avaient été modifiées après la définition originale du sous-texte.
- L'option TOUT permet d'annuler le sous-texte courant qui devient alors identique au texte entier.

On peut aussi supprimer ou renommer un sous-texte existant.

Indiquons, pour terminer, que SATO propose des analyseurs qui concernent directement l'axe des occurrences. Il s'agit en particulier de COMPARAISON, COMPTAGE, LISIBILITÉ et TAMISAGE.

- COMPARAISON permet de comparer deux segments textuels quasi identiques et de marquer les différences.

- COMPTAGE permet de compter des classes de mots à l'intérieur de contextes préalablement constitués par concordance ou segmentation du corpus. Par exemple, on pourrait segmenter le texte en phrases et compter le nombre de verbes conjugués par phrase. On pourrait segmenter le corpus en documents et compter le nombre d'occurrences de lexèmes susceptibles d'être des descripteurs de contenu.



Analyseur COMPTAGE de SATO (3.4.2d, définition)

Six options de comptage sont disponibles dans la version 4.3 de SATO.

- L'option DIFFÉRENCES donne le nombre de formes lexicales du contexte courant qui n'étaient pas présentes dans le contexte précédent.
- L'option NOUVEAUX donne le nombre de formes du contexte courant qui n'apparaissaient dans aucun segment précédent.

Ces deux types de comptage peuvent être utilisés pour repérer l'arrivée de nouveaux termes dans le déroulement d'un texte. Cela pourrait donner une indication sur des points de discontinuité correspondant à des changements de thème par exemple.

- L'option LEXÈMES compte le nombre de formes lexicales utilisées dans chacun des contextes.

Pour des segments textuels de longueur similaire, ce comptage peut donner un indice de l'ampleur du vocabulaire utilisé.

- L'option MOTS donne le nombre d'occurrences par contexte d'un ensemble de formes appartenant à un vocabulaire donné.
- L'option VALEURS compte le nombre d'occurrences par contexte pour chacune des valeurs d'une propriété lexicale symbolique.
- L'option FRÉQUENCES donne le nombre d'occurrences par contexte de chacune des formes appartenant à un vocabulaire donné.

L'analyseur peut produire une matrice d'unités par segments pouvant être soumise à des logiciels statistiques externes pour faire de la classification automatique, de l'analyse factorielle des correspondances, etc.

L'analyseur peut calculer divers indices de répartition et de dispersion des objets comptés. Ce sont le nombre total d'occurrences, la moyenne par contexte et l'écart-type associé. On a finalement un indice de répartition qui indique la proportion de contextes où apparait l'objet compté par rapport au nombre total de contextes.

Pour les options FRÉQUENCES, MOTS et VALEURS, on obtient des indices complémentaires.

On a un indice discriminant dû à Salton (Salton 1989) qui donne le poids discriminant maximum de l'objet compté pour l'ensemble des contextes considérés. Pour une forme lexicale donnée, cet indice est calculé de la façon suivante : *Fréq-max* x ln(1/*répartition*)

où *Fréq-max* est la plus grande des fréquences relatives de la forme calculées pour chacun des contextes, et *répartition* est le rapport entre le nombre de contextes où apparait la forme et le nombre total de contextes. *x* indique la multiplication et *ln* le logarithme naturel. Cette mesure est nulle lorsque l'indice de répartition est de 100% et elle croit de façon logarithmique avec la diminution de l'indice de répartition.

On utilise aussi le Chi2 (ou la loi normale si le nombre de contextes est important) pour mesurer l'écart entre les fréquences relatives observées et les fréquences relatives attendues sous l'hypothèse d'une répartition uniforme de l'objet compté.

L'indice de Salton et la mesure du Chi2 (ou cote Z) n'ont pas la même portée. Le Chi2 s'applique aux objets fréquents ayant un fort taux de répartition. Inversement, l'indice de Salton, de nature heuristique, est destiné aux objets peu fréquents et ayant un faible taux de répartition.

- LISIBILITÉ fournit quelques indices de difficulté/facilité de lecture du texte. Notons que nous avons aussi développé SATO-CALIBRAGE, une application utilisant SATO et qui fournit des indices beaucoup plus élaborés que l'indice Gunning calculé par LISIBILITÉ.
- TAMISAGE permet de dresser le lexique des mots ou d'une partie des mots qui apparaissent dans un ensemble de contextes construits par concordance ou segmentation.

3.4.3 Opérations sur l'axe lexical

Un type d'opérations spécifiques à l'**axe lexical** concerne la manipulation de dictionnaires. Pour SATO, un dictionnaire est un fichier externe, une base de données, qui permet d'associer des valeurs de propriété à des chaines de caractères qui représentent normalement des formes lexicales. SATO fournit un ensemble de dispositifs pour créer, consulter et modifier des dictionnaires. On peut aussi les fouiller au moyen de filtres comme on le fait pour le lexique et le texte.

SATO propose quelques analyseurs lexicométriques. L'analyseur DISTANCE permet de repérer les différences dans l'utilisation du vocabulaire entre deux parties du corpus (appelées *sous-textes* dans SATO). Cette mesure est basée sur la *distance du Chi2* appliquée aux fréquences lexicales relatives calculées pour chacun des deux sous-textes comparés. DISTANCE permet aussi d'indiquer quelles sont les formes lexicales, ou les valeurs de propriété de ces formes, qui contribuent le plus à la distance entre les deux sous-textes dans l'espace lexical. On trouvera en 3.5 une exemple d'analyse utilisant l'analyseur DISTANCE.



Analyseur DISTANCE de SATO (3.4.3a, définition)

Imaginons que nous ayons deux propriétés entières pour le lexique, *fable1* et *fable2*. La première (*fable1*) contient la fréquence des formes lexicales dans la fable Le corbeau et le renard. La seconde (*fable2*) contient la fréquence de ces formes dans la fable La grenouille qui veut se faire aussi grosse que le bœuf.

Il serait intéressant de savoir jusqu'à quel point les fréquences obtenues en *fable1* s'écartent de celles obtenues en *fable2*. En d'autres mots, ce que l'on veut, c'est savoir jusqu'à quel point l'emploi d'un vocabulaire donné varie d'une fable à l'autre.

En termes géométriques, on peut voir les deux propriétés, par exemple *fable1* et *fable2*, comme les coordonnées de deux points dans l'espace des formes lexicales (plus précisément le sous-espace des lexèmes décrits par un filtre). Ces coordonnées représentent en fait les fréquences d'utilisation de chaque lexème dans chacun des deux sous-textes. L'analyseur DISTANCE calcule la distance (Chi2) entre ces deux points qui représentent les deux textes dans l'espace lexical.

Supposons, à titre d'illustration, que notre univers sémantique ne soit composé que de trois formes : le point d'interrogation (?), la virgule (,) et le point d'exclamation (!). Les fréquences relatives de ces trois formes sont les suivantes :

	?	,	!
fable1	0.00%	10.00%	1.76%
fable2	2.70%	4.73%	0.00%

Ces chiffres peuvent être reportés sur un système d'axes : l'axe des fréquences d'utilisation du point d'interrogation, l'axe des fréquences d'utilisation de la virgule et l'axe des fréquences d'utilisation du point d'exclamation. Les sous-textes f*able1* et *fable2* peuvent donc être associés à deux points dans cet espace à trois dimensions. La distance mesure l'éloignement entre ces deux points. Dans cet exemple, on utilise la fréquence de formes lexicales. Mais, on pourrait aussi utiliser les fréquences de valeurs de propriétés lexicales, ce qui signifierait que l'on prendrait en compte, pour chacune des valeurs, l'ensemble des formes possédant cette valeur.

La *distance du Chi-carré* utilisée ici a la particularité de pondérer par une fréquence moyenne les écarts de fréquence calculés sur chacun des sous-textes. Cette fréquence de pondération est généralement *fréqtot*, la fréquence calculée sur l'ensemble du corpus.

Après avoir écrit la distance calculée, la commande ANALYSEUR DISTANCE repère les 250 axes lexicaux qui contribuent le plus à cette mesure de distance. Ces axes sont présentés dans l'ordre de tri décroissant des carrés dont la somme sert à calculer la distance. On a ainsi une image de la contribution relative de chacun des axes lexicaux à la distance totale. Ainsi peut-on voir quelles sont les formes (ou valeurs de propriété) qui marquent davantage l'originalité du vocabulaire d'une partie du corpus par rapport à une autre.

L'analyseur PARTICIPATION est le complément naturel de DISTANCE. Il permet de calculer la part relative d'un ensemble de mots dans les divers sous-textes du corpus. En effet,

avec DISTANCE on compare deux sous-textes dans l'espace lexical pour identifier les axes les plus discriminants. Avec PARTICIPATION, on se concentre sur un axe, ou plus judicieusement, sur une combinaison des axes portant une même valeur catégorielle, ce qui permet d'obtenir un nombre d'occurrences suffisamment élevé pour autoriser un test statistique.

On utilise la *cote Z* (écart-moyen centré et réduit) pour évaluer, sur la base de la loi normale, la signification statistique de l'écart de fréquence entre un sous-texte donné et l'ensemble du corpus pour une catégorie de mots donnée. Par exemple, dans un corpus d'entrevue, si la mesure de distance indique que le pronom *je* distingue les répondants masculins des répondants féminins, on pourra faire la conjecture que l'utilisation des formes pronominales selon le sexe des répondants, pourrait indiquer une position d'implication plus forte chez l'un que chez l'autre. On utilisera l'analyseur PARTICIPATION pour mesurer la portée statistique de l'utilisation différenciée des pronoms associés à la première personne (je me moi...) dans les divers sous-textes définis répondants féminins, masculins, âgés, jeunes, etc. afin de tester cette hypothèse sur les différents profils de répondants. On trouvera en 3.5 un exemple d'analyse utilisant l'analyseur PARTICIPATION dans ce sens.

PARTICIPATION peut aussi être appliqué directement sur l'axe des occurrences. Il permet alors de calculer la proportion d'une classe quelconque de mots dans le corpus ou le sous-texte courant.

3.4.4 Opérations combinant les deux axes.

Sur le plan lexique/occurrences de la représentation SATO, nous avons placé le scénario à la **jonction des deux axes**. En effet, le plus souvent, les scénarios déploient des stratégies faisant appel à la combinaison de fonctions qui agissent sur l'un ou l'autre des deux axes.

Les scénarios prennent la forme de fichiers en format texte composés de séquences de commandes SATO. Ils sont, le plus souvent, composés à partir d'extraits du journal qui enregistre automatiquement toutes les opérations effectuées grâce à SATO. Une fois que l'on a mis au point des stratégies d'analyse en mode interactif, on reprend ces stratégies et on en fait des analyseurs.

Comme on peut le constater, SATO fournit peu d'analyseurs prédéfinis. Les outils statistiques disponibles sont simples et facilement interprétables par tout analyste ayant une formation de base en statistiques pour sciences humaines. Par ailleurs, SATO permet de produire des matrices de données qui pourront être soumises à des progiciels statistiques spécialisés.

En fait, ce qui signe l'originalité de SATO, ce sont surtout des opérations dont la combinaison permet à l'utilisateur de construire ses propres procédures et protocoles d'analyse. Ceux-ci vont prendre la forme de scénarios que l'on fait exécuter. Certains ont d'ailleurs comparé SATO à un *tableur textuel*. Le tableur, ou feuille de calcul électronique, est ce type de logiciel qui permet à l'utilisateur de définir diverses fonctions de calcul associées à une matrice de données. Évidemment, les données textuelles et les fonctions de calcul disponibles en SATO ne sont pas celles du tableur mais la métaphore indique que SATO est, comme le tableur, un outil permettant de développer ses propres *formules*, c'est-à-dire des propres protocoles expérimentaux destinés à saisir les frontières des discours dont la trace se trouve dans le corpus sous analyse.

3.5 Un exemple d'analyse illustrant la construction d'une grille catégorielle.

3.5.1 Introduction

Pour concrétiser cette présentation des fonctionnalités de SATO, on pourrait faire appel à de multiples exemples d'utilisation dans une variété de projets. Certains de ces projets seront d'ailleurs présentés dans le chapitre 4. Même si ce choix peut-être arbitraire, nous avons opté ici pour une analyse ayant fait l'objet de deux communications aux JADT (Gélinas-Chebat et coll. 2004, Daoust et coll. 2006). Il s'agit de l'analyse d'un corpus d'entrevues de jeunes sur leur attitude par rapport au tabagisme. Comme ce projet a aussi été l'occasion d'utiliser d'autres logiciels d'analyse de données textuelles, sa présentation permettra de montrer comment on peut combiner des approches inductives et exploratoires et une approche hypothético-déductive qui vise à construire des protocoles visant à valider des hypothèses découvertes en exploration.

3.5.2 Analyse exploratoire d'entrevues de groupe : les jeunes Français et le tabac

L'article *Analyse exploratoire d'entrevues de groupe : les jeunes Français et le tabac* (Gélinas-Chebat et coll. 2004) a pour objectif de présenter, en prenant comme exemple une analyse exploratoire de corpus, une méthodologie d'analyse textuelle visant à construire un système de catégories de manière itérative. Le corpus est constitué de transcriptions d'entrevues de groupe sur l'usage du tabac. Des algorithmes simples de comparaison statistique entre lexiques associés à des sous-textes correspondant aux profils sociologiques des jeunes sont utilisés pour appuyer la construction d'un système catégoriel faisant le pont entre la problématique de recherche et les données textuelles.

Cette recherche tente de comprendre la portée des messages contre l'usage du tabac chez les adolescents avec l'objectif à long terme de réduire significativement leur consommation de cigarettes. Les adolescents minimisent les risques de l'usage des produits dangereux et tendent à sous-estimer les dangers de l'usage du tabac (Leventhal et coll., 1987). Quel discours doit-on tenir dans ces messages d'avertissement pour qu'ils soient efficaces? Une étude comparative de différentes recherches empiriques sur les effets des menaces dans le domaine des mises en garde sur la santé montre que plus le message suscite des émotions de peur, plus les effets sur l'attitude, l'intention et les changements de comportements sont grands. De même, plus la sévérité du message est forte, plus l'attitude, l'intention et les comportements changent (Witte et Allen, 2000). Mais le goût du risque, par exemple des sports extrêmes et l'attrait du « fruit défendu » (Parker-Pope, 1997) n'a-t-il pas l'effet contraire à celui désiré ?

C'est dans ce contexte qu'a été constitué un corpus rassemblant des données recueillies dans le cadre d'une discussion ; des groupes de plusieurs participants discutaient sur le thème de la cigarette, puis ils étaient exposés à message d'avertissement. Le corpus comprend neuf entrevues sur le tabagisme chez les jeunes et leur perception de la publicité contre l'usage du tabac. Elles ont été réalisées à Rennes en 2000 auprès de 48 jeunes Français qui, pour la plupart, fréquentent une institution scolaire et qui sont âgés de 15 à 25 ans. Chacune des séances réunit 5-6 jeunes (fumeurs ou non-fumeurs, hommes et femmes) et un intervenant, et se divise en deux parties. La première partie se déroule après que l'intervenant a posé quelques questions pour amorcer la discussion et la seconde se caractérise par l'introduction

d'une brochure. Il existe différentes versions de la brochure selon deux paramètres : les effets du tabagisme sur la santé et les solutions pour arrêter de fumer. Ces deux paramètres se présentent comme suit, trois niveaux de menace (faible, moyen et fort), et deux niveaux de solution (faible et fort).

Les entrevues ont été enregistrées sur bandes audio et retranscrites en format texte. Au début de chaque transcription, les données sociologiques des personnes qui participent à l'entrevue sont précisées : âge, sexe, fumeur/non-fumeur. Pour le traitement avec le logiciel SATO, les annotations éditiques initiales ont été remplacées par un balisage symbolique conforme à la syntaxe de SATO. Un astérisque introduit les balises, aussi appelées *propriétés*, et celles du corpus sont les suivantes : *locuteur, *sexe, *fumeur, *page et *thème. En voici un exemple :



Codification SATO du corpus d'entrevues (3.5.2a, exemple)

*page=gallo02/11

***thème=brochure (…) *locuteur=s36 *fumeur=non *sexe=ho** Bah, la brochure là, elle nous présente ce qui nous attend si on fume. Mais c'est très... quoi, moi j'ai lu ça, mais je ne sais pas je ne suis pas fumeur, donc je ne ressens peut-être pas ça de la même façon. À la limite on passe dessus comme ça, ça apporte quelques chiffres.

Dans un premier temps, la démarche exploratoire utilisée pour l'analyse de ce corpus ressemble à l'approche adoptée pour le corpus *Message d'amour* (Daoust 1999) et qui vise à *faire parler les données*. Le principe de base de la démarche consiste à comparer, avec des indices statistiques simples, les lexiques associés à des sous-textes découpés d'après nos variables sociologiques. Dans un deuxième temps, cette comparaison sera reprise de façon itérative de façon à s'appuyer sur le lexique brut pour construire un lexique catégorisé reflétant les points d'ancrage de la chaine interprétative.

Le découpage du texte et la constitution des lexiques associés procèdent comme suit. Les balises (*propriétés*) introduites dans le corpus permettent de segmenter les données en opposant, par exemple, l'ensemble des interventions *avant* et *après* la présentation du document publicitaire. De la même façon, on peut découper le corpus entre, d'une part, les interventions des hommes et, d'autre part, celles des femmes, excluant les interventions des modérateurs. Ces balises étant indépendantes, elles peuvent être combinées à loisir lors de

l'exploration du corpus. Ainsi, on pourra comparer les interventions des hommes seulement, ou des femmes seulement, *avant* et *après* la présentation de la brochure pour voir si la réaction à la brochure dépend du sexe des sujets.

Pour comparer les lexiques des sous-textes, on utilisera un *algorithme de distance lexicale* basé sur la *distance du Chi2*. La mesure évalue l'écart dans l'emploi d'un vocabulaire donné entre deux sous-ensembles du corpus. Les formes lexicales sont triées par ordre décroissant de contribution à la mesure de distance, ce qui permet d'identifier, par ordre d'importance, les spécificités de chaque sous-texte. L'algorithme peut être appliqué aux formes lexicales elles-mêmes ou aux valeurs de propriétés correspondant à une catégorisation lexicale. Ici, l'approche est essentiellement dichotomique : on compare un sous-texte à un autre, via leur lexique respectif. On peut aussi avoir recours à un *algorithme de participation* qui calculera les moyennes normalisées d'un ensemble de formes lexicales, correspondant généralement à une catégorie lexicale, pour chacun des sous-textes constitués en cours d'analyse.

La démarche exploratoire est fondée sur un va-et-vient interactif entre ce que nous révèle l'analyse lexicale et les contextes d'utilisation des mots mis en évidence par les algorithmes de distance et de participation. Dans ce premier temps de l'analyse, c'est une approche univariée qui a été privilégiée afin de mieux saisir la spécificité de la stratification induite par chacune des variables sociologiques. Dans un deuxième temps et avec les mêmes outils, on va comparer des sous-textes constitués selon plusieurs caractéristiques pour tenir compte, par exemple, à la fois du sexe des intervenants et de l'introduction de la brochure dans la discussion.

Au premier niveau de l'analyse, la comparaison porte sur les données brutes, c'est-à-dire les formes lexicales elles-mêmes. On se donne ainsi la possibilité de voir apparaitre des différenciations portées par la morphologie des mots en termes de nombre, genre, personne, temps. On s'intéresse tout autant, sinon davantage, à des marqueurs d'énonciation, comme les pronoms personnels, les marqueurs phatiques, les marques de la négation, de l'interrogation, les verbes épistémiques (croire, penser...), etc. qu'aux termes pleins. C'est en s'appuyant sur l'analyse lexicale des données brutes que seront élaborées les grilles catégorielles.

Le retour constant aux énoncés, ne serait-ce que par un parcours rapide des contextes courts de type KWIC (*key words in context*), est cependant essentiel pour ébaucher des hypothèses sur le fonctionnement du discours et le positionnement des locuteurs d'après leur profil social.

Ce va-et-vient entre l'analyse lexicale et les énoncés permet en effet d'inscrire les unités lexicales dans des systèmes de catégories sémantiques et énonciatives susceptibles de traduire, dans le discours même, ce que l'on cherche à comprendre, à savoir ici l'attitude des jeunes par rapport au tabagisme et l'influence de messages publicitaires dissuasifs. La catégorisation vise donc à établir le pont entre la problématique de recherche et les données textuelles. Elle peut rappeler la procédure de codage de l'analyse qualitative à cette différence qu'elle s'appuie sur des procédures d'analyse lexicale qui permettent de tenir compte de l'ensemble des données et sur l'examen de phénomènes discursifs difficilement repérables par une simple lecture linéaire.

La reprise des analyses univariées et multivariées sur les données catégorisées s'inscrit dans les procédures de validation des hypothèses interprétatives. La première approche, inductive, qui implique à la fois une sensibilité aux procédés linguistiques et à la problématique de la recherche, sera donc relayée par une approche hypothético-déductive.

L'analyse de distance permet de déterminer le vocabulaire qui caractérise un sous-texte, c'est-à-dire les formes qui marquent l'originalité du vocabulaire d'une partie du corpus par rapport à l'autre. Dans le tableau qui suit (tableau I, 3.5.2b), les mots qui caractérisent le plus le discours avant l'introduction de la brochure sont suivis d'un astérisque. Les mots sans astérisque caractérisent davantage les propos tenus après l'introduction de la brochure.



Analyse de distance sur les formes lexicales brutes avant/après l'introduction de la brochure (3.5.2b, tableau I)

	*				
Fréqtot	**A**	**B**	**explique**	**cumul**	
0.07	0.14	0.02	0.44	0.44	clair *
0.23	0.38	0.18	0.40	0.84	aussi *
0.05	0.11	0.02	0.31	1.15	plaisir *
0.06	0.11	0.02	0.31	1.46	dépendance *
0.02	0.00	0.05	0.28	1.75	témoignage
0.09	0.04	0.15	0.28	2.02	"
0.01	0.03	0.00	0.26	2.28	3ème *
0.02	0.05	0.00	0.25	2.54	doigts *
0.06	0.01	0.09	0.24	2.78	risques

0.02	0.05	0.00	0.24	3.02	primaire *
0.37	0.45	0.25	0.24	3.25	ils *
0.59	0.62	0.87	0.23	3.49	j'
0.03	0.01	0.06	0.23	3.72	concret
0.01	0.00	0.04	0.23	3.95	cinq
0.09	0.13	0.04	0.22	4.17	santé *
0.02	0.00	0.04	0.21	4.38	solution
0.02	0.04	0.00	0.20	4.58	appelle *
0.02	0.00	0.05	0.20	4.78	chiffres
0.03	0.01	0.06	0.20	4.98	routière
0.01	0.03	0.00	0.19	5.17	choqué *
0.01	0.03	0.00	0.19	5.37	influencé *
0.01	0.03	0.00	0.19	5.56	dents *
0.15	0.10	0.21	0.19	5.74	elle
0.36	0.36	0.53	0.18	5.93	!
0.01	0.00	0.03	0.18	6.11	morts
0.03	0.02	0.06	0.18	6.29	y
0.02	0.04	0.00	0.18	6.47	dérange *
0.07	0.04	0.11	0.18	6.65	cela
0.28	0.17	0.32	0.18	6.83	te
0.02	0.00	0.04	0.18	7.00	image
0.02	0.00	0.04	0.18	7.18	provoque
0.02	0.04	0.00	0.17	7.35	odeur *
0.04	0.09	0.03	0.17	7.52	effectivement *
0.02	0.04	0.00	0.17	7.69	jaunes *
0.16	0.17	0.06	0.17	7.86	toi *
0.06	0.10	0.03	0.17	8.03	niveau *
0.01	0.03	0.00	0.17	8.20	publicitaire *
0.02	0.00	0.04	0.17	8.36	long

Si on s'attarde aux items lexicaux pleins, c'est-à-dire aux noms, adjectifs et aux verbes, il semble que les mots qui décrivent l'apparence physique et la santé en général sont ceux qui caractérisent le plus le vocabulaire avant l'introduction de la brochure *clair, doigts, dents,*

santé sans oublier la notion de *plaisir* et de *dépendance* ; après l'introduction de la brochure, il est remarquable de constater que les mots *témoignage, concret, solution, chiffres, mort* sont ceux qui apparaissent en tête de liste. Il appert que les deux sous textes ne font pas ressortir les effets du tabac dans les mêmes termes. Avant on parle de plaisir et des effets néfastes sur la santé et particulièrement sur l'apparence physique, les dents et les doigts jaunes et sur la dépendance. Après, les effets font toujours partie du discours, mais alors non plus en termes de plaisir, mais en termes de risque et de mort. Notons par ailleurs la présence importante du pronom *j'* après l'introduction de la brochure. Ceci pourrait suggérer que la brochure provoque une plus grande implication personnelle des sujets.

Pour valider ces observations, la première stratégie de vérification consiste à parcourir rapidement les contextes. Ainsi, on a constaté que le mot *clair* n'a rien à voir avec l'apparence, mais est plutôt utilisé comme marque évaluative : *C'est clair, c'est évident*.

Dans un deuxième temps, on s'est donné une procédure de vérification des hypothèses d'interprétation construites à partir des premiers résultats obtenus lors de l'analyse de distance sur les unités lexicales brutes. C'est ici qu'entre en jeu la catégorisation des unités lexicales. Il s'agit de déterminer avec plus de raffinement les thèmes traités au cours des discussions, toujours dans la perspective de déterminer si l'introduction de la brochure produit des changements dans le discours des participants.

On a donc introduit une propriété lexicale que nous avons nommée *thème*. Cette propriété englobe en fait plusieurs grilles d'analyse qui auraient pu être regroupées dans des propriétés différentes. Mais dans l'idée de procéder de façon itérative par raffinements successifs de nos procédures, il était suffisant, à ce stade-ci, de n'avoir qu'un système de catégories dont certaines peuvent déjà faire l'objet d'une structuration plus fine. Ainsi, seront rajoutées à la grille cinq catégories (*Soc-X*) associées aux formes lexicales décrivant l'environnement social. Pourquoi? On avait remarqué, au cours de l'analyse de distance, que les items lexicaux qui faisaient référence à l'aspect social du tabagisme, c'est-à-dire de ses conséquences sur les rapports sociaux des jeunes, ressortaient beaucoup. On a donc établi en suffixe une échelle décrivant le niveau de l'environnement social, allant du plus intime au plus général : *soc-je, soc-ami, soc-famille, soc-jeune, soc-gens*.

Voici le descriptif de la propriété thème. On a déterminé 28 catégories *: apparence, arrêt, négation, concret, danger, dépendance, soc-je, maladie, mort, plaisir, publicité, tabac,*

nicotine, drogue, interdiction, fumeur, soc-ami, soc-famille, soc-gens, liberté, envie, conscience, volonté, soc-jeune, coûts, début, santé, éducation, prévention.

La procédure de catégorisation procède des mots caractéristiques révélés par l'algorithme de distance, vers l'ensemble du vocabulaire. Après ces mots, on a examiné de façon systématique les mots fréquents et complété la catégorisation en examinant le lexique trié par ordre alphabétique pour catégoriser les variantes flexionnelles pertinentes. Pour confirmer nos intuitions, on a repris les analyses statistiques sur les valeurs de la propriété *thème* comme l'illustre le tableau II (3.5.2c) qui suit.



Analyse de distance sur les valeurs de la propriété thème avant/après l'introduction de la brochure (3.5.2c, tableau II)

	*				
Fréqtot	**A**	**B**	**explique**	**cumul**	
0.21	0.43	0.11	31.23	31.23	apparence *
0.09	0.02	0.16	13.85	45.08	concret
0.08	0.14	0.05	6.75	51.83	plaisir *
0.13	0.21	0.10	6.63	58.46	dépendance *
0.14	0.19	0.08	5.64	64.10	santé *
0.11	0.17	0.08	5.39	69.49	éducation *
0.18	0.11	0.22	5.12	74.61	volonté
0.10	0.08	0.17	4.75	79.36	mort
1.95	2.19	1.82	4.53	83.89	tabac *
0.05	0.10	0.05	3.26	87.15	soc-ami *
0.17	0.25	0.16	3.12	90.27	coûts *
0.32	0.28	0.40	2.81	93.09	maladie
0.75	0.59	0.72	1.44	94.53	publicité
0.21	0.26	0.20	1.37	95.90	soc-famille *
0.11	0.14	0.11	0.84	96.74	drogue *
0.20	0.22	0.17	0.82	97.55	liberté *
0.74	0.69	0.78	0.67	98.23	soc-gens

0.17	0.14	0.18	0.66	98.89	envie
0.31	0.29	0.24	0.54	99.44	soc-jeune *
0.05	0.08	0.06	0.19	99.62	nicotine *
0.63	0.60	0.64	0.18	99.81	arrêt
0.08	0.07	0.08	0.09	99.90	conscience
0.24	0.17	0.19	0.03	99.93	danger
0.22	0.17	0.16	0.03	99.96	début *
0.13	0.11	0.12	0.02	99.98	prévention
2.28	2.48	2.50	0.01	99.99	négation
0.48	0.44	0.45	0.01	100.00	fumeur
2.14	2.54	2.53	0.00	100.00	soc-je *

La notion d'*apparence* se confirme. Les sujets abordés, avant la brochure, concernent les effets superficiels du tabagisme, à savoir, la couleur des dents et des doigts, le teint, l'odeur des vêtements et des cheveux… La notion de plaisir ressort aussi comme thème avant l'introduction de la brochure, ainsi que les notions de dépendance, santé et éducation.

Après l'introduction de la brochure, la catégorie *concret* (impact et solutions) ressort. On voit aussi émerger les notions de *volonté*, *mort* et *maladie*. D'autre part, on constate que l'hypothèse sur les pronoms personnels de la première personne (*soc-je*) ne se confirme pas. L'écart observé avant et après la brochure était spécifique à la forme *j'*.

On peut affiner l'analyse en comparant les interventions avant et après l'introduction de la brochure selon le profil sociologique des sujets. Ainsi, en comparant les données qui suivent (tableau III, 3.5d), il est remarquable de constater que ce sont les non-fumeurs qui semblent le plus touchés par la brochure comme en témoigne la dominance des thèmes relatifs aux effets négatifs du tabagisme : *maladie* et *mort*.

Analyse de distance avant/après pour les fumeurs vs. les non-fumeurs (3.5.2d, tableau III)

Fumeurs						Non-fumeurs					
Fréqtot	Afu	Bfu	explique	cumul		Fréqtot	Anf	Bnf	explique	cumul	
0.21	0.47	0.11	32.55	32.55	apparence*	0.75	0.42	0.99	15.22	15.22	publicité
0.09	0.03	0.19	18.05	50.60	concret	0.21	0.38	0.10	12.52	27.74	apparence*
0.11	0.20	0.06	10.15	60.75	éducation*	0.05	0.14	0.02	10.64	38.38	soc-ami*
0.13	0.20	0.08	6.26	67.01	dépendance*	0.17	0.37	0.15	9.73	48.11	coûts*
0.18	0.07	0.21	5.79	72.79	volonté	0.14	0.20	0.03	8.21	56.32	santé*
0.08	0.15	0.06	4.85	77.64	plaisir*	0.32	0.29	0.54	6.94	63.26	maladie
0.48	0.35	0.53	3.96	81.60	fumeur	0.10	0.06	0.19	5.96	69.22	mort
0.17	0.15	0.25	3.77	85.37	envie	0.08	0.14	0.04	4.51	73.72	plaisir*
0.75	0.71	0.51	2.70	88.07	publicité*	1.95	2.32	1.82	4.38	78.11	tabac*
1.95	2.09	1.81	2.23	90.30	tabac*	0.09	0.01	0.11	4.01	82.11	concret
0.21	0.28	0.20	1.79	92.08	soc-famille*	0.48	0.57	0.34	4.00	86.12	fumeur*
0.10	0.09	0.15	1.74	93.83	mort	0.13	0.24	0.13	3.15	89.27	dépendance*
0.14	0.18	0.12	1.60	95.43	santé*	0.11	0.19	0.11	2.32	91.59	drogue*
0.22	0.18	0.12	0.99	96.42	début*	0.20	0.20	0.13	1.09	92.68	liberté*
0.63	0.57	0.67	0.98	97.41	arrêt	0.05	0.08	0.04	1.05	93.73	nicotine*
2.14	2.74	2.93	0.91	98.32	soc-je	2.14	2.25	2.01	0.93	94.66	soc-je*

Une première analyse des résultats obtenus nous a amenés à conclure que les hommes semblaient plus interpellés par l'introduction de la brochure que les femmes. Ce résultat confirme-t-il les autres analyses sur les effets des campagnes contre le tabagisme, à savoir que les femmes sont moins touchées que les hommes? Une réponse positive, à cette étape de l'analyse, serait prématurée.

Une autre façon de visualiser les résultats nous est donnée par l'analyseur PARTICIPATION. Pour une catégorie donnée, l'analyseur calcule sa fréquence relative dans les divers sous-textes, ce qui permet de voir si les variables sociologiques ont une influence sur l'utilisation des mots catégorisés. Les tableaux IV et V illustrent la distribution des catégories *apparence*

et *mort* dans le corpus complet et divers sous-textes. **A** et **B** désignent *avant* et *après* la brochure. Les particules **fu** et **nf** sont utilisée pour *fumeur* et *non-fumeur*, ainsi que **ho** et **fe** pour *homme* et *femme*.



Analyseur PARTICIPATION (thème=apparence) (3.5.2e, tableau IV)

Propriété	Couverture	Lexèmes	Occurrences	Cote Z
Fréqtot	78703/78703(100.00%)	37/3985 (0.93%)	168/78703 (0.21%)	0.00
A	23544/78703 (29.91%)	30/2087 (1.44%)	101/23544 (0.43%)	7.17
B	28074/78703 (35.67%)	18/2351 (0.77%)	30/28074 (0.11%)	-3.87
Afu	13758/78703 (17.48%)	24/1580 (1.52%)	64/13758 (0.47%)	6.40
Bfu	15923/78703 (20.23%)	13/1749 (0.74%)	18/15923 (0.11%)	-2.75
Anf	9786/78703 (12.43%)	19/1240 (1.53%)	37/9786 (0.38%)	3.53
Bnf	11898/78703 (15.12%)	8/1425 (0.56%)	12/11898 (0.10%)	-2.66
Aho	14468/78703 (18.38%)	16/163 (4 0.98%)	44/14468 (0.30%)	2.36
Bho	16010/78703 (20.34%)	11/1797 (0.61%)	19/16010 (0.12%)	-2.60
Afe	9076/78703 (11.53%)	24/1153 (2.08%)	57/9076 (0.63%)	8.56
Bfe	11811/78703 (15.01%)	9/1379 (0.65%)	11/11811 (0.09%)	-2.83



Analyseur PARTICIPATION (thème=mort) (3.5.2f, tableau V)

Propriété	Couverture	Lexèmes	Occurrences	Cote Z
Fréqtot	78703/78703 (100.00%)	9/3985 (0.23%)	80/78703 (0.10%)	0.00
A	23544/78703 (29.91%)	4/2087 (0.19%)	19/235440 (0.8%)	-1.01
B	28074/7870335 (67%)	6/2351 (0.26%)	47/28074 (0.17%)	3.46
Afu	13758/78703 (17.48%)	4/1580 (0.25%)	13/13758 (0.09%)	-0.26
Bfu	15923/78703 (20.23%)	6/17490.(34%)	24/15923 (0.15%)	1.94
Anf	9786/7870312.(43%)	2/1240 (0.16%)	6/9786 (0.06%)	-1.25
Bnf	11898/78703 (15.12%)	3/1425 (0.21%)	23/11898 (0.19%)	3.14
Aho	14468/78703 (18.38%)	4/1634 (0.24%)	8/14468 (0.06%)	-1.75
Bho	16010/78703 (20.34%)	4/1797 (0.22 %)	21/16010 (0.13%)	1.17
Afe	9076/78703 (11.53%)	2/1153 (0.17%)	11/9076 (0.12%)	0.58
Bfe	11811/78703 (15.01%)	5/1379 (0.36%)	26/1181 (0.22%)	11-04-04

Ces quelques exemples illustrent une approche permettant de construire un protocole d'analyse de corpus qui soit à la fois transparent et respectueux de la spécificité du contexte d'énonciation, ici des échanges oraux analysés sous forme de transcriptions. Il s'agit d'une démarche itérative qui combine l'approche inductive, souvent associée aux méthodes qualitatives, l'utilisation d'outils simples de statistique lexicale, et une approche plus sensible à la pragmatique textuelle. Ce traitement textuel des données a aussi l'avantage de produire des données qualifiées qui traduisent la démarche interprétative de l'analyste.

3.5.3 Analyse exploratoire d'entrevues de groupe : quand ALCESTE, DTM, LEXICO et SATO se donnent la main

L'article *Analyse exploratoire d'entrevues de groupe : quand ALCESTE, DTM, LEXICO et SATO se donnent la main* (Daoust et coll. 2006) reprend le corpus d'entretiens avec des groupes de jeunes sur l'usage du tabac et l'influence de messages contre l'usage du tabac. d'entrevues de groupe sur l'usage du tabac pour montrer comment on peut combiner plusieurs logiciels de lexicométrie (ALCESTE, DTM et LEXICO 3) pour valider et compléter l'analyse avec SATO.

La première analyse faisait appel à la construction d'un système de catégories lexicales basé sur une démarche itérative qui contrastait les données partitionnées selon diverses combinaisons de variables sociologiques décrivant les participants et le moment de leurs interventions. L'objectif de la démarche était donc de s'appuyer sur le profil sociologique des sujets afin de proposer une grille catégorielle permettant de comprendre, à travers les interventions de chacun, l'influence de messages d'avertissement sur l'usage du tabac.

Dans le cadre des groupes de travail du JADT et du réseau ATONET (Duchastel et al 2005), on a voulu voulu valider cette démarche par l'utilisation combinée de divers logiciels de statistique textuelle : ALCESTE (Reinert), DTM (Lebart), LEXICO 3 (Salem) en plus de SATO (Daoust).

La transcription des entrevues illustrée dans le tableau 3.5.2a constitue une forme de balisage *pré-XML* facile à lire pour un humain, mais n'offrant pas l'universalité des formats d'échange

normés. La plupart des logiciels d'analyse textuelle utilisent un *format propriétaire* impliquant un travail de conversion non trivial pour faire circuler les données d'un logiciel à l'autre. C'est dans ce contexte que plusieurs développeurs de logiciels se sont réunis pour élaborer une stratégie de conversion des formats de données entre les logiciels. Cette proposition, qui fait l'objet d'une communication aux JADT 2006 (Daoust et Marcoux 2005), est basée sur un format pivot en XML développé à partir de recommandations de la *Text Encoding Initiative.* Des programmes *Perl* (passerelles) permettent de convertir les données des formats propriétaires vers le format XML-TEI et du format XML-TEI vers les formats propriétaires.

Pour cette deuxième phase de l'analyse, on a utilisé le corpus en format SATO et nous avons eu recours au logiciel pour reconfigurer les données aux fins d'exploitation dans les logiciels ALCESTE, LEXICO et DTM. Ces nouvelles versions du corpus, après exportation par SATO en format XML-TEI, seront converties par les passerelles PERL vers les formats propriétaires ALCESTE, LEXICO et DTM. Voici la liste de ces diverses reconfigurations du corpus.

1. Corpus *Initial*. Ce corpus contient la transcription des entrevues dans leur découpage original en interventions. Pour l'analyse, on exclut les interventions des intervenants.

2. Corpus *Avant*. Ce corpus contient la partie du corpus *Initial* précédant l'exposition au message contre l'usage du tabac.

3. Corpus *Après*. Ce corpus contient la partie du corpus *Initial* suivant l'exposition au message contre l'usage du tabac.

4. Corpus P*articipant*. Ce corpus est le résultat d'une reconfiguration du corpus original. Il rassemble de manière continue l'ensemble des interventions de chaque participant identifié par un nom résumant son profil et suffixé par *a* ou *b* pour identifier le discours du participant *avant* et *après* le message contre l'usage du tabac. Il est à noter que nous avons éliminé du corpus les participants dont le profil sociologique est incomplet afin de concentrer l'analyse sur les variables principales.

5. Participant *catégorisé*. Ce corpus reprend les données du corpus *Participant,* mais dans lequel les unités lexicales thématisées au cours de l'analyse SATO sont remplacées par la valeur de la propriété catégorielle *thème*. Les mots non thématisés restent inchangés.

6. Participant *réduit*. Ce corpus reprend les données du corpus *Participant,* mais en remplaçant tous les mots par les catégories qui leur ont été attribuées lors de l'analyse avec SATO. Les mots qui n'avaient pas été catégorisés seront remplacés par la catégorie vide *x*.

Le principe de base de la démarche de 2004 consistait, rappelons-le, à comparer, avec des indices statistiques simples, les lexiques associés à des sous-textes découpés d'après les variables établies au départ : sexe, fumeur/non-fumeur, avant/après le message contre l'usage du tabac. Cette comparaison, appliquée au lexique brut, a fourni les premiers indices pour construire un lexique catégorisé reflétant les points d'ancrage de notre chaine interprétative.

Pour comparer les lexiques, on a utilisé une mesure de distance lexicale basé sur la *distance du Chi2*. La mesure évalue l'écart dans l'utilisation d'un vocabulaire donné entre deux sous-ensembles du corpus. Il ne s'agit pas d'un test statistique, mais simplement de l'utilisation d'une métrique appliquée à l'espace ou à un sous-espace lexical. Les formes lexicales sont triées par ordre décroissant de contribution à la mesure de distance, ce qui permet d'identifier, par ordre d'importance, les spécificités de chaque sous-texte. L'algorithme peut être appliqué aux formes lexicales elles-mêmes ou aux valeurs de propriétés correspondant à la catégorisation lexicale. L'approche est essentiellement dichotomique : on compare un sous-texte à un autre, via leur lexique respectif. On a aussi eu recours à un *algorithme de participation* qui calcule les moyennes normalisées d'un ensemble de formes lexicales, correspondant généralement à une catégorie lexicale, pour chacun des sous-textes constitués en cours d'analyse.

Cette démarche exploratoire se fondait sur un va-et-vient interactif entre ce que révélait l'analyse lexicale et les contextes d'utilisation des mots mis en évidence par les algorithmes de distance et de participation. Au premier niveau de l'analyse, on a travaillé sur les données brutes, c'est-à-dire les formes lexicales elles-mêmes. Le recours aux formes marquées donne la possibilité de voir apparaitre des différenciations portées par la morphologie des mots en termes de nombre, genre, personne, temps. De plus, on s'est intéressé tout autant, sinon davantage, à des marqueurs d'énonciation, comme les pronoms personnels, les marqueurs phatiques, les marques de la négation, de l'interrogation, les verbes épistémiques (croire, penser...), etc. qu'aux mots sémantiquement pleins. C'est en s'appuyant sur l'analyse lexicale de ces données brutes qu'on a élaboré la grille catégorielle sémantique.

Le retour constant aux énoncés, ne serait-ce que par un parcours rapide des contextes courts de type KWIC, s'est avéré essentiel pour ébaucher des hypothèses et inscrire les unités lexicales dans des systèmes de catégories sémantiques et énonciatives susceptibles de traduire, dans le discours même, ce que nous cherchions à comprendre, à savoir l'attitude des jeunes par rapport au tabagisme et l'influence de publicités dissuasives. La catégorisation visait donc à établir le pont entre la problématique de recherche et nos données textuelles. Elle correspondait un peu à la procédure de codage de l'analyse qualitative à cette différence qu'elle s'appuyait sur des procédures d'analyse lexicale qui permettent de tenir compte de l'ensemble des données et sur l'examen de phénomènes discursifs difficilement repérables par une simple lecture linéaire.

La procédure de catégorisation procédait des mots caractéristiques révélés par l'algorithme de distance vers l'ensemble du vocabulaire. Après les mots caractéristiques, on a examiné de façon systématique les mots fréquents et complété la catégorisation en examinant le lexique trié par ordre alphabétique pour catégoriser les variantes flexionnelles pertinentes. Pour confirmer nos intuitions, nous avons repris nos analyses de distance et de participation en les appliquant cette fois sur les valeurs de la propriété *thème*.

Lors de la reprise du corpus en 2006, on a décidé, avec l'utilisation du logiciel ALCESTE (Reinert 2002), de prendre, dans un premier temps, le contrepied de notre approche de 2004 en ayant recours à une méthode de classification qui ignore, dans la construction des classes, les *variables externes*, c'est-à-dire, dans notre cas, les profils sociologiques des locuteurs, l'impact du message anti-tabac, etc.

Il est intéressant de noter qu'ALCESTE , qui s'inscrit dans la tradition française d'analyse de données initiée par Benzécri (1973, 1981), répond aussi aux préoccupations de l'analyse de discours en psychologie clinique. Les deux approches s'appuient donc sur la notion de discours : davantage dans une perspective sociologique avec SATO, et davantage avec une perspective psychanalytique et sémiotique avec ALCESTE.

D'un certain point de vue, la méthode ALCESTE se situe à l'opposé de celle qu'on a utilisée avec SATO pour procéder à la construction itérative d'une grille catégorielle. D'abord, contrairement à SATO, ALCESTE propose une méthode complètement automatique qui vise à faire émerger des *mondes lexicaux*. Pour ce, ALCESTE construit des *énoncés simples* dont l'approximation statistique correspond à des segments de texte de longueur comparable

respectant les frontières de l'*unité de contexte élémentaire* (UCE), généralement la phrase. Ainsi, ALCESTE tente de faire émerger la structure du discours par le dépistage de profils de répétition dans les énoncés simples, alors qu'avec SATO, nous sommes partis d'hypothèses structurantes du discours pour *faire parler les données.*

Pour l'analyse du corpus avec ALCESTE, nous avons procédé à quatre expérimentations à partir des premières configurations du corpus décrites à la section.2 : corpus *Initial,* corpus *Avant*, corpus *Après et* corpus *Participant*.

ALCESTE a produit 2042 UCE sur le corpus *Initial*, réparties en deux classes qui comptent respectivement environ le tiers et les deux tiers des UCE. La première classe est fortement caractérisée par des UCE provenant des interventions exprimées après l'exposition au message contre l'usage du tabac (Chi2=33.82). On trouve aussi, mais plus faiblement, une présence significative des UCE des non-fumeurs. La deuxième classe est fortement caractérisée par les UCE provenant des interventions précédant la présentation du message contre l'usage du tabac (Chi2=33.82). On trouve aussi, mais plus faiblement, une présence significative des UCE des fumeurs (Chi2=8.81). Le tableau qui suit montre le vocabulaire le plus caractéristique de ces deux classes.

Classes produites par ALCESTE sur le corpus Initial (3.5.3a, tableau)

Formes représentatives de la classe n°1

Chi2	u.c.e. dans la classe	Formes réduites
100.00	51	cancer+
93.85	38	image+
83.51	31	choc+
82.20	38	poumon+
81.60	35	choqu+er
73.64	42	preventi+f
61.71	23	routier+
53.58	20	temoign+23
53.47	107	voir.
50.88	19	tele
49.69	39	pub+
46.79	22	femme+
45.39	24	mort+
42.83	23	mourir.
42.16	46	tabac+

Formes représentatives de la classe n°2

Chi2	u.c.e. dans la classe	Formes réduites
102.21	446	fum+er
68.65	233	arret+er
28.50	95	commenc+er
28.44	170	fum+eur
22.54	64	essa+yer
21.46	87	envi+e
20.22	69	arrete+
19.36	108	cigarette+
17.11	61	paquet+
16.34	64	volonte+
16.04	68	prendre.

À la lumière de ces résultats, il nous a été possible d'interpréter les classes crées par ALCESTE, comme on l'a fait pour le tableau de distance de SATO, en regroupant ce vocabulaire autour de plusieurs axes catégoriels. Cependant, à la différence de SATO, il ne sera pas possible de marquer cette interprétation par un balisage permettant une validation ultérieure.

La première classe fait ressortir des thèmes tels que la prise de conscience (*voir, choquer, choc, image, témoignage*), la mort et la maladie (*cancer, poumon, mort*), la médiatisation (*pub, télé, spot, prévention routière*). Les interventions après le message contre l'usage du tabac touchent des thèmes plus graves et marquent une réaction par rapport aux campagnes de

publicité. Ce discours est davantage exprimé par les non-fumeurs. Pour la deuxième classe, on voit des verbes et des noms qui semblent renvoyer directement à la consommation de cigarette en termes d'arrêt, d'envie et de volonté. Ce vocabulaire est surtout marqué par les fumeurs.

Mais, qu'ALCESTE confirme que la variable *avant/après* le message contre l'usage du tabac représente le premier élément de structuration du corpus constitue pour nous le résultat le plus significatif. Cette dominance est telle qu'elle empêche le repérage d'autres classes. Soulignons de plus la présence de l'opposition *fumeur/non-fumeur* qui est la deuxième variable prise en compte dans l'analyse SATO.

Pour neutraliser l'effet dominant du profil *avant/après,* on a présenté à ALCESTE deux autres corpus contenant respectivement les interventions avant et après le message contre l'usage du tabac.

Pour le corpus *Avant,* on obtient trois classes représentant respectivement 36 %, 48 % et 16 % des 674 UCE retenues. La première classe est marquée par une dominance des fumeurs, la seconde par les non-fumeurs et les hommes tandis que les femmes dominent la troisième. L'analyse du corpus *Après* crée deux classes représentant respectivement 28 % et 72 % des 1168 UCE retenues. La première classe est marquée par une dominance des non-fumeurs. La seconde par les fumeurs.

Enfin, nous avons soumis à ALCESTE le corpus *Participant.* Pour ce corpus, nous avons rassemblé toutes les interventions de chaque participant, ce qui peut avoir pour effet de réunir des énoncés interrompus dans le corpus d'origine. Mais surtout, nous avons éliminé les participants dont le profil sociologique n'était pas connu. C'est sans doute ce qui aura permis de contraster davantage les données de telle sorte qu'ALCESTE produit immédiatement trois classes rassemblant 64 %, 21 % et 15 % des 877 UCE retenues. La première classe est d'abord caractérisée par les énoncés avant le message contre l'usage du tabac et secondairement par les UCE des fumeurs. La deuxième classe est caractérisée par les non-fumeurs après le message contre l'usage du tabac tandis que la troisième classe est très caractéristique des énoncés après le message et teintée par le discours des femmes.

Signalons que le calcul des segments répétés, qu'on trouve aussi dans les autres logiciels statistiques, fait ressortir l'expression *sécurité routière* qui renvoie à une discussion sur une

campagne sur la sécurité routière. On aurait donc eu avantage à figer cette expression dans SATO pour la classer sous la catégorie *publicité.*

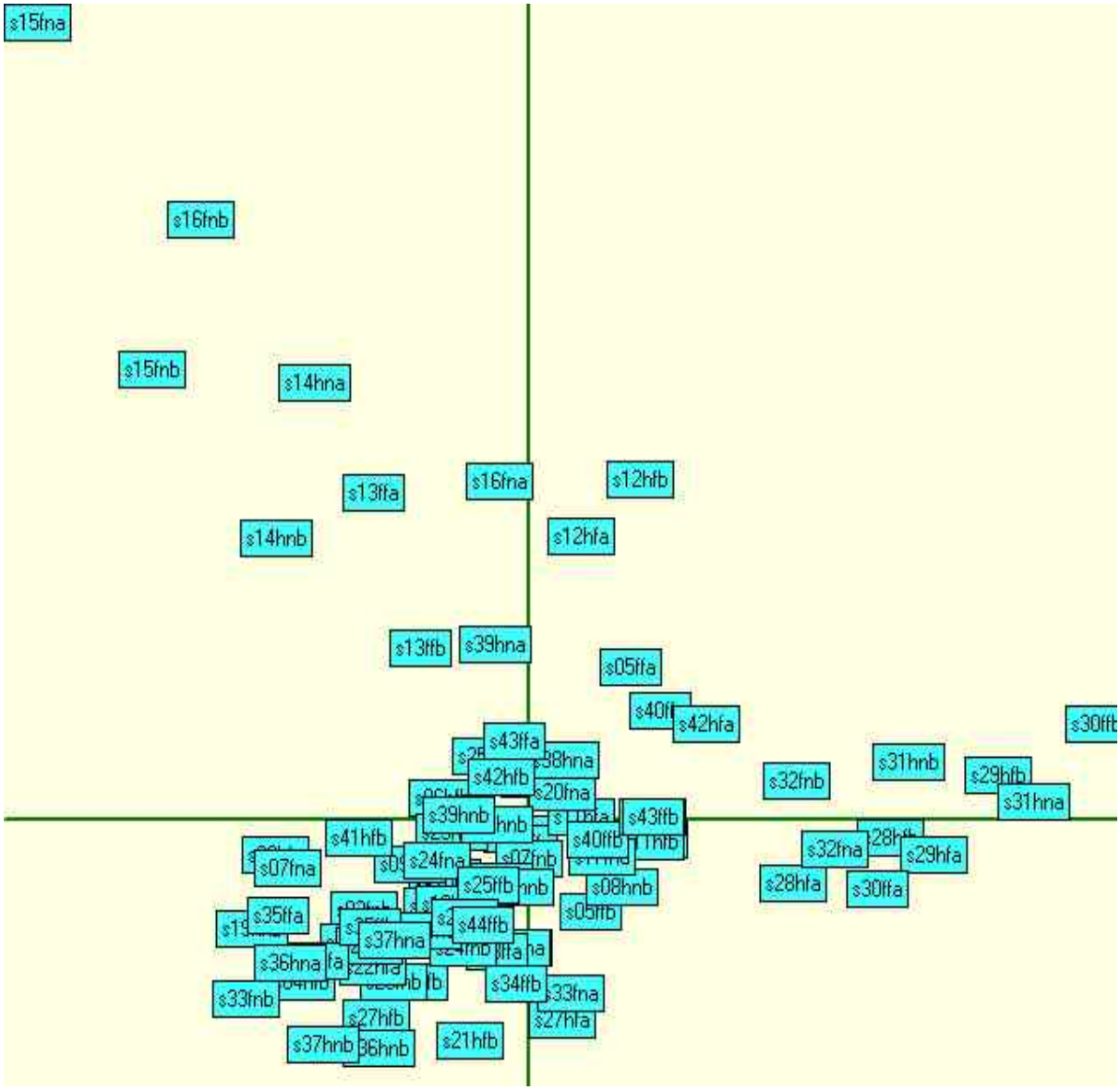
ALCESTE a donc retrouvé, à partir des énoncés simples, ce que nous avions observé lors de la comparaison de lexiques construits sur la base d'un découpage global du corpus sur la base de profils. Ce point de rencontre entre les approches ascendantes et descendantes est un outil important de validation de l'interprétation.

LEXICO (Salem et coll. 2003) permet de calculer les spécificités lexicométriques de parties d'un corpus d'après un modèle probabiliste basé sur la loi hypergéométrique (cf. Lebart & Salem 1994). Il rend aussi possibles des analyses factorielles des correspondances (AFC) sur un corpus partitionné. Les unités décomptées sont exclusivement constituées à partir de la liste des délimiteurs fournie par l'utilisateur (ponctuations et autres caractères non alphanumériques), sans recours à des ressources dictionnairiques extérieures.

Nous avons soumis à LEXICO le corpus *Participant* qui délimite le corpus en participants avant et après la présentation du message contre l'usage du tabac.

AFC produite par Lexico sur le corpus *Participant* (3.5.3b, figure)

(individus sur le plan des 2 premiers axes de l'AFC)



Comme on le constate, il est difficile de tirer des conclusions claires à partir de ce graphique qui situe les individus dans l'espace vectoriel des lexèmes. Certes, les noms des individus décrivent leur profil sociologique, mais on ne voit pas ici de patrons clairs. On reviendra sur ce type d'analyse avec DTM qui permet de tracer les modalités des variables catégorielles sur le plan de l'AFC. Avec LEXICO, on a plutôt utilisé le calcul des spécificités pour repérer les particularités associées à des groupes d'individus. On s'est demandé jusqu'à quel point une mesure probabiliste comme les spécificités pouvait converger ou s'écarter de la simple

métrique du Chi2 utilisée par l'analyseur DISTANCE de SATO. On a donc demandé à LEXICO de calculer les spécificités associées aux participants avant l'exposition au message contre l'usage du tabac. Nous avons reporté ces résultats sur la sortie de l'analyseur DISTANCE de SATO appliqué au lexique avant et après le message.



Comparaison entre les spécificités et la distance du Chi2 (3.5.3c, tableau)

Fréqtot	avant	après	explique	cumul	
0.08	0.15	0.03	0.55	0.55	**clair** * (lexico 6)
0.05	0.00	0.09	0.50	1.05	**brochure**
0.25	0.37	0.17	0.49	1.54	**aussi** * (lexico 6)
0.46	0.60	0.36	0.40	1.94	t' * (lexico 5)
0.07	0.12	0.03	0.39	2.33	**santé** * (lexico 5)
0.77	0.95	0.64	0.39	2.72	**ouais** * (lexico 3)
0.02	0.04	0.00	0.32	3.03	**appelle** * (lexico 4)
0.05	0.01	0.09	0.31	3.35	**risques** (lexico -5)
0.06	0.10	0.03	0.31	3.66	**dépendance** * (lexico 5)
0.06	0.10	0.03	0.31	3.96	**plaisir** * (lexico 5)
1.65	1.88	1.49	0.30	4.26	**je** * (lexico 3)
0.02	0.05	0.00	0.28	4.54	**doigts** * (lexico 4)
0.01	0.03	0.00	0.26	4.80	**odeur** * (lexico 4)
0.16	0.09	0.21	0.25	5.05	**elle** (lexico -5)
0.11	0.06	0.15	0.24	5.30	**beaucoup** (lexico -3)
0.03	0.00	0.05	0.24	5.53	**lire** (lexico -4)
0.13	0.18	0.09	0.23	5.76	**toi** * (lexico 4)
0.01	0.03	0.00	0.23	5.99	**3ème** * (lexico 3)
0.03	0.00	0.04	0.23	6.22	**témoignage**
0.05	0.09	0.03	0.22	6.44	**grave** * (lexico 3)
0.42	0.32	0.49	0.22	6.66	**!**
0.26	0.33	0.20	0.22	6.88	**ben** * (lexico 3)
0.08	0.04	0.11	0.21	7.09	"
0.61	0.49	0.69	0.21	7.30	**-**
0.44	0.34	0.51	0.21	7.50	**peut** (lexico -3)
0.02	0.03	0.00	0.20	7.70	**caractère** * (lexico 3)
0.28	0.20	0.34	0.19	7.90	**te** (lexico -4)
0.27	0.34	0.21	0.19	8.09	**suis** * (lexico 3)
0.02	0.04	0.01	0.18	8.26	**jaunes** * (lexico 3)
0.01	0.02	0.00	0.17	8.44	**choqué** *(lexico 3)
0.01	0.02	0.00	0.17	8.61	**conséquence** * (lexico 3)
0.01	0.02	0.00	0.17	8.78	**influencé** * (lexico 3)
0.01	0.02	0.00	0.17	8.95	**net** * (lexico 3)
0.01	0.02	0.00	0.17	9.12	**publicitaire** * (lexico 3)
0.02	0.00	0.03	0.17	9.30	**solution**
0.01	0.03	0.00	0.17	9.46	**aimes** * (lexico 3)
…					

Ce tableau est constitué à partir d'une sortie de l'analyseur *distance* de SATO. La première colonne contient la fréquence d'une forme lexicale dans l'ensemble du corpus. Les deuxième et troisième colonnes indiquent la fréquence de la forme dans l'ensemble des interventions des participants avant et après le message contre l'usage du tabac. La colonne *explique* donne la contribution de la forme lexicale à la mesure de distance entre les parties *avant* et *après* le message contre l'usage du tabac. La colonne *cumul* contient la somme partielle de ces

contributions. Suit ensuite la forme lexicale elle-même à laquelle nous avons ajouté manuellement les calculs de spécificité de LEXICO accompagné d'un exposant qui rend compte du degré de signification de l'écart constaté. Un exposant négatif est la marque d'une sous-représentation significative de l'entrée lexicale.

On observe qu'il y a un très large recouvrement entre les formes lexicales qui contribuent le plus à la distance et les spécificités calculées par LEXICO. Parmi les mots manquants, il y a les ponctuations qui, apparemment, ne sont pas prises en compte par LEXICO, de même que les formes absentes dans le corpus Avant, mais que l'on retrouve dans les spécificités du corpus *Après : brochure* (9), *témoignage* (4), *solution* (4), *image* (3). On voit que la mesure de spécificité de LEXICO fournit un bon complément à la DISTANCE du Chi2 pour les formes fréquentes par l'ajout d'un indice de spécificité. Par ailleurs, DISTANCE peut prendre en compte les formes moins fréquentes, puisque la distance du Chi2 est utilisé comme étalon de mesure ne donnant pas lieu à un test statistique soumis à des contraintes de taille de l'échantillon. Dans l'ergonomie de SATO, il est aussi possible de catégoriser directement les formes repérées avant de poursuivre l'analyse sur la base de données ainsi qualifiées.

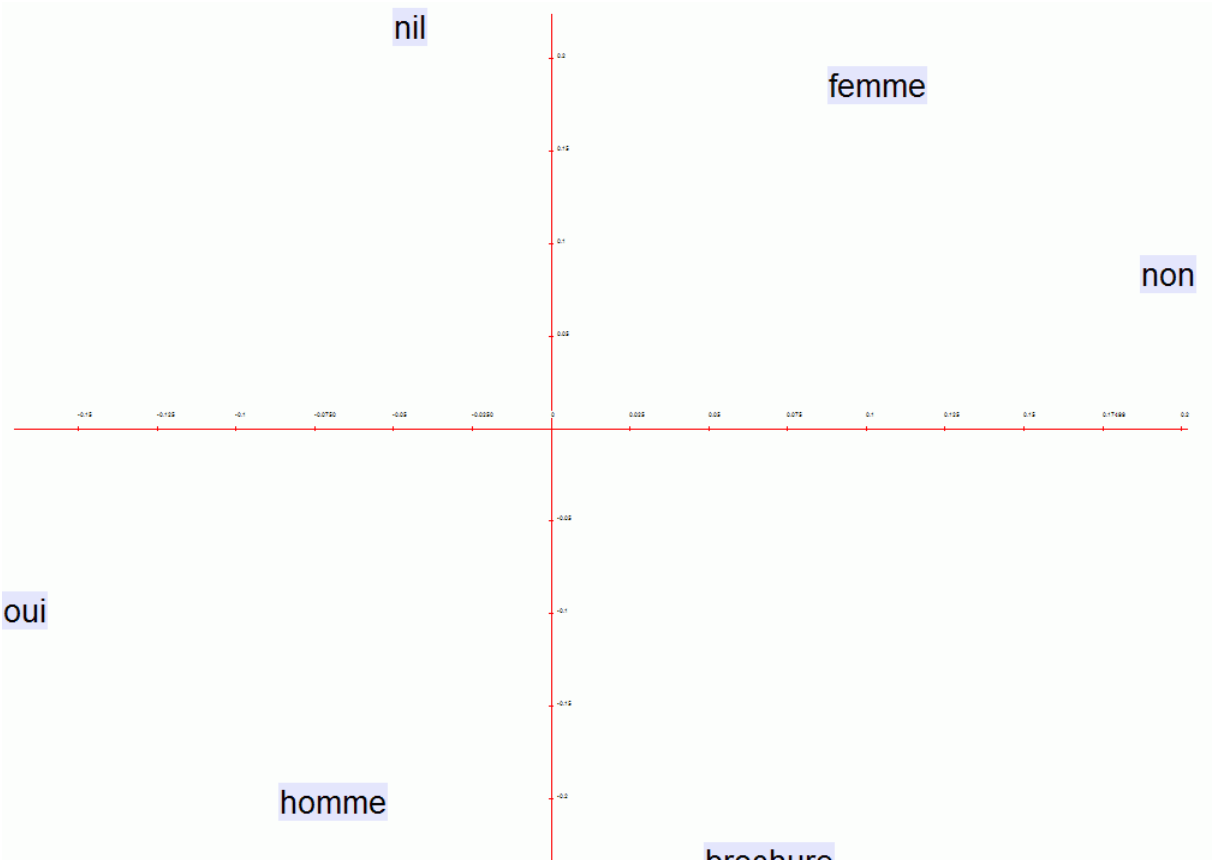
Le logiciel DTM (Lebart 2005) se présente comme un outil dédié à l'analyse exploratoire de données numériques multivariées et de données textuelles. L'exemple type de données admissibles au logiciel est la compilation de sondages comprenant à la fois des réponses à des questions fermées et à des questions ouvertes. Les questions fermées produisent soit des données directement numériques (poids, âge, etc.) ou des données catégorielles qui peuvent être codées par leur numéro d'ordre dans une liste fermée. Les réponses aux questions ouvertes produisent des données qualitatives, du texte brut utilisant des formes graphiques auxquelles on peut associer des variables comptant le nombre d'occurrences de la forme.

Nous avons utilisé ce modèle de couplage des questions ouvertes et fermées pour l'analyse du corpus P*articipant.* On considère le corpus comme un ensemble de textes rassemblant les interventions de chacun des 87 individus. Le profil sociologique est enregistré comme autant de réponses catégorielles à des questions fermées : pub (nil, brochure), sexe (homme, femme) et fumeur (non, oui). La question ouverte est unique et la *réponse* est composée de l'ensemble des interventions du participant, l'*avant* et l'*après* message contre l'usage du tabac étant considérés comme deux *questionnaires* distincts.

DTM procède à une analyse factorielle des correspondances croisant ces 87 individus et les 903 formes lexicales dont la fréquence est supérieure à quatre. Ensuite DTM représente les variables catégorielles dans l'espace de l'AFC. Les oppositions entre les diverses modalités de nos variables sociologiques apparaissent sur les trois premiers axes de l'AFC. Voici le plan constitué par les deux premiers axes.

AFC produite par DTM sur le corpus Participant (3.5.3d, figure)

(*v*ariables catégorielles sur le plan des 2 premiers axes de l'AFC)



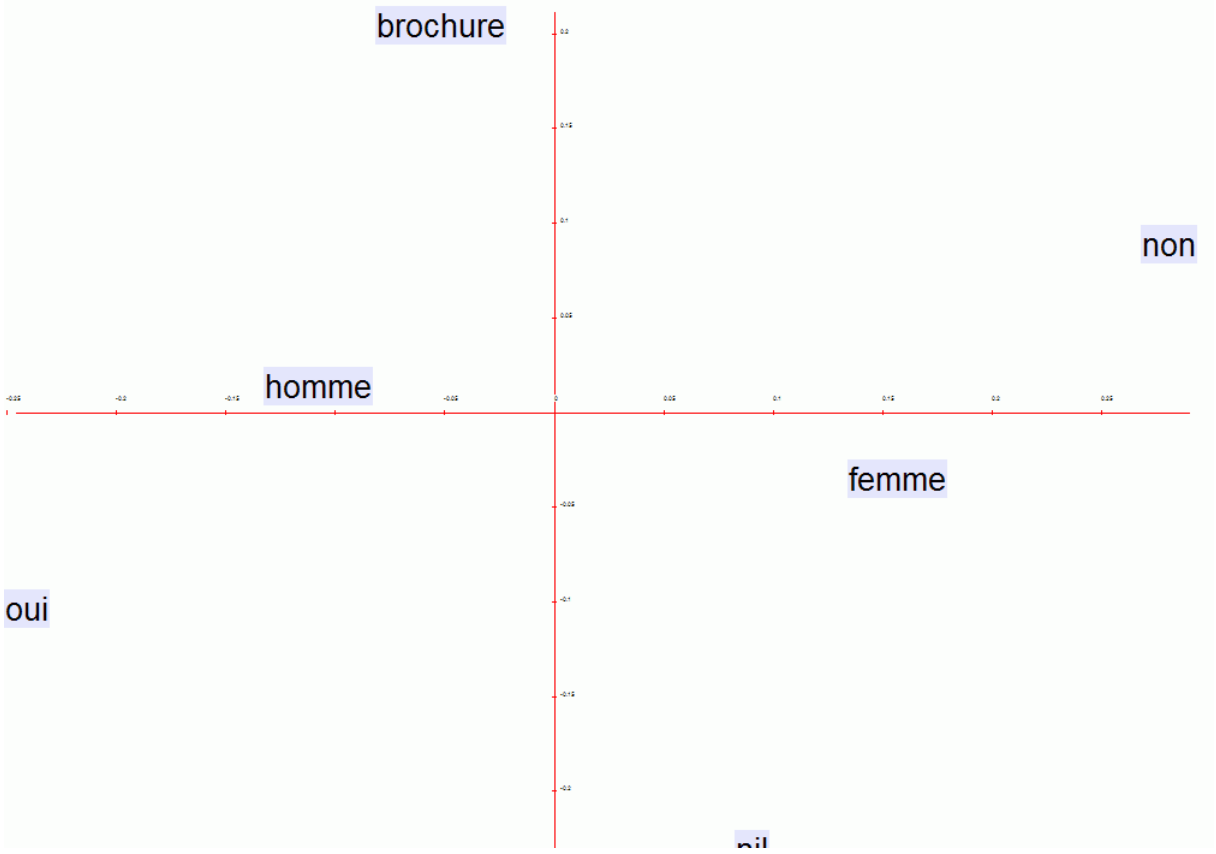
L'influence du message contre l'usage du tabac et de nos variables catégorielles sur la structure du discours semble confirmée. Mais, l'objectif de l'analyse catégorielle avec SATO va au-delà de cette constatation et vise, par la construction d'une grille de catégories lexicales, à interpréter les objets du discours. Si elle s'appuie dans un premier temps sur des mots saillants repérés par la distance du Chi2, la grille catégorielle s'élabore sur des bases sémantiques. On y écarte des unités lexicales jugées trop circonstancielles et on y ajoute d'autres unités contribuant à une de nos catégories socio sémantiques. Il n'est pas assuré que la classification établie d'après ces critères interprétatifs soit performante d'un point de vue

statistique, même si c'est ce que nous souhaitons. Le bon ajustement statistique sera considéré comme un critère de validation de notre grille.

Pour procéder à cette validation, on demandera à SATO de substituer la catégorie aux unités lexicales qu'elle décrit. Ce corpus artificiel, nommé *Participant catégorisé*, suivra la chaine de traitement habituel : exportation en XML-TEI avec sélection des variables pertinentes, conversion dans le format DTM en utilisant la passerelle ATONET et soumission à DTM. L'AFC est alors calculée en croisant les 87 participants avec 702 variables textuelles composées des formes lexicales non catégorisées et des catégories sémantiques remplaçant les formes lexicales catégorisées sémantiquement (propriété *thème*). Cette substitution recouvre 12,26 % des occurrences. Ici encore, comme on peut le voir, la projection des variables sociologiques sur le plan factoriel suit le même jeu d'oppositions.

AFC produite par DTM sur le corpus *Participant catégorisé* (3.5.3e, figure)

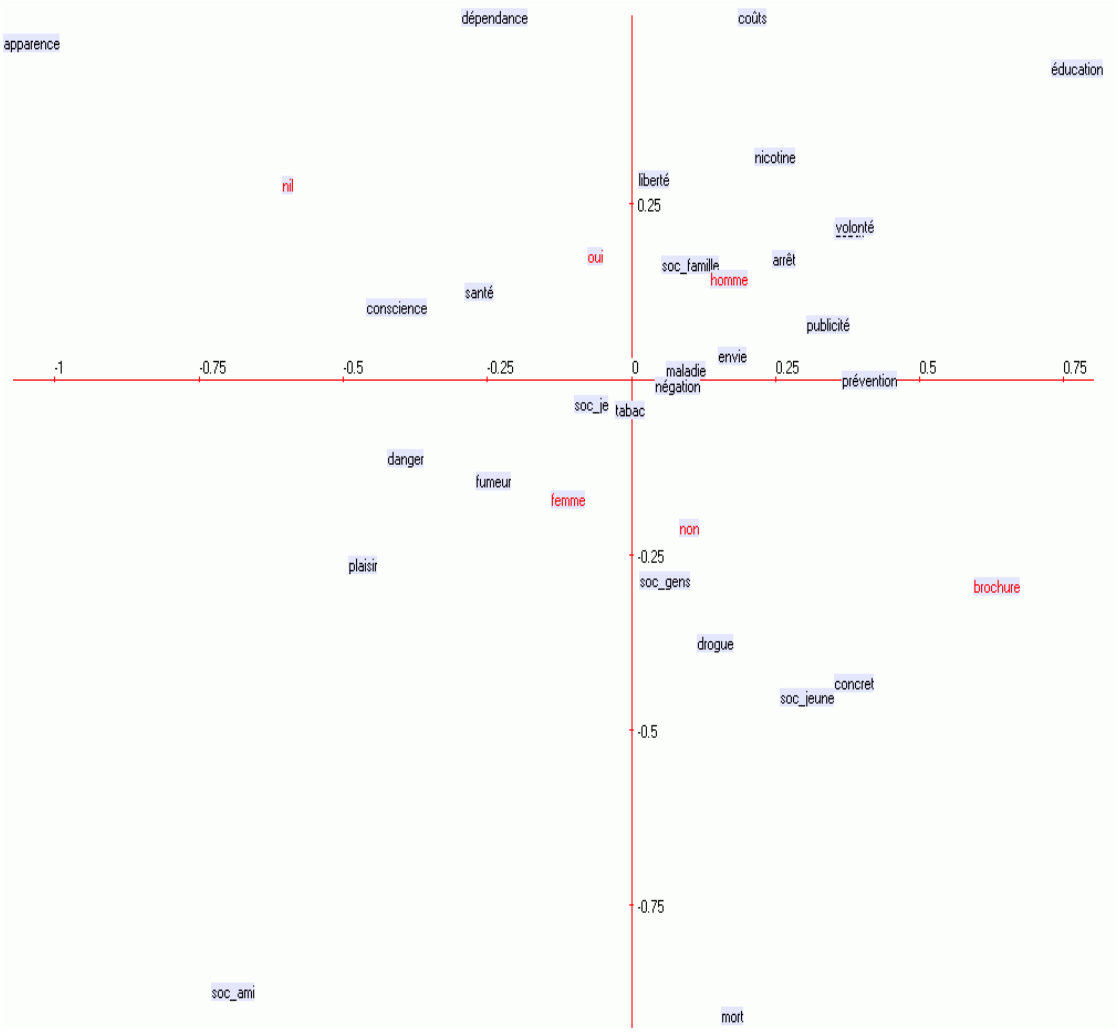
(*v*ariables catégorielles sur le plan des 2 premiers axes de l'AFC)



Nous ferons un pas de plus pour valider notre modèle catégoriel en réduisant toutes les formes lexicales qui ne font pas partie de la grille à une catégorie vide nommée arbitrairement *x*. Le corpus, que nous appellerons *Participant réduit*, contiendra encore le même nombre d'occurrences, mais avec un lexique de 29 unités lexicales seulement se substituant à l'ensemble des occurrences du corpus *Initial* sachant que les 28 catégories utiles représentent un peu plus de 12% des occurrences. Le corpus, ainsi réduit à notre grille socio-sémantique, permet-il toujours de faire les ressortir les variables externes? Voici le graphique.

AFC produite par DTM sur le corpus *Participant réduit* (3.5.3f, figure)

(*v*ariables catégorielles et lexique sur le plan des 2 premiers axes de l'AFC)



Comme l'espace des variables se réduit aux catégories, il est possible de visualiser correctement à la fois le lexique et les modalités des *questions fermées.* On dispose ainsi d'un très bel outil de validation de la construction de la grille de catégories lexicales. Présentée de cette façon, la visualisation des catégories sémantiques dans le plan factoriel ouvre aussi de nouvelles fenêtres d'investigation pour revenir aux contextes et affiner la grille si nécessaire. Il est quand même assez remarquable de voir s'étaler aux quatre points cardinaux les catégories les plus excentriques : *apparence, dépendance, coûts, éducation, mort* et *soc-ami*. À l'inverse, on voit apparaitre au centre du plan les catégories *banales* qui constituent les référents communs du discours.

Cette cartographie traduit l'intention même de l'analyse du discours qui vise à faire ressortir les positions sociales à l'intérieur même du langage. Allant au-delà de l'observation descriptive et du commentaire, la démarche illustrée ici montre comment l'interprétation peut s'appuyer sur des méthodologies transparentes et explicites.

Cette utilisation combinée de logiciels d'analyse textuelle augmente la fiabilité des conclusions en fournissant des moyens de corroborer ou d'infirmer des hypothèses et des conclusions. C'est ainsi qu'on peut aller au-delà des impressions et des commentaires descriptifs pour produire des représentations de discours sociaux susceptibles d'agir comme modèles. Cette analyse, à défaut d'illustrer toutes les caractéristiques de SATO, notamment l'héritage des propriétés, permet tout de même de mettre en évidence l'originalité du logiciel en tant que plateforme permettant d'enrichir et de reconfigurer les données pour constituer des protocoles expérimentaux transparents et ouverts à la diversité des méthodes.

Bibliographie du chapitre 3

Benzécri, 1973. Benzécri, J.-P. *L'Analyse des Données* (tome 1 et 2). DUNOD, Paris.

Benzécri, 1981. Benzécri, J.-P. *Pratique de l'Analyse des Données : linguistique et lexicologie.* DUNOD, Paris.

Corbin, 1997b. Corbin, D. Locutions, composés, unités polylexématiques : lexicalisation et mode de construction. In M. MARTINS-BALTAR, Ed., *La locution entre langue et usages*,

Langages, pp. 53-101. Fontenay-aux-Roses: ENS Éditions Fontenay Saint-Cloud. Cité par Habert 1998.

Daoust, Dobrowolski, Dufresne, Gélinas-Chebat, 2006. Daoust F.; Dobrowolski, G.; Dufresne, M.; Gélinas-Chebat, C. Analyse exploratoire d'entrevues de groupe : quand ALCESTE, DTM, LEXICO et SATO se donnent la main, in *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp- 313-326, Presses universitaires de Franche-Comté, 2006. ISBN 2.84867130.0
http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/028.pdf

Daoust, 1996, 2004. Daoust, F. *SATO 4, Manuel de référence*, Centre ATO, UQAM, Montréal.

Daoust et Marcoux, 2006. **Daoust, F. et Marcoux, Y.** Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés. In *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp- 327-340, Presses universitaires de Franche-Comté, 2006. ISBN 2.84867130.0
http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/029.pdf

Daoust, 1999. Daoust, François. *Corpus Message d'amour : analyse exploratoire.* Centre ATO, Montréal,
http://www.ling.uqam.ca/sato/analyses/amour1.html

Daoust et Dupuis, 1996. Daoust, F., Dupuis, F. Analyse de texte et parallélisme, un protocole pour la mise au point d'algorithmes de désambiguisation catégorielle. In *Actes du colloque sur le traitement automatique du français écrit : développements théoriques et applications*, édités par Louisette Emirkanian et Lorne Bouchard, 1996.

Duchastel et coll., 2005. Duchastel, J. et coll. ATONET, Réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur. http://www.atonet.net

Gélinas-Chebat et coll. 2004. Gélinas-Chebat, C.; Daoust, F.; Dufresne, M.; Gallopel, K. et Lebel, M.-H. *Analyse exploratoire d'entrevues de groupe : les jeunes Français et le tabac.* In Purnelle, G., Fairon, C. et Dister, A. éditeurs, I, *Actes des JADT-2004*, pages, 479-487.

Gélinas-Chebat et coll. 2004b. Gélinas-Chebat, C.; Daoust, F.; Dufresne, M. Gallopel, K.; Lebel, M.-E. «What They Say : A Computer Text Analysis of teenagers' Interviews on Smoking», *Advances in Marketing: Concepts, Issues and Trends, Proceedings of the Annual Meeting of the Society for Marketing Advances* (SMA-2004), William J. Kehoe & Linda K. Whitten (Eds), St-Pete Beach, Fl. (US), pp. 220-223.

Habert, 1998. *Des mots complexes possibles aux mots complexes existants : l'apport des corpus*, *Mémoire présenté pour l'obtention d'une habilitation à diriger des recherches.* Document de synthèse, Université Lille III - Charles de Gaulle
http://www.limsi.fr/Individu/habert/Publications/Fichiers/hdr/node4.html

Lebart, 2005. Lebart, L. *Data and Text Mining.* École nationale supérieure de télécommunications, Paris. http://www.enst.fr/egsh/lebart/

Lebart et Salem, 1994. Lebart, L. et Salem, A. *Statistique textuelle*. Paris: Dunod.

Leventhal et coll., 1987. Is the Smoking Decision an 'Informed Choice? Effect of Smoking Risk Factors on Smoking Beliefs. Leventhal, Howard ; Glynn, Kathleen; Fleming, Raymond. *JAMA*. 1987;257(24):3373-3376.

Parker-Pope, 1997. Parker-Pope, Tara. Danger: Warning Labels May Backfire, *Wall Street Journal* (April 28), B1, B8.

Reinert, 2002. Reinert, M. *ALCESTE, Manuel de référence*, Université de Saint-Quentin-en-Yvelines, CNRS.

Salem, Lamalle, Martinez et Fleury, 2003. Salem, A.; Lamalle, C.; Martinez, W. et Fleury, S. *Manuel Lexico 3*, version 3.41. http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/lexicoWWW/team.htm

Salton, 1989. Salton, Gerald. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley, p.279

Witte & Allen 2000. **Witte, Kim & Allen, Mike.** *A Meta-Analysis of Fear Appeals: Implications for Effective Public Health Campaigns*. Health Education & Behavior, Vol.27 (5): 591-616.

4 Trente ans de développement et d'utilisation du logiciel SATO

4.1 Introduction : de l'ordinateur central au PC, et du PC au traitement distribué

Le logiciel SATO, tel qu'on le connait (versions 3 et suivantes), est apparu au début des années 1980 après des années de développement à temps très partiel. Il faisait suite à une évaluation critique d'une première génération du logiciel développée dans les années 1970. Suite à la création du Centre d'analyse de texte en 1983, la vie du logiciel est intimement reliée au développement de méthodologies et de projets d'analyse de texte assistée par ordinateur. Voilà pourquoi, dans ce chapitre qui expose les divers jalons de cette histoire, nous aborderons tout autant les aspects méthodologiques et théoriques que les aspects plus proprement informatiques.

Comme informaticien, j'ai toujours considéré que mon travail devait être guidé par une préoccupation de formalisation devant conduire à des modèles de calcul et à des logiciels qui, tout en s'inspirant de pratiques et de méthodes en action, auraient la rigueur d'un objet formel avec des propriétés logiquement bien établies et capable d'affronter l'épreuve du temps au-delà de l'évolution de la technologie qui redéfinit sans cesse les contraintes et les langages informatiques de même que les paradigmes de programmation et d'interface utilisateur.

Ainsi, malgré la variété des environnements informatiques et des projets d'analyse textuelle, nous verrons, au cours de ce chapitre, comment le modèle SATO (version 3 et plus) a accompagné ces projets tout en évoluant, du point de vue informatique, à travers les multiples changements techniques qui ont caractérisé et caractérisent toujours la quincaillerie informatique et les paradigmes de programmation.

Jusqu'à la fin des années 1980, l'informatique scientifique passait par l'utilisation d'ordinateurs centraux. Périodiquement, les centres de calcul changeaient de génération d'ordinateur

impliquant des changements de structures internes de représentation des données, des changements de systèmes d'exploitation et de compilateurs de programmes. SATO s'est donc d'abord développé dans ce contexte technique contraignant. Mais, du côté organisationnel, cette période des ordinateurs centraux a aussi été l'occasion de projets visant à partager les ressources dans l'idée d'offrir des librairies de programmes, de procédures, de corpus et de ressources langagières. Un exemple typique de cette tendance est le projet SACAO (1987-1989) qui visait à fournir un environnement intégré d'analyse de texte assistée par ordinateur sur ordinateur VAX. Comme on le verra, au-delà de l'aspect technique, ce projet s'inscrivait dans une certaine conception et pratique de l'analyse de discours.

L'arrivée de la micro-informatique a provoqué une baisse des coûts du *temps calcul*. Mais cet avantage a eu un prix : augmentation de la charge d'entretien par l'utilisateur et par les développeurs qui devaient faire face en même temps à des ordinateurs très différents, IBM-PC et MacIntosh, par exemple, sans parler des changements à répétition des systèmes d'exploitation. L'arrivée des interfaces graphiques a aussi eu pour effet de mettre beaucoup de pression pour le développement des interfaces et du *presse-bouton* au détriment, quelques fois, de la rigueur méthodologique dans l'utilisation de l'ordinateur à des fins d'analyse.

Heureusement, le développement de SATO a pu bénéficier au cours de la décennie 1990 de plusieurs années de recherche-action commanditée par divers ministères du gouvernement du Québec. Reconnaissant le potentiel de l'analyse textuelle pour la *lecture professionnelle*, ces projets ont permis de sortir l'ATO du cercle strictement académique par le développement de méthodologies bien documentées sur des corpus importants. L'apport économique de ces projets a permis de soutenir un développement logiciel qui ne pouvait trouver, au Québec, d'appui sérieux du côté des entreprises. Certes, on a connu avec le projet *Alex* (1992-1993). une courte période d'investissement privé aux allures d'abris fiscaux. Cette période s'est achevée abruptement avec la fin de ces programmes fiscaux et sans réelles retombées. En même temps, l'appui gouvernemental à la recherche-action s'est progressivement étiolé dans la dernière moitié des années 1990 avec une vague très importante de restrictions budgétaires. Nous sommes alors passés en mode survivance jusqu'à ce que de nouveaux programmes nous donnent un peu d'oxygène dans les années 2000: projet ATO-MCD (2002-2005) et ATONET (2005-2008).

Dès 1996, avec le projet *Visibilité*, plutôt que de développer des versions de SATO pour PC avec interface Windows et pour Mac avec son interface, on décide d'orienter le développement de SATO vers une architecture client-serveur utilisant un protocole Web standard. Dans cette perspective, toute la partie interface-utilisateur devait prendre la forme de pages Web actualisées à partir de gabarits comprenant des formulaires permettant de produire des commandes de façon interactive. En plus de permettre un accès au logiciel à partir de tout ordinateur et système offrant une interface Web, cette stratégie permet un accès décentralisé à des ressources centralisées comme à l'époque des ordinateurs centraux. Avec le projet ATO-MCD (2002-2005), cette idée sera portée à maturité avec, en plus, le déploiement d'une grappe de calcul permettant de fédérer plusieurs PC en fonction de la demande. Une même architecture peut donc être déployée du simple portable en mode autonome au puissant serveur fédératif. Sur cette base, il est maintenant possible d'envisager une normalisation XML, en particulier XML-TEI, des ressources et interfaces, avec un accès possible sous forme de services Web éventuellement répartis. Ce plan stratégique pourrait être complété par des systèmes institutionnels de dépôts de données permettant la description, la conservation et l'accès aux ressources, en particulier aux corpus, à leurs annotations et analyses.

Pour concrétiser ce que nous venons d'esquisser à grands traits, ce chapitre comprend d'abord une section retraçant les premières versions de SATO et sa critique qui a mené à la conception du logiciel que l'on connait maintenant. Viendront ensuite plusieurs sections faisant la synthèse d'un certain nombre des projets évoqués dans le tableau 4.1, en donnant la priorité aux projets qui ont donné lieu à des publications.



Repères historiques (4.1a, remarque)

- 1972. Début du projet SATO sous la direction de Jean-Guy Meunier. Réalisation d'une première version de SATO par Stanislas Rolland.
- 1975. SATO, version 2 du progiciel SATO réalisée par François Daoust.
- 1978. SATO, version 2 : progiciel FORTRAN amélioré.
- 1984. SATO version 3.0 et 3.1 : nouveau modèle, nouvelle programmation en PASCAL réalisée par François Daoust.
- 1986. SATO version 3.2 et 3.3.

- 1987. SATO 3.4.
- 1987-1989. Projet SACAO (*Système d'Analyse de Contenu Assistée par Ordinateur,* financée par le FCAR.
- 1988-1998. *SATO-CALIBRAGE*.
- 1988-1989. Financement par le Comité Consultatif en gestion du personnel du gouvernement du Québec en vue d'une utilisation de SATO pour l'indexation des conventions collectives.
- 1989-1990. Projet PLSD (analyse du vocabulaire gouvernemental) financé par le ministère des Communications du Québec.
- 1991-1993. Conception d'un système expert pour l'aide à l'analyse (tri, classification et indexation) des documents de jurisprudence-CEFRIO-SOQUIJ. Chercheur principal : Suzanne Bertrand-Gastaldy.
- 1992. SATO, version 3.6.
- 1990-1994. Projet ACTE (Atelier cognitif et textuel), financé par le ministère des Communications du Québec.
- 1992-1993. Projet AlexAto (Analyse de textes et parallélisme). Responsable : Jules Duchastel ; membres de l'équipe : Josiane Ayoub, Suzanne Bertrand-Gastaldy, Gilles Bourque, François Daoust, Monique Lemieux.
- 1992-1993. Analyse des programmes d'études au moyen de SATO. financé par le ministère de l'Éducation.
- 1993-1994. Étude documentaire du corpus Hydro-Québec (Grande Baleine). Projet CEFRIO-Hydro. Chercheur principal, Fernande Dupuis.
- 1993-1994. Analyse de rédactions d'élèves du primaire en vue de constituer une échelle orthographique. Commandite de la Direction de la recherche au ministère de l'Éducation du Québec.
- 1996 SATO, version 4.0.
- 1996-1998. Projet Visibilité : début du développement de la version HTML de SATO. Subventionné par le programme *Action concertée FCAR-CEFRIO-CRIM.*
- 1997. Évaluation de l'utilisation de SATO-CALIBRAGE pour les textes en

français langue seconde, financé par le ministère de l'Éducation du Québec.

- 1998. Développement d'une version Internet de SATO-CALIBRAGE, financé par le ministère de l'Éducation du Québec.
- 1999. Analyse de lisibilité et de cohérence lexicale des nouveaux programmes de l'enseignement primaire québécois, financé par le ministère de l'Éducation du Québec.
- 1999. Corpus Bernier, publication sur le Web, contrat du Secrétariat à la politique linguistique. Chercheur principal : Christine Portelance.
- 1999-2000. Corpus linguistique en environnement québécois avec Philippe Barbaud et Fernande Dupuis. Contrat du Secrétariat à la politique linguistique.
- 2002-2005. Projet ATO-MCD d'infrastructure XML-Web d'analyse de texte par ordinateur pour la *Chaire de recherche du Canada Mondialisation, démocratie et nouvelles régulations politiques* dirigée par Jules Duchastel de l'UQAM.
- 2005-2008. ATONET Réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur.

4.2 SATO, versions 1 et 2

Le projet SATO a démarré en 1972 à l'occasion d'un projet de recherche interdisciplinaire dirigé par Jean-Guy Meunier, un professeur de philosophie de l'Université du Québec à Montréal. Ce projet a donné lieu à une première version de SATO réalisée par Stanislas Rolland, un étudiant en mathématiques à l'UQAM. Suite à son départ du projet, alors que j'étais aussi étudiant en mathématiques à l'UQAM, je prendrai la relève pour réaliser la version 2 du SATO. Le Manuel de l'usager sera publié en 1975. Un article publié en 1976 dans la revue *Computers and the Humanities* décrit l'architecture de la version 1 de SATO. D'emblée, l'article situe SATO dans ce qui deviendra le leitmotiv d'une certaine tradition d'analyse de texte à l'Université du Québec à Montréal.

> A textual data processing system for literacy and social sciences, called SATO (Système d'analyse des textes par ordinateur) has been implemented with a view of providing an interaction between textual data description and interpretation. It helps to generate hypotheses which serve as starting points for the researcher's interpretations. (Meunier, Rolland et Daoust, 1976 : 281)

Cette idée d'un outil permettant au chercheur d'interagir avec le matériau textuel pour élaborer des hypothèses interprétatives s'appuiera aussi sur une démarche philologique visant à coder dans le texte électronique les caractéristiques éditoriales du texte manuscrit ou imprimé. C'est ainsi qu'on prévoyait plusieurs alphabets pour encoder des textes multilingues. L'article insiste beaucoup sur la nature interactive de SATO alors que beaucoup des programmes de l'époque, par exemple *Jeudemo* (Ouellette 1972) développé à l'Université de Montréal, fonctionnaient en mode lot en produisant des résultats sous forme de listages imprimés. La chose peut paraitre banale aujourd'hui, mais, pour l'époque, cette position méthodologique était avant-gardiste. L'accès aux textes et à des fonctions d'analyse de texte par l'utilisation de terminaux disséminés à travers le campus, et au-delà par l'utilisation de lignes externes de connexion, amorçait un nouveau rapport aux textes. C'était, pour ainsi dire, une préfiguration de la bibliothèque Internet, même si les bibliothèques virtuelles donnent encore peu accès, même aujourd'hui, aux outils de l'analyse de texte assistée par ordinateur.

L'interactivité du SATO de l'époque se traduisait par une série de menus emboités permettant d'activer le traitement sur l'ordinateur central de l'université. Mis à part ce mode d'accès interactif, le logiciel fédérait des modules indépendants à la manière des progiciels de l'époque. On y retrouvait les fonctionnalités classiques de l'analyse de texte par ordinateur avec des variantes originales, comme la possibilité de définir des sous-textes pour la production d'éditions, de concordances et de comptages.

Sur le plan de la représentation interne des données, on détecte l'influence d'un paradigme informatique en émergence à l'époque autour du langage Lisp (McCarthy, et coll. 1962), même si on n'y fait pas référence explicitement.

> Once it has been validated and corrected, the text is stored in the computer in a way that permits economical interactive work. To this end, the linear representation of the text has been abandoned in favor of a list structure. (idem : 282).

Sans le décrire précisément, l'article de 1976 fera référence à un système complexe de pointeurs permettant de reconstituer l'ordre séquentiel des mots du texte. On en trouve une description dans deux chapitres (Rolland et Daoust 1976a, 1976b) d'un Cahier de recherche de 1976. De fait, la structure de la représentation interne des fichiers SATO de cette première génération du logiciel était complexe et présentait plusieurs incohérences. Le facteur d'expansion de la représentation interne par rapport au document texte était important. Ainsi, au moment où j'ai pris connaissance du programme, chaque occurrence occupait 180 bits, soit plus de 22 octets sans parler de l'espace occupé par le lexique et l'index des pages. On avait des pointeurs arrières et avants qui devaient être utilisés pour recréer l'ordre séquentiel des occurrences dans le cas où on aurait décidé de modifier la représentation initiale du texte. Aussi, l'algorithme de génération de ces fichiers internes se dégradait considérablement lorsque le fichier-texte dépassait une certaine dimension. Plusieurs parties du code informatique étaient non documentées et très difficiles à maintenir. Cette première version de SATO, développée en Fortran contenait un certain nombre de bonnes idées mais qu'il fallait reprendre dans un modèle beaucoup plus rigoureux nécessitant une nouvelle programmation. SATO 3 et sa suite sera donc bâti sur la base d'un bilan critique du SATO des années 1970.

Voici, en résumé, les éléments principaux de cette évaluation critique.

1. Le modèle de liste évoqué dans l'article de 1976 visait la mauvaise cible. L'enjeu n'est pas de contourner la structure linéaire séquentielle du texte, mais de l'asseoir sur une typologie lexicale émergeant des données textuelles. La suite des occurrences dans le fichier informatique doit correspondre à la séquentialité des mots du texte si on veut une implantation informatique efficace du parcours de la linéarité textuelle. Donc, plutôt que de parler de *chaine des items, selon la terminologie* du modèle initial, il faudrait parler de *suite d'occurrences* de formes lexicales.
2. L'avantage de remplacer les chaines de caractères par la chaine des occurrences, est de produire, pour chacun des mots, une structure de longueur fixe facile à gérer et qui libère le programme du décodage répétitif des chaines constituant les mots. Mais, au-delà de cet avantage technique, l'enjeu fondamental est d'établir une véritable relation d'héritage entre la classe lexicale (le type) et l'instanciation de la classe dans un contexte donné, c'est-à dire dans la séquentialité textuelle. Du point de vue informatique, cela se traduit par le remplacement des caractères du mot par un pointeur vers la classe lexicale qui contient les caractères du mot, possiblement

normalisés, et toute information lexicale qui caractérise le type. Or, dans le modèle initial de SATO, la référence au type lexical n'est utilisée que s'il n'est pas possible d'inscrire directement la chaine des caractères dans l'espace fixe associé au pointeur lexical. Donc, au lieu d'une structure cohérente d'héritage, on a simplement un dispositif technique de compression des chaines. Ainsi, l'information lexicale obligatoire, comme le numéro de la langue, sera inscrit directement sur l'occurrence plutôt que de bénéficier d'une relation d'héritage dynamique avec la classe lexicale.

3. Le modèle initial de SATO prévoyait un mécanisme de journalisation des opérations. Mais, ce journal, au lieu d'enregistrer la démarche analytique de l'usager, n'était qu'une compilation statistique globale des fonctionnalités du logiciel. C'était le journal du développeur plutôt que le journal de l'analyste.
4. Le principe du dialogue interactif avec le corpus est une idée essentielle. Mais, le modèle d'interaction utilisé faisait perdre une autre idée essentielle, celle de la *scénarisation* permettant à l'usager de constituer des *macro-commandes* permettant de construire des dispositifs expérimentaux exposant des démarches d'analyse stabilisées. Cette faiblesse importante nous a amené à un nouvel examen du logiciel *Jeudemo* (voir 4.2a) dont l'interface était basée sur l'idée de *programme*. Cela nous a conduit à la conclusion qu'il fallait que l'interface interactive soit fondée sur un langage de commande cohérent permettant de composer les fonctions de calcul et d'élaborer des scénarios de commandes.
5. La structure fonctionnelle de ce premier SATO fédérait des fonctions de calcul indépendantes et difficilement composables. Ainsi, même si l'article de 1976 annonçait des fonctionnalités avancées encore à développer, catégorisation, lemmatisation, analyse syntaxique et dépendances structurelles, aucun mécanisme n'était prévu dans le modèle pour intégrer cet ajout d'informations.



Jeudemo, logiciel contemporain à SATO 1. (4.2a, remarque)

Le logiciel Jeudemo a été développé à l'Université de Montréal par Francine Ouellette (Ouellette 1972) à la même période que la première version de SATO à l'UQAM. Jeudemo est inspiré de *Cocoa* (Russell 1967) et de Concord (Hamilton-Smith, 1970). Le système construit un index des mots (formes graphiques)

contenus dans les textes soumis. Les index permettent d'avoir la liste des mots avec leur fréquence et la ligne des occurrences dans le corpus et ils sont utilisés pour produire des concordances de longueur définie par l'usager. Les index sont construits et interrogés au moyen de programmes rédigés par l'usager et composés d'une suite d'instructions spécifiques à Jeudemo.

Outre les mots, la codification des textes admet des codes de langue, et des codes de titre. Chaque langue produit des index distincts en suivant les directives d'une instruction ALPHABET qui permet à l'usager de définir les codes de caractères et les séparateurs de la langue. Les formes graphiques peuvent aussi être accompagnées de codes de mots qui produiront des entrées distinctes dans l'index. Ces codes pourraient, par exemple, être utilisés pour distinguer des mots identiques, mais portant une catégorie grammaticale distincte. Les codes de titre pourront être utilisés pour définir des sous-sections dans les textes et pourront donner lieu à une table des matières. Ces règles permettent un codage des textes élaboré mais assez complexe.

Le logiciel permet de repérer des mots ou expressions en spécifiant le préfixe ou le suffixe des mots, un intervalle de fréquence ou l'appartenance à une section ou à une langue. Dans les expressions, Jeudemo permet aussi de spécifier une distance entre les mots.

Voici la liste des commandes de Jeudemo.

- INDEX est utilisé pour construire les index ;
- LMOT permet de construire des listes de mots ou de groupes de mots dans le corpus ou dans des sections du corpus en spécifiant des critères supplémentaires en option ;
- CONCO sert à construire des concordances faisant appel aux même type de spécifications que LMOT ;
- TABLE permet d'obtenir les mots du texte sous forme de table, sans construction préalable d'index;
- KWIC permet d'obtenir une concordance complète du texte; sans construction

préalable d'index;

- COPIE copie le texte sur imprimante.

Les fonctions de codage des textes de Jeudemo et de la première version de SATO sont très semblables, même si la syntaxe diffère. Les deux logiciels utilisaient la notion d'alphabet pour dresser des lexiques distincts. La notion de *section* dans Jeudemo correspond à peu près à celle de *code de ligne* dans le SATO de l'époque. La première grande différence a trait à l'interface d'utilisation. Jeudemo fonctionnait en lot sous la gouverne de jeux de commandes appelés *programmes* alors que SATO fonctionnait en mode interactif à l'aide de menus emboités. Mais, la différence principale a trait à l'implantation. Jeudemo s'inspirait d'un modèle informatique issu de la tradition de la recherche documentaire (*information retrieval*) avec ses *fichiers inverses* permettant d'accéder aux fiches de données constituées ici des diverses lignes de texte. Cette approche permettait de réutiliser des technologies existantes, mais elle passait à côté du concept de classe lexicale et d'héritage qui, à défaut d'être exploité de façon conséquente dans le SATO initial, sera à la base de la future version en Pascal de SATO. En fait, comme on le verra dans la présentation de la version actuelle de SATO au chapitre 5, les fonctions de SATO ne reposent pas sur des calculs d'index mais sur un accès direct aux mots en tant que références aux classes lexicales. Et, si certains algorithmes font appel à des dispositifs apparentés à des index, il s'agit simplement de mécanismes d'optimisation interne qui n'interviennent pas dans le modèle formel de données.

C'est donc cette évaluation critique d'abord de SATO, et aussi de Jeudemo, qui nous a conduit à proposer un nouveau modèle, et une nouvelle programmation, pour un nouveau système d'analyse de texte par ordinateur (SATO 3) dont un premier devis est rendu public en 1979 (Daoust 1979). Il faudra cependant attendre jusqu'à 1984 pour que soit livrée la première version publique du nouveau logiciel écrit en Pascal. Ce long délai pour la réalisation de ce nouveau logiciel vient du fait qu'il a été développé à temps perdu par son auteur qui occupait un poste de soutien à temps plein au sein du service informatique de l'UQAM. Aussi, suite à la création du Centre d'analyse de texte par ordinateur en 1983, l'intérêt du nouveau logiciel a été reconnu et une entente a été conclue pour que le Centre ATO s'en fasse le distributeur. Son développement, par la suite, s'est inscrit dans le cadre des activités de recherche et de formation du Centre ATO.

4.3 Projet SACAO : ressources partagées sur ordinateur central

SACAO (Système d'Analyse de Contenu Assistée par Ordinateur) était un projet d'intégration de procédures de lecture assistée de données textuelles. L'objectif du projet, soutenu par une subvention de 1987 à 1989, était d'offrir aux utilisateurs, dans un environnement logiciel relativement intégré, divers modules de description, d'exploration et d'analyse de données textuelles, tout en leur laissant le soin de paramétrer ces procédures en fonction de leurs propres hypothèses de lecture. Ces procédures ne comportaient qu'un minimum de préconstruction théorique alors que l'intégration informatique était assurée par l'établissement de liens entre fichiers comportant des structures de données communes.

Le projet regroupait un certain nombre de professeurs et professionnels de recherche associés au Centre ATO du l'UQAM : Jules Duchastel, Luc Dupuy, Louis-Claude Paquin, Jacques Beauchemin et François Daoust. Le bassin des utilisateurs potentiels du système était plus large que la seule communauté des chercheurs. C'était tous les usagers de la langue écrite (documentalistes, gestionnaires, décideurs, etc.) qui étaient vus comme d'éventuels bénéficiaires du système. Un thème récurrent de toute cette période était la constatation que la disponibilité croissante de l'information textuelle sur support magnétique rendait plus criant le problème de la lecture. Tous ces documents répartis dans les banques de données et les répertoires de textes demeuraient donc largement sous-exploités.

> D'un côté, on trouve des usages en traitement informatique de la langue et une quantité croissante de données textuelles déjà disponibles, de l'autre, des procédures diversifiées d'écriture et de lecture assistées. Par contre, il existe peu de méthodologie pour l'usage intégré de ces procédures selon des protocoles définis. Ces procédures sont partielles, peu standardisées et souvent difficilement accessibles. Leur utilisation, quand elle a lieu, est peu stratégique faute de modèles d'utilisation susceptibles de guider les usagers. (Duchastel et coll. 1989b.)

Un article de Coulon et Kayser (1986) a eu une grande influence à l'époque sur l'énoncé de la problématique, comme en font foi les références à Coulon et Kayser dans notre article de 1989 (Duchastel et coll. 1989b).

Depuis leur origine, les recherches reliées à la modélisation informatique des langues naturelles se profilent suivant deux axes : l'adaptation des modèles linguistiques et logiques à des contextes informatiques et la mise au point des techniques d'"ingénierie du langage". Coulon et Kayser définissent deux optiques possibles correspondant à ces axes: le modèle philosophique dont le but est d'accroître la connaissance de la langue et le modèle ergonomique qui est orienté vers la production et l'utilisation d'outils. Dans un cas, il s'agit du projet de programmer une machine pour la compréhension automatique des phénomènes langagiers, dans l'autre, il s'agit plutôt de proposer des outils pour faciliter, par étape, cette compréhension.

(...)

L'histoire de ce domaine de recherche est traversée, de part en part, par ces deux optiques, mais elle est également caractérisée par une succession d'approches théoriques différentes qui ont dominé le champ durant des périodes données. En effet, chaque période est définie par la prévalence de l'une ou l'autre de ces approches, bien que chacune d'entre elles se soit superposée aux autres et continue, encore aujourd'hui, de se développer simultanément. Une première période (1945-1955), relativement étanche, a été caractérisée par l'approche statistico-morphologique. Elle fut suivie d'une dominance de la syntaxe de 1955 à 1970. Mais dès 1963, la recherche s'affairait à la programmation de modèles logico-sémantiques. Enfin, depuis 1974, le souci majeur est la représentation et l'organisation de la connaissance en faisant appel à des modèles cognitifs.

(...)

Ces recherches ont permis des avancées notables, mais elles ont mis en évidence un très grand nombre de problèmes. La prévalence épisodique de l'une ou l'autre approche souligne, à loisir, les espoirs maintes fois déçus d'avoir trouvé l'angle d'attaque privilégié pour atteindre la compréhension automatique des langues. Les développements disciplinaires ou d'écoles ont favorisé des avancées significatives, mais les contradictions entre diverses approches théoriques ainsi que l'opacité de certains modèles ont peu favorisé l'intégration des connaissances ainsi produites. La relative courte durée des projets indique l'existence fréquente d'impasses théoriques. La projection très problématique des avancées théoriques dans les applications

> pratiques a mis en évidence l'incomplétude des systèmes. A travers ce cheminement complexe, pourtant, les limites de couverture linguistique, conceptuelle ou inter-disciplinaire qui se sont révélées au grand jour, ont permis de réévaluer les difficultés liées à la compréhension des phénomènes de langue et de discours et certains problèmes sont ainsi apparus comme prioritaires. On pense à la contextualisation nécessaire des phénomènes de discours, à la représentation des connaissances, à la nécessité d'incorporer une quantité considérable de données extra-linguistiques dans les modèles de TAL, à la prise en compte de la logique dite naturelle. (voirCoulon et Kayser, 1986). (Duchastel et coll. 1989b)

C'est sur la base de ce bilan critique que le projet SACAO a privilégié une orientation pragmatique de valorisation des données textuelles, ce que Coulon et Kayser qualifient d'*optique ergonomique*. Aussi, le projet favorisait l'analyse des *morphologies du discours* plutôt qu'une approche trop strictement syntaxique ou sémantique.

> L'automatisation n'est recherchée que sur une base pragmatique et ne constitue pas une condition première. Nous mettons de l'avant une approche hybride, alliant procédures automatiques et assistées et une substitution de l'idée d'intégration maximale des outils à l'objectif de complétude des systèmes. Ce point de vue n'est pas uniquement pratique, en ce qu'il serait motivé uniquement par l'impératif d'une couverture large du monde réel. Il répond à une conception extensive du problème de la compréhension des phénomènes de langue et de discours. Il est fondé également sur la conviction du caractère créatif qui revient à l'usager dans le processus d'analyse. Les systèmes automatiques, aussi puissants soient-ils, proposent avant tout une boite noire aux utilisateurs. SACAO propose une méthode interactive où le chercheur investit ses hypothèses et construit progressivement son analyse à l'aide d'outils performants.
>
> Le projet SACAO s'est donc défini une posture épistémologique de nature empirico-constructiviste. De manière succincte, cette approche conçoit la connaissance des phénomènes langagiers comme le produit d'un processus non-univoque de construction des objets. Cela implique d'abord la coexistence de plusieurs procès de construction complémentaires (par exemple, multiplication des niveaux d'analyse) et potentiellement contradictoires (par exemple, la coexistence d'approches non exclusivement compatibles), ensuite la nécessité d'une démarche d'aller-retour entre

> la constitution des modèles et leur validation empirique. Cette démarche favorise la méthode inductive et le caractère interactif du système. Par exemple, nous évitons la projection du modèle aux données, et de manière plus ou moins déterministe, de modèles théoriques préconstruits sur le réel. Nous favorisons, au contraire, l'ajout de descriptions successives du texte en alternance avec l'exploration de résultats provisoires. (Duchastel et coll. 1989b)

Le présupposé théorique du projet SACAO envers l'analyse des morphologies du discours lui fait prendre ses distances par rapport à une certaine *approche étapiste* qui, distinguant les divers niveaux des phénomènes socio-linguistiques (morpho-lexical, syntaxique, sémantique, logique et pragmatique), en vient à proposer un *étagement* des niveaux de la langue et du discours dictant un ordre souhaitable dans les étapes de la recherche. Au contraire, « SACAO considère les divers niveaux de description comme la résultante d'un découpage et d'une construction différentielles de cet objet, et non comme les étapes ordonnées d'un parcours obligé qui mènerait de la description lexico-syntaxique à la compréhension globale de la langue naturelle ». (Duchastel et coll. 1989b)

SACAO considère donc le texte comme un espace diversement structuré selon qu'on l'observe du point de vue de la narration, du point de vue de l'argumentation, etc. « Il nous intéresse donc de repérer les modes de segmentation qui caractérisent l'organisation d'un texte et les condensations de sens qui se produisent en certains lieux privilégiés » (Duchastel et coll. 1989b). Cette perspective de SACAO n'impose pas un cadre conceptuel précis. Elle conduit plutôt à privilégier un environnement offrant une panoplie de moyens de lecture diversifiés et minimalement contraints. L'objectif est d'offrir des outils de manipulation des données dont les à priori théoriques sont identifiés et qui devront être sciemment employés dans des stratégies de recherche définies par l'analyste.

Le volet informatique, visant à établir des ponts entre logiciels et ressources interopérables, sera facilité par l'utilisation privilégiée d'un ordinateur central de type VAX. Il s'articulait directement à un volet d'expérimentation sur de grands corpus accompagné de l'écriture de fiches techniques devant servir de base à la rédaction de manuels. Mais, comme c'est souvent le cas dans ce type de projet, l'ambition dépasse les ressources allouées dans le cadre de subventions de recherche *ordinaires*. Ainsi, par exemple, l'ordinateur VAX deviendra vite insuffisant alors que la puissance des microordinateurs ne cessera d'augmenter avec,

cependant, des systèmes d'exploitation encore trop pauvres. Aussi, avec les acquis du balisage XML et de la normalisation des formats de documents, on est à même de constater que les problèmes de modélisation et de publication des données numériques ont été nettement sous-estimés alors que ces questions demeurent, aujourd'hui encore, d'une brûlante actualité. Il reste que l'intention méthodologique à la base de ce projet sera systématiquement reprise par la suite, par exemple dans le projet *Visibilité* (1996-1998), ATO-MCD (2002-2005) et ATONET (2005-2008). Ainsi, plusieurs projets d'analyse de corpus menés au Centre ATO, ou en collaboration avec le Centre, dans les années qui suivront SACAO permettront d'expérimenter et de documenter de multiples chaines de traitement donnant lieu à un savoir-faire indéniable dans le domaine de l'analyse de texte par ordinateur. En ce sens, on peut dire que le projet SACAO aura été précurseur.

4.4 Projet ACTE : influence du courant cognitiviste

Comme indiqué en 2.4, l'influence du courant cognitiviste à partir du début des années 1990 a été très marquée au sein du Centre ATO en amenant un nouveau paradigme de recherche, celui de l'ingénierie des connaissances, dans l'idée de produire des systèmes experts et d'extraire des connaissances à partir des textes. Ce paradigme était notamment motivé par des impératifs économiques visant l'augmentation de la production au sein de l'appareil gouvernemental par l'utilisation de ces systèmes qui devaient assister le travail des fonctionnaires appelés à diverses tâches d'évaluation et d'analyse : vérification fiscale, gestion du personnel, analyse des mémoires dans le domaine de l'évaluation environnementale, aide à l'évaluation dans le domaine de l'éducation, etc. La perspective du départ à la retraite de plusieurs fonctionnaires d'expérience donnait à ce programme de recherche un certain caractère d'urgence. C'est dans ce contexte que s'est développé un ensemble de projets de recherche-action entre des chercheurs du Centre ATO et des fonctionnaires stimulés par cette perspective de la *lecture experte*. C'est sur cette base qu'est né le projet d'atelier cognitif et textuel (ACTE) visant à marier l'approche SATO et la technologie des systèmes experts.

La nature de SATO en faisait un instrument privilégié de modélisation. Dans l'esprit du point de vue pragmatique promu par le projet SACAO, on constatait que l'analyse de texte assistée par l'ordinateur se traduisait le plus souvent par des heuristiques permettant de modéliser des

fonctionnements localisés du discours. On a alors considéré que l'expertise de modélisation de l'analyse textuelle à l'aide de SATO pourrait en partie bénéficier de l'ajout d'un moteur d'inférences permettant de réaliser des systèmes à base de connaissance pour exploiter le matériau textuel. Il s'agissait, en somme, de faire appel au mécanisme des systèmes experts pour tenter de modéliser des lectures au-delà de l'analyse lexicale traditionnelle.

> Une analyse de texte, telle que pratiquée dans les sciences humaines demande des niveaux de description supplémentaires, proprement textuels appelées macro-structure. Parmi ces derniers mentionnons les figures de style ou de pensées, la logique du réseau d'argumentation, l'environnement communicationnel, la thématique, etc. Ces systèmes s'appliquent à des unités d'une autre nature: à géométrie variable tels la phrase, le paragraphe ou encore tout autre découpage arbitraire justifié par une grille ou d'autres critères. Leur description ne semble pas uniquement dépendre de la structure arborescente de la description lexico-syntaxique. Dans la plupart des cas, les indices ne sont pas assez nombreux pour qu'une analyse puisse se faire. Par contre, lorsqu'il y a des indices, un filtrage basé sur des séquences de patrons morphologiques semble suffisant. Une connaissance du cadre formel propre au type de texte est de plus requise. Est-ce une lettre, un mémo, une documentation, un règlement, un article de loi, le résumé d'un texte, etc?
>
> Qui plus est, non seulement une connaissance de l'univers particulier du texte est requise, mais le lecture doit être informée des conventions sociales qui ont présidé à l'émergence du texte. Cette dimension, appelée intertextualité, situe le texte à décoder au-delà les systèmes linguistiques. La seule façon de contourner l'incertitude quant aux indices nécessaires et fournir quand même un cadre computationnel utile, c'est d'inclure le lecteur dans le processus de la fabrication du sens. Cette intuition est confirmée autant par les dernières théories de la psychologie que de la sociologie affirmant que les textes n'ont pas un sens univoque; le sens est plutôt construit par le lecteur au travers ses structures cognitives et culturelles résultantes de sa socialisation. Dans cette perspective, l'expertise de la lecture doit être prise en compte par le système. (...)
>
> Les systèmes à base de connaissance résolvent des problèmes en parcourant une chaine d'informations générée à partir des faits de l'espace de problème (base de faits). Cette façon de faire nous a inspiré un renversement d'approche

> computationnelle, le passage d'une stratégie de parsage déterministe pour une sémantique procédurale. Par ce terme nous entendons la reconstruction de la signification au moyen d'une chaine inférentielle dirigée par un but particulier: l'hypothèse de lecture. Cette chaine est faite par le déclenchement de règles d'interprétation ou de recatégorisation des segments à partir des faits dont on dispose. D'un côté, la configuration d'indices relevés dans les descriptions disponibles (morphologiques, syntaxiques, sémantiques) et les heuristiques du lecteur (sens commun). Cette stratégie présente l'attrait de respecter le caractère hautement associatif des propriétés associées aux unités lexicales. (Paquin, Daoust et Dupuy , 1990).

La technique des systèmes experts semblait très appropriée pour manipuler ces savoirs heuristiques déployées par le lecteur dans ses tâches de *lecture professionnelle*. Elle était vue comme un mécanisme assez simple que l'on pouvait informatiser de façon économique sans remettre en question le rapport interactif au texte qui est à la base de SATO. On évaluait aussi que la montée en popularité du paradigme des systèmes experts se justifiait entre autres par son apparente simplicité. Ainsi, pouvait-on penser qu'il s'agissait là d'un outil de modélisation dont la maitrise ne faisait pas appel à des habiletés particulières de programmation.

Comme Louis-Claude Paquin, un chercheur du Centre ATO, intervenait déjà auprès de plusieurs organismes en utilisant un prototype fonctionnel de moteur d'inférences appelé *D_expert*, il devenait possible de proposer un projet de développement informatique permettant de fusionner SATO et ce moteur d'inférence en respectant les fortes contraintes de l'informatique gouvernementale de l'époque. Ce nouveau système, baptisé ACTE pour atelier cognitif et textuel a reçu l'appui financier d'un consortium de ministères coordonné par le Ministère des Communications du Québec.

Ce projet d'un atelier cognitif et textuel (ACTE) est né d'un double besoin. D'abord, il était nécessaire d'augmenter la robustesse du moteur d'inférence, prototype fonctionnel en LISP fonctionnant complètement en mémoire vive. L'objectif était de développer un moteur d'inférence en module exécutable et portable sur les micro-ordinateurs en usage, en particulier sur les PC-DOS de l'époque et de jumeler ce moteur à la version PC de SATO. Il s'agissait ensuite de créer une synergie entre l'analyse de texte et l'approche des systèmes experts. D'un côté, la méthodologie développée pour l'extraction des connaissances repose en bonne partie

sur l'analyse de données textuelles. De l'autre, la manipulation des textes avec des outils informatiques devait profiter de la méthodologie des systèmes experts pour la construction graduelle d'algorithmes complexes.

Dans sa version Lisp, le moteur d'inférences D_expert utilisait un algorithme classique. Le moteur d'inférences compare la prémisse des règles d'inférences aux faits (chainage avant) ; à chacun des cycles, toutes les règles pertinentes sont invoquées (l'arbre de recherche est parcouru en largeur) ; le conflit entre plusieurs règles pertinentes est résolu par une mise en ordre croissant selon le nombre de filtres contenus par leur prémisse ; le traitement de l'incertitude se fait par combinaison de coefficients de confiance (Mycin) ; le traceur est multi-niveau, il est possible de préciser quelles informations sur le déroulement d'un traitement sont souhaitées : l'identité des règles, le résultat du filtrage, le cumul des coefficients et des statistiques. Une des grandes limites informatiques du prototype venait du fait que toutes les structures étaient en mémoire vive entrainant un engorgement du système avec l'inévitable accroissement du dictionnaire de la connaissance et de la base de règles d'inférences qui accompagne le passage des systèmes experts de l'état de maquettes à celui de prototype fonctionnel. (voir Paquin, Daoust et Dupuy, 1989).

L'écriture d'un nouveau moteur de recherche capable de gérer de grandes bases de connaissances sous PC-DOS posait donc des défis informatiques considérables qui ont exigé de concevoir de nouveaux algorithmes. D'abord, il a fallu dégager les fonctionnalités propres au moteur d'inférences de celles qui relevaient de la gestion des données : dictionnaire de la connaissance, règles d'inférences et requêtes. Ces données, dont la structure prenait la forme d'arborescences en mémoire, ont été transformées en fiches gardées sur disque et indexées de diverses façons. Ensuite, tout a été reprogrammé en Pascal en réutilisant, autant que possible, les librairies de code développées pour SATO également écrit en Pascal. Enfin, le moteur d'inférences lui-même a été repensé, comme nous l'exposions en 1992 à une conférence régionale de l'ACM.

> Un système expert est une mise en forme, selon un certain modèle informatique, de connaissances reliées à un domaine particulier d'expertise. La construction de systèmes experts pose donc la question de l'extraction de la connaissance du domaine. Or, cette connaissance est souvent enfouie dans des textes: traités, directives, manuels de procédures, transcriptions d'entrevues, etc. Les méthodologies d'analyse de texte permettent

d'isoler les segments textuels pertinents. Un atelier logiciel intégré facilitera le va-et-vient entre le texte et les règles d'inférence.

Mais, le besoin le plus important est d'un autre ordre. Il fait appel à l'idée d'utiliser les règles de production du système-expert pour modéliser certains processus de lecture. Plutôt que de formaliser le savoir contenu dans les textes, il s'agit de formaliser des savoir-faire de lectures particulières (Paquin, 1992). Le projet *ACTE* est donc surtout un instrument destiné à fournir de nouveaux outils pour l'analyse de textes par ordinateur.

Illustrons par un exemple. Déjà, avec SATO, on dispose d'un système capable de réaliser une variété d'opérations de repérage, d'annotation et d'analyse sur les textes. Par exemple, on se sert de SATO pour calibrer des textes destinés à l'enseignement (Laroche 1990). Pour ce faire, on dépiste divers indices sur le texte : longueur du texte, nombre de mots longs, nombre de mots inconnus, nombre de phrases possédant des constructions jugées difficiles, par exemple «nous» «nous» suivi d'un verbe.

Avec *ACTE,* nous pourrons modéliser des raisonnements sur ces indices. Par exemple, si nous rencontrons une phrase longue, sa difficulté de lecture peut être moindre si elle comporte une énumération en fin de phrase («J'invite à ma fête chez MacDonald mes amis Pierre, Jean, Jacques et Lucie») plutôt qu'une proposition emboitée («J'invite mes amis, que j'aime le plus, à ma fête qui aura lieu chez MacDonald»). Avec *ACTE* et son moteur d'inférences, nous pourrons bâtir des raisonnements: par exemple, «Si j'ai une phrase de plus de 20 mots, et si j'ai plus de 2 propositions, alors la phrase est difficile». (...)

«Système expert», «système à base de connaissances» et «moteur d'inférences», ces trois termes décrivent trois aspects d'une même technologie de base. Sans anticiper sur la suite du texte, on peut signaler que «système expert» renvoie à 1'objectif le plus courant de cette technologie, à savoir une simulation informatique des raisonnements effectués par un expert humain. Il s'agit généralement de raisonnements qu'on aurait de la difficulté à traduire en algorithmes stables et bien définis.

«Système à base de connaissances» renvoie davantage à la caractéristique fonctionnelle du système, à savoir qu'elle repose sur une base de données qui contient des énoncés formels qui représentent des parcelles de savoir: définitions

d'objets cognitifs et relations entre objets (généralement sous la forme de règles SI prédicat(objet) ... ALORS objet).

Finalement, le terme «moteur d'inférence» renvoie davantage à l'aspect organique du système. Il s'agit du dispositif logique, de la fonction de calcul qui permet à un système expert de chaîner des connaissances disparates déposées dans une base afin de procéder à des déductions logiques. C'est ainsi que s'opère un raisonnement aboutissant à des conclusions.

Dans un système informatique traditionnel, le «savoir-faire» est directement traduit sous forme de programme exécutable. Le programme est relativement fixe, bien structuré, et agit sur des données qui sont nombreuses. Le programme répond à un devis précis et à une tâche bien identifiée.

Les problèmes soumis aux systèmes à base de connaissances ont en quelque sorte des attributs inverses. Les données (qu'on appelle généralement les faits), sont en général relativement peu nombreuses («Docteur j'ai mal aux dents!»). La tâche à exécuter est par ailleurs assez mal définie et fait appel à des raisonnements approchés («C'est probablement la molaire»). Le «programme» (l'expertise ou savoir-faire) est généralement volumineux, sujet à de multiples modifications et peu structuré. En fait, on parle de système à base de connaissances parce que les connaissances sont un peu l'équivalent des données dans un système informatique traditionnel. Le système expert est donc un programme «éclaté» dont les instructions seraient déposées dans une base de données.

Dans un tel système, les instructions sont des énoncés d'implication qui prennent généralement la forme de règles de production: Si Prémisse Alors conclusion. Dans un moteur d'inférences, on dispose généralement de coefficients permettant d'accorder à la conclusion une valeur d'incertitude ou de croyance. Les mécanismes de composition de ces coefficients font partie du système et permettent de gérer des raisonnements approchés à travers l'ensemble de la chaîne inférentielle.

Dans un système à base de connaissances, la séquence implicite des instructions est supprimée et c'est un dispositif de calcul, qu'on appelle précisément le moteur d'inférences, qui assure l'enchaînement des instructions (règles de production). Pour ce faire, le moteur confronte chaque fait aux prémisses de l'ensemble des règles, quel que soit

l'ordre dans lequel elles ont été déclarées. La véracité de la prémisse d'une règle implique la réalisation de sa conclusion. Si les conclusions produisent de nouveaux faits, alors le moteur d'inférence va entreprendre un nouveau cycle. Cela signifie que les faits produits seront à leur tour confrontés aux prémisses des règles. Ces règles, si elles sont déclenchées, pourront produire de nouveaux faits et entraîner un autre cycle du moteur.

La notion de programme, en tant que structure déclarative d'ensemble, est donc remplacée par un processus dynamique de chaînage des règles. Par conséquent, on peut introduire dans le système des parcelles de connaissance sans devoir les inscrire dans une structure d'appel prédéfinie. Contrairement à un programme traditionnel, la modification d'une règle ne nécessite pas la connaissance de l'ensemble du programme. Cet aspect est très important lorsqu'on veut modéliser des raisonnements humains dont on a une connaissance approximative, des raisonnements qui se modifient constamment et qui n'ont pas ni la complétude, ni la rigueur d'un algorithme.

Le projet *ACTE* constitue un défi de taille du point de vue informatique. Il faut rappeler en effet que la plupart des générateurs de systèmes experts sont des systèmes assez gros faisant appel à des ressources importantes et utilisant des langages évolués. Le prototype D_Expert qui nous a servi de modèle pour *ACTE* est écrit en LISP et requiert l'utilisation d'un Macintosh bien équipé. Or, la commandite gouvernementale pour l'*ACTE* impose le développement du logiciel sur des ordinateurs de type IBM-PC avec le système DOS. De plus, l'objectif est de réaliser un système pleinement fonctionnel et donc efficace. Cela nous a conduits à un certain nombre de choix stratégiques.

1- Le logiciel sera développé autour du noyau existant que constitue le système SATO écrit en TURBO-PASCAL; on va utiliser au maximum la librairie existante et réaliser une interface intégrée.

2- Le D_Expert servira de modèle fonctionnel; le modèle d'implantation sera totalement repensé.

3- Le système sera un sur-ensemble fonctionnel du D_Expert; sa conception permet de réaliser un moteur multi-expertises et multi-requêtes; le système pourrait donc être implémenté dans un cadre transactionnel sur serveur central.

4- Les connaissances (définitions, faits et règles) devront résider sur disque étant donné les contraintes de mémoire; divers index devront être construits pour nous permettre d'accéder directement aux objets cognitifs.

5- Les règles devront faire l'objet d'une compilation afin d'en accélérer l'exécution.

6- Le système devra être dirigé par les faits. Cela signifie que les faits devront permettre de sélectionner les règles pertinentes et seulement celles-là. En conséquence le temps d'exécution d'une expertise ne dépendra pas du nombre total de règles dans le système mais de la complexité du problème à résoudre.

7- Le moteur devra pouvoir fonctionner en mode menu comme le D_Expert mais aussi en mode commande; l'ensemble de la base de connaissances et des commandes pourra être représenté en ASCII selon une syntaxe rigoureusement définie en BNF.

Illustration d'un cycle du moteur d'inférences.

Dans l'*ACTE*, le moteur d'inférences fonctionne en deux moments bien distincts: le chargement des expertises puis l'acceptation et l'exécution des requêtes. L'étape de chargement consiste premièrement à recevoir les objets cognitifs (définitions des données, les règles et les faits de départ) ; deuxièmement, le moteur charge celles-ci sous forme d'enregistrements dans un fichier séquentiel indexé.

Lors du chargement, on construit des index dans un fichier d'identificateurs et de références. Ce sont ces index qui permettront de parcourir la base de connaissances de façon efficace. Pour comprendre le fonctionnement de ces index, il faut indiquer que les faits (données) manipulés par le moteur d'inférences sont des objets structurés. On a donné le nom de «granules» aux définitions des diverses structures des faits. Ces granules sont organisés en base (ensemble de granules) et se caractérisent par des traits (des variables) qui permettent d'en qualifier les attributs. Ainsi, on pourrait dire que le granule «table» appartient à la base des «meubles» et se caractérise par les traits «couleur», «hauteur», «prix», etc. Chacun de ces traits peut prendre une valeur d'un type prédéfini. Ces divers identificateurs (base, granule traits) nous fournissent diverses clés pour accéder aux objets cognitifs. (...)

La règle fournit aussi diverses clés d'accès. En fait, dans l'*ACTE*, la règle est une structure (fiche du fichier séquentiel indexé) composée d'objets primitifs qui sont aussi des fiches. Ces objets appartiennent à deux catégories selon leur position dans

la règle. Il peut s'agir de clauses de la prémisse ou de conclusions qui sont déclenchées lorsque la prémisse est évaluée à vrai. Une clause est un prédicat sur un fait et la prémisse est une expression booléenne sur les prédicats. Les conclusions sont des actions, ou des «inférences» (des implications) consistant à générer de nouveaux faits ou à ajouter du nouveau à des faits existants. Les faits référés dans la prémisse ou la conclusion sont des clés d'accès à la clause ou à la conclusion. Dans un deuxième temps, la clause (chaînage avant) ou la conclusion (chaînage arrière) servira de clé d'accès à la règle.

Pour illustrer la logique du moteur, nous allons examiner de façon schématique les diverses étapes constituant un cycle d'inférences.

La première étape consiste à amorcer la requête. Le moteur examine chacun des faits de départ. Cet examen consiste à aller chercher les clauses qui font référence aux triplets Base-Granule-Traits qui caractérisent la structure du fait. Ces clauses sont filtrées, c'est-à-dire que le prédicat est appliqué au fait. Si la clause est vraie, elle sera empilée dans une pile de clauses.

Lorsque tous les faits ont été examinés, on procède à l'examen de la pile des clauses. Cet examen implique deux opérations. D'abord, on met à jour l'«état du monde» qui nous indique quelles sont les clauses vraies pour cette requête et avec quel coefficient de certitude. Ensuite, on consulte le fichier des identificateurs et des références pour constituer la pile des règles qui utilisent les clauses nouvellement vraies.

On est alors en mesure d'amorcer un cycle du moteur. On commence d'abord par vider la pile des clauses. Ensuite on procède à l'évaluation de chacune des règles de la pile des règles. Cette évaluation s'opère de la façon suivante:

- On évalue la prémisse de la règle. Cette évaluation signifie qu'on vérifie la véracité de la prémisse en appliquant l'algèbre de Boole sur la valeur de chacune des clauses telle que consignée dans l'«état du monde».

- Si la prémisse est vraie, on procède à l'exécution des conclusions de la règle. Si une conclusion est une inférence, à savoir un nouveau fait déduit de la véracité de la prémisse, alors on procède comme s'il s'agissait d'un fait de départ. On consulte le fichier des identificateurs et des références pour aller chercher les clauses afférentes

au nouveau fait. On filtre les clauses en les confrontant au fait et, si la clause s'avère vraie, on la rajoute à la pile des clauses nouvellement vraies.

- Lorsque toutes les règles ont été évaluées, en d'autres mots lorsque la pile des règles est vide, on examine à nouveau la pile des clauses et on génère une nouvelle pile de règles. On est alors prêt à amorcer un nouveau cycle.

- Si la pile des règles est vide, c'est que le moteur d'inférences a réalisé toutes les inférences possibles et le moteur s'arrête. (Daoust 1992)

Comme nous l'indiquions en 2.4, le modèle de la lecture experte se pose comme alternative au projet de *machine pour la compréhension automatique des phénomènes langagiers* (Coulon et Kayser 1986). Mais ce choix a aussi entrainé, dans une certaine mesure, un renoncement aux formalismes au profit de stratégies heuristiques de lecture humaine, comme l'illustre notre article de 1989.

D'une part, l'analyse des groupes nominaux d'un corpus de textes permet le dépistage d'unités cognitives et leur structuration en objets valués. Une fois que, parmi tous les substantifs, les concepts pertinents ont été retenus, les configurations nominales, appelés ingrédients, qui leur sont associés sont recherchées. Ainsi, par exemple pour le substantif "projet" on aura des configurations telles, l'assujettissement d'un projet, la pertinence d'un projet, etc. Les formes adjectivales présentes dans les contextes dépistés font apparaitre les quantifications et les échelles argumentatives qui positionnent virtuellement les autres valeurs qualitatives ou quantitatives possibles.

D'autre part, l'analyse des groupes verbaux assiste la rédaction des règles d'inférences. En effet, l'examen des verbes d'action permet le dépistage des opérations définies sur les objets. Leurs flexions et leur contexte en fournissent la modulation (actif, passif, nécessaire, facultatif, etc.), la localisation et la temporalité.

En plus de permettre l'accès sélectif aux textes par une recherche de contenu, en plus de fournir une assistance à la conversion des objets du discours en objets valués, l'atelier est aussi conçu comme un outil général pour l'analyse du contenu des textes. La neutralité de l'instrument, qui permet la coexistence de plusieurs niveaux d'analyse potentiellement contradictoires, favorise une démarche d'aller-retour entre la constitution de modèles sur les textes et leur validation empirique.

> Il faut voir en effet, qu'il n'y a pas dans *ACTE* de projection déterministe d'un modèle pré-construit sur le texte. Le savoir sémantique et procédural appartient à l'usager. L'approche privilégiée par l'atelier est donc la mise à jour de l'organisation du texte par l'ajout de descriptions successives du texte en alternance avec l'exploration de résultats provisoires. Grâce à l'analyseur lexico-textuel, l'usager peut très facilement projeter sur le texte ses propres systèmes de catégories issus d'hypothèses explicites quant à l'interprétation du texte. Ainsi, les dénombrements pourront être effectués sur les catégories tout autant que sur les mots. Cette façon de faire amène le lecteur à expliciter les éléments textuels susceptibles d'être porteurs de sens et à arrêter les critères à partir desquels ceux-ci seront retenus et comptabilisés. (Paquin, Daoust, Dupuy, 1989)

Telle que décrite, la lecture experte tend donc à simuler directement les heuristiques de la lecture humaine plutôt que de s'inspirer des indices fournis par les experts pour construire des modèles de lecture électronique des structures textuelles. Or, comme nous l'avons expérimenté dans les projets SOQUIJ et Sato-calibrage, présentés dans les sections suivantes, des algorithmes statistiques sont susceptibles de produire à leur manière des résultats obtenus d'une autre façon par des stratégies de lecture humaine. Le mimétisme du raisonnement humain par les règles de production du système expert peut s'avérer beaucoup plus lourd qu'un algorithme reposant sur des bases mathématiques bien maitrisées. En effet, la multiplication des règles les plus simples de l'algorithmie, les *si-alors* peut conduire, comme le savent les informaticiens, à une enchevêtrement qui en vient à échapper à la *compréhension*, au sens étymologique du terme. L'intention de simplicité et de transparence des règles de production risque donc rapidement d'être confrontée au problème de complexité que la formalisation tente justement d'aplanir. Cela dit, le mécanisme de l'inférence peut s'appliquer à des modes de représentation plus sophistiquées que les simples relations *si-alors.* Sur le plan informatique, on retiendra l'intérêt du modèle d'implantation du moteur d'inférence développé pour l'atelier cognitif et textuel. Aussi, comme nous le verrons, l'annotation structurelle est en constante tension entre la matérialisation d'une structure et la découverte des éléments de surface qui la déclenchent ou la réalisent. Ainsi, une annotation joue un rôle analogue à la découverte d'un fait nouveau qui tire à lui les clauses ou contraintes qui en présupposent l'existence pour la réalisation d'une structure attendue. L'approche algorithmique envisagée pour le moteur d'inférence pourrait donc nous inspirer pour

l'implantation d'un moteur d'appariement facilitant la complétion de structures d'annotation en attente de complétion.

4.5 Le projet SOQUIJ

Le *projet CEFRIO-SOQUIJ* est un projet de recherche dirigé par Suzanne Bertrand-Gastaldy, professeure à l'École de bibliothéconomie de l'Université de Montréal et chercheure associée au Centre ATO de l'UQAM. Ce projet concerne la modélisation des pratiques de lecture documentaire des professionnels de l'information juridique et la conception de systèmes informatisés d'aide au travail de classification et d'indexation de jugements motivés. Deux méthodologies seront mises à contribution dans cette recherche : l'enquête cognitive auprès des professionnels et l'analyse de texte assistée par ordinateur appliquée aux documents *primaires*, c'est-à-dire les jugements eux-mêmes, et aux documents *secondaires* produits par les professionnels : classification, clés d'indexation et résumés. C'est au début des années 1990 que s'est mené ce projet d'envergure auprès de la *Société québécoise d'information juridique*, la SOQUIJ, avec l'appui financier du *Centre francophone d'informatisation des organisations*, le CEFRIO.

> Pour faire face à l'afflux prochain de jugements sur support informatique généré par la saisie à la source, une équipe de recherche constituée de chercheurs du Centre d'analyse de textes par ordinateur(ATO) de l'Université du Québec à Montréal et de l'École de bibliothéconomie et des sciences de l'information de l'Université de Montréal a présenté un projet de conception de système expert pour assister l'analyse des jugements. (…)
>
> Le projet s'appuie sur la triple hypothèse qu'il est possible : 1) de modéliser les décisions prises par les conseillers juridiques pour analyser les arrêts ; 2) de construire des algorithmes d'analyse des jugements en plein texte pour assister ces décisions ; 3) d'opérationnaliser les algorithmes dans un milieu réel pour un corpus fortement normalisé comme celui de la jurisprudence. (Bertrand-Gastaldy et coll., 1992)

Deux approches de recherche seront combinées. On s'appuiera d'abord sur l'analyse des textes secondaires produits par les conseillers juridiques.

> Il s'agit : a) des rubriques de classification ; b) des mots-clés contrôlés issus du thésaurus et des mots-clés libres assignés dans la manchette à la suite de l'indexation ; c) du résumé structuré et des autres éléments d'une notice (nom des parties, tribunal, citations de lois ou de jurisprudence, etc.) (Bertrand-Gastaldy et coll., 1992).

Cette analyse des textes secondaires sera alimentée par une enquête cognitive auprès des conseillers visant à leur faire expliciter leur démarche de lecture. Aussi, les résultats de l'analyse textuelle seront utilisés pour stimuler les échanges avec les conseillers juridiques.

> Les conseillers juridiques mettent en œuvre un savoir très spécialisé qui tient à la fois de la connaissance intime des différents domaines de droit dans lesquels chacun a développé une expertise, de la nature des textes analysés, des politiques régissant chacune des publications et banques de données produites ainsi que des besoins des différents types d'utilisateurs auxquels elles sont destinées. (Bertrand-Gastaldy et coll., 1992).

La lecture des jugements effectuée par les conseillers juridiques correspond donc à un savoir-faire institutionnel qui produit des textes secondaires au service de la communauté des juristes. L'analyse de texte par ordinateur visera à identifier les caractéristiques de ces lectures professionnelles par l'examen des textes qui en résultent, en comparaison avec les textes intégraux et les divers outils du langage documentaire.

> Les traitements consistent à mettre au jour une série de caractéristiques propres aux unités lexicales ou textuelles, l'objectif étant de découvrir lesquelles de ces unités et lesquelles de leurs propriétés permettent de reproduire les résultats des analyses effectuées antérieurement par les conseillers juridiques. (…)
>
> Les propriétés attribuées aux données textuelles, en contexte ou hors contexte, consistent en l'ajout de connaissances de nature diverse qui enrichissent les chaines de caractères immédiatement accessibles à l'ordinateur et accroissent le nombre d'opérations auxquelles on peut ensuite les soumettre. Il peut s'agir d'informations résultant de décomptes statistiques, de connaissances générales de la langue (nature grammaticale des lexèmes), de connaissances spécifiques au domaine (vocabulaire, structure des jugements), de connaissances "documentaires" (champs d'une notice, appartenance des lexèmes aux langages documentaires, signification des différentes

> conventions typographiques dans les enregistrements), etc. Elles sont interprétables par un être humain et résultent de traitements automatiques, assistés ou humains.» (Bertrand-Gastaldy et coll., 1992)

Le traitement opéré sur les textes pourrait se résumer ainsi :

- Une catégorisation des unités lexicales et textuelles ;
- Une segmentation des documents selon divers critères de classement : le plan de classification, les unités d'indexation ou les éléments de la macro-structure ;
- Un regroupement des catégories et des mots pour augmenter la robustesse des indices (analyse en *cluster*) ;
- Une analyse discriminante pour calculer des fonctions permettant de reproduire la classification humaine à partir des indices de surface.

Après une phase de pré-traitement faisant appel à des logiciels de traitement de chaines de caractères, la catégorisation et les dénombrements se font avec SATO. Finalement, on fait appel à des progiciels statistiques pour l'analyse des données numériques produites par SATO.

L'enquête cognitive, pour sa part, a permis de mieux comprendre la chaine logique de l'analyse en fonction de la structure du discours juridique à l'œuvre dans le texte des jugements.

> Nous cherchons les techniques employées pour parcourir un texte, les différentes parties du texte examinées pour prendre une décision de tri, de classification, de résumé, d'indexation, les connaissances utilisées (importance de telle ou telle cour, poids à accorder à la nature des parties en cause, marqueurs du raisonnement du juge, contenu actuel de la base de données, besoins des utilisateurs, etc.), les catégorisations effectuées, les inférences faites pour passer des expressions en langue naturelle à leurs équivalences dans le thésaurus.» (Bertrand-Gastaldy et coll., 1992)

L'enquête cognitive a permis de constater que plusieurs règles heuristiques des documentalistes font appel à des chaines de raisonnement qui impliquent la macro-structure du document et le modèle logique qui structure le genre que constitue un jugement : par exemple, la distinction entre le *litige*, qui fait référence aux procédures légales, le *contexte*, qui décrit les faits, les parties, les circonstances, et la *problématique* dans laquelle on retrouve la matière juridique du jugement (Poirier, 1985).

> Nous avons interviewé les conseillers juridiques et procédé à de nombreux allers-retours entre leurs dires et le corpus. Ils ont identifié les éléments de la macro-structure des jugements qu'ils parcourent ; certains de ces éléments relèvent plus du paratexte que du texte lui-même et incitent d'ailleurs à cette forme de lecture : intitulé, tribunal qui a rendu la décision, nom des parties, lois ou articles de lois cités.
>
> D'autres indices sont purement lexicaux, ce sont les mots employés par le juge. Plusieurs d'entre eux sont de très bons discriminants pour un domaine donné et se retrouvent souvent dans le plan de classification et le thésaurus ; ceux qui pointent vers plusieurs domaines recevront une pondération appropriée. À la suite de diverses stratégies de catégorisation automatique, ces éléments se trouvent marqués explicitement et peuvent être repérés par le nom de leur propriété et les différentes valeurs possibles. (Bertrand-Gastaldy et coll., 1992)

La dernière étape du prototype a consisté à développer un système expert pour reproduire des chaines de déductions utilisées par les conseillers juridiques. Le système, à base de règles alimentant un moteur d'inférence, permet de faire des déductions et de quantifier la confiance que l'on porte aux conclusions par un calcul combinant les divers coefficients de certitude associés à la fois aux règles et aux faits dépistés. Le déploiement des règles s'accompagne d'une trace explicite rendant compte de la chaine des inférences.

> Pour l'implantation du système d'aide à l'analyse, nous avons retenu la technologie des systèmes experts, malgré les limitations de ce formalisme (découpage arbitraire de l'espace du problème, entre autres). La formulation des modèles en énoncés conditionnels et les règles d'inférences présentent des avantages ergonomiques. En effet, le recours à un générateur de système expert (GSE) permet à des non-informaticiens, après un entraînement approprié, de formuler de façon autonome les règles pour un système qui peut être très complexe. (Bertrand-Gastaldy et coll., 1992)

Comme le précise Bertrand-Gastaldy,

> Le recours à un système expert n'est toutefois pas indispensable. On peut très bien se contenter d'un système d'aide à la prise de connaissance du contenu qui, comme SATO, permette de visualiser les propriétés jugées importantes par un lecteur ou encore affiche seulement les passages de textes répondant aux propriétés souhaitées

(par exemple, toutes les premières et dernières phrases des paragraphes), ou mette en évidence ces passages par un surlignement ou une couleur distincte (toutes les phrases qui contiennent un terme consigné dans le thésaurus). C'est le lecteur qui effectue les opérations de sélection et de mise en forme des éléments ainsi soulignés.

Comme tout texte est susceptible d'être soumis à un ensemble extrêmement diversifié de parcours interprétatifs, chaque lecteur devrait idéalement pouvoir mettre en place la série de traitements qui correspond à ses objectifs de lecture. (Bertrand-Gastaldy, 1994b)

Dans un colloque en droit, Bertrand-Gastaldy résumait ainsi l'apport de cette lecture assistée par ordinateur: « Grâce aux traitements effectués pour le tri, la classification et l'indexation, il est possible de construire une forme d'aide personnalisée à la lecture des jugements. » (Bertrand-Gastaldy, 1993)

Décrivant la nature de cette aide, elle poursuit:

> Les propriétés lexicales et textuelles mobilisées pour les autres opérations : étiquetage des citations de lois, de la mention des parties en présence, des termes du domaine, ou tout simplement la recherche de certains marqueurs textuels introduisant des passages jugés importants ("motifs suivants", "chefs d'accusation", "les faits se résument comme suit", "ne répond pas aux critères", "j'en conclus", "loi", "arrêt", "dans l'affaire" ..., etc.) peuvent servir à diriger la lecture. Le logiciel SATO permet de souligner ou surligner en différentes couleurs les unités dotées de telle ou telle propriété. De plus, il est possible de demander l'affichage des seuls passages (phrases, paragraphes ou contexte numérique spécifié par l'utilisateur) dans lesquels apparaissent les unités dotées des propriétés requises: on peut ainsi ne visualiser que les paragraphes dans lesquels la jurisprudence est citée, dans lesquels se trouvent des citations de lois ou d'articles de lois, dans lesquels il est question des prétentions des parties, dans lesquels il y a débat sur un point de droit, etc. Ces opérations sur le texte peuvent être encapsulées sous forme de macro-commandes par chacun des conseillers juridiques, selon ses habitudes de prise de connaissance du contenu d'une décision dans un domaine donné. Les stratégies de lecture et de résumé varient, en effet, selon les individus et selon le domaine, car les jugements sont structurés différemment, contiennent des renseignements différents. On ne lit pas un jugement

> classé dans le domaine Famille comme on lit une jugement relevant du domaine Procédure civile. Dans certains domaines, ce qui est important ce sont les questions de droit discutées par le juge situées dans la problématique, dans d'autres domaines ce sont les argumentations des avocats localisées dans la partie contexte.
>
> Les valeurs de propriétés surimposées au texte de départ jouent donc le rôle d'un paratexte orientant la lecture humaine et déterminant l'affichage sur écran ou l'impression papier. (Bertrand-Gastaldy, 1993).

Plusieurs fonctionnalités de SATO ont été mises à contribution pour la constitution des corpus, leur enrichissement et leur analyse. Sans en faire une liste exhaustive, nous présentons, dans les paragraphes qui suivent un certain nombre de solutions que nous avons déployées au cours de cette recherche et dont le caractère exemplaire illustre bien nos méthodes d'analyse textuelle. Des guides seront d'ailleurs rédigés par la suite, guides qui montrent l'utilisation de SATO à des fins de contrôle de vocabulaire et d'indexation (Bertrand-Gastaldy 1994; Bertrand-Gastaldy et Pagola 1994b).

L'enrichissement lexical permet d'identifier les unités terminologiques qui seront lexicalisées, au-delà du découpage en occurrences (*tokenization* en anglais) effectué sur le texte brut. En particulier, on voudra lexicaliser les locutions terminologiques pertinentes au domaine, et d'autres indices lexicaux tels les noms propres de lieu, les abréviations, etc. Par exemple, du côté nominal, on aura des termes tels *acceptation du risque*, *acte criminel*, *action en réclamation*. Un certain nombre de ces locutions découlent de l'analyse des corpus constitués des outils documentaires utilisés par la SOQUIJ, index, thésaurus, plans de classification, etc. Sans être exhaustives, ces entrées normalisées fournissent des termes canoniques qui seront utilisés pour produire des termes effectifs tels qu'on peut les rencontrer dans les textes. Ces expressions seront utilisées par SATO comme autant de patrons qui seront appliqués sur les corpus afin de marquer les expressions à lexicaliser.



Dépistage des locutions et des termes complexes (4.5a, notice technique)

L'interface de SATO donne accès, dans la section tâches, à un procédurier sur le dépistage des locutions et des termes complexes. En voici le sommaire.

---

1. Introduction
2. Dépistage des mots composés
3. Créer un fichier de locutions
4. Modifier un fichier de locution
5. Appliquer un fichier de locutions
6. Dépistage de locutions par patrons syntaxiques
7. Annuler les liaisons

Le projet SOQUIJ a été un véritable laboratoire permettant la mise en place d'une variété de dispositifs expérimentaux combinant, comme nous l'avons indiqué précédemment, diverses approches d'analyse cognitives et textuelles. Un des objectifs était de traiter les jugements en texte intégral afin de fournir au conseiller juridique un maximum d'information pour l'aider dans son travail.

Pour construire ce dispositif d'analyse des jugements, on a d'abord analysé les outils documentaires utilisés par les conseillers : thésaurus, plan de classification, notices bibliographiques, index, etc. Ces outils ont été traduits en ressources linguistiques utilisables par SATO : dictionnaires de catégories, patrons de locutions terminologiques, sigles, abréviations, etc. Ensuite, on a analysé les documents d'analyse de jugements en texte intégral produits par les conseillers juridiques. Ces résultats prennent la forme de résumés, de décisions de pertinence et de mots clés pour la classification et l'indexation. On s'est tout particulièrement intéressé à l'analyse des résumés qui prennent la forme de *notices* rassemblées dans une publication spécialisée appelée *Jurisprudence Express*. Nous disposions d'un échantillon de plus de 1057 notices publiées entre janvier et juin 1991. Ces notices suivaient un plan logique dans lequel on retrouvait des éléments de référence (nom des

parties, tribunal, citations de lois et jurisprudence), les rubriques de classification du jugement, une liste de mots-clés contrôlés et libres et le résumé lui-même. Le résumé en texte libre suit un plan général en trois parties : litige, contexte, problématique ou décision.

Comme les jugements en texte intégraux, les résumés et outils documentaires étaient destinés à une édition papier, ils se présentaient, selon la norme de l'époque, sous la forme de fichiers truffés de codes destinés à la photocomposition. Dans le meilleur des cas, on disposait de formats d'entrée dans des bases de données informatiques ou en traitement de texte (*WordPerfect)*. Il a donc fallu développer une chaine de traitement des caractères afin de produire des fichiers structurés qui permettaient de récupérer les codes typographiques significatifs et de marquer les éléments structurels des documents grâce à la syntaxe de SATO avec son système de propriétés. À partir de ce format, il était possible d'utiliser SATO pour générer des ressources et les appliquer sur les notices et les textes intégraux.

Parallèlement, on procédait à l'enquête cognitive auprès des conseillers pour qu'ils verbalisent leur expertise en expliquant comment ils procédaient à l'analyse manuelle. Finalement, l'analyse textuelle produite à l'aide de SATO fournissait des outils de rétroaction auprès des conseillers qui se voyaient présenter le résultat annoté de leur texte d'analyse. Il faut d'ailleurs préciser que tout ce travail de recherche était fait en étroite collaboration avec les experts de l'organisme dans un contexte de transfert réciproque d'expertise. Ainsi, plusieurs rencontres de formation ont permis de présenter la démarche de recherche, d'exposer le fonctionnement des outils d'analyse textuelle et statistique et d'en présenter les résultats en toute transparence.

Comme nous venons de l'exposer, ce travail considérable sur les données textuelles nous a permis de constituer un corpus enrichi et décrit de diverses manières. On pouvait donc envisager d'exploiter ce matériel en le soumettant à des analyseurs statistiques destinés à construire des fonctions permettant d'associer le texte intégral décrit au jugement des conseillers : niveau de pertinence du jugement en termes de jurisprudence, classification et termes d'indexation du jugement.

Comme la mise à notre disposition des jugements en texte intégral avait subi de nombreux retards, nous avons utilisé notre corpus de résumés (notices) pour tester nos analyseurs statistiques, en particulier l'analyse discriminante. Finalement, après un travail considérable de mise en forme, nous avons pu constituer un corpus d'apprentissage de 565 textes intégraux répartis en 29 classes et sous-classes du plan de classification pour un minimum de 10

jugements par classe. Un corpus test de 103 documents a aussi été constitué pour valider la performance prédictive des indicateurs établis à partir du corpus d'apprentissage. Les dictionnaires et scénarios SATO résultant de l'analyse des outils documentaires de la SOQUIJ ont été appliqués sur les textes intégraux afin d'identifier les termes du domaine.

Une première phase d'analyse a permis de sélectionner un ensemble de 1871 lexèmes excluant les formes fonctionnelles, les codes et les noms propres. Ces termes potentiellement discriminants ont été retenus en raison de leur distribution manifestement inégale entre les classes. On a ensuite procédé à une classification des termes visant à constituer des groupes de termes partageant des profils de distribution similaires. Ce sont finalement ces groupes de termes qui ont été soumis à l'analyse discriminante visant à établir le *portrait-robot* de chaque classe. Plusieurs expérimentations avaient été envisagées pour la mise au point du meilleur algorithme. Mais, en raison du retard de livraison des textes intégraux, nous avons dû nous contenter de deux expérimentations. Malgré tout, on a obtenu un taux de bonne classification de 68% des jugements du corpus test. Donc, à près de 70%, la fonction statistique permettait d'attribuer à un jugement la bonne classe parmi 29 possibilités.

Le potentiel de recherche du *projet SOQUIJ* était considérable étant donné l'ampleur et la qualité du matériel rassemblé. Mais, comme le projet de saisie électronique à la source des jugements a été retardé, la pression administrative sur la SOQUIJ pour l'améliorer de ses processus de traitement a diminué et cette commandite privée de recherche n'a pas été renouvelée.

4.6 Le projet SATO-CALIBRAGE

L'objectif du projet SATO-Calibrage était de bâtir un dispositif de lecture électronique permettant de classer un texte selon le niveau scolaire auquel il serait le plus adapté en termes de facilité de lecture. Le projet a été mené en collaboration avec des enseignants et des professionnels du système scolaire québécois (Daoust, 1994, Daoust, 1996b). Plutôt que d'aborder directement le problème de la complexité des textes, pour lequel on ne possède pas de modèles informatisables simples, on s'est plutôt fondé sur le fait que le système scolaire produit son propre *discours pédagogique* s'exprimant à travers les textes fournis aux élèves des diverses classes d'enseignement. Et, comme on se situait dans un contexte d'apprentissage de la langue

maternelle, on s'est appuyé sur l'hypothèse selon laquelle le niveau de difficulté des textes devrait croitre avec le nombre d'années de scolarité. C'est sur cette base qu'on a décidé de constituer un corpus de référence rassemblant des textes produits par le système scolaire pour chacune des onze premières classes d'enseignement. Comme on savait aussi que les caractéristiques de genre allaient influencer considérablement la structure des textes, on s'est limité aux textes en prose en excluant la poésie et le théâtre.

On a aussi réuni un comité d'experts constitué de conseillers pédagogiques, d'enseignants et de spécialistes en évaluation. Ce comité a suivi le projet pendant toute sa réalisation en suggérant aux chercheurs des critères à prendre en considération dans l'analyse textuelle et en réagissant à l'ergonomie générale de l'application. L'équipe de chercheurs a fait appel au logiciel SATO pour bâtir des dispositifs d'analyse textuelle comprenant une variété de petits analyseurs sous forme de scénarios de commandes SATO. Les mesures produites par ces analyseurs ont été soumises à des analyses statistiques destinées à sélectionner et à combiner les diverses mesures de façon à produire un indice composite corrélé avec le niveau connu des textes du corpus témoin. C'est ainsi que nous avons produit l'*indice SATO-CALIBRAGE* qui, appliqué sur de nouveaux textes, permet d'estimer le niveau du texte par sa similitude avec une classe d'enseignement exprimée en nombre d'années de scolarité.

Divers outils de visualisation et d'explicitation accompagnent le rapport produit par SATO-CALIBRAGE de telle sorte que le lecteur humain puisse évaluer la portée de l'avis du *lecteur électronique*. Donc, dans ce projet, il n'y a pas eu utilisation de systèmes experts à base de règles d'inférence. C'est l'analyse statistique sur corpus qui a permis d'évaluer la performance des indices parcellaires et de les combiner de façon rigoureuse. L'explicitation de cet indice, appliqué sur un texte, de même qu'un ensemble de scénarios de navigation sur le lexique et le texte, permettent au lecteur de s'approprier l'outil pour se faire un jugement sur les facteurs de difficulté d'un texte en prenant en charge son contexte d'utilisation.

Plusieurs communications, articles et cahiers de recherche documentent la démarche de recherche et l'utilisation de l'application. L'article publié en 1996 dans la *Revue québécoise de linguistique* (Daoust, Laroche, Ouellet 1996) constitue une bonne présentation de l'ensemble du projet.

SATO-CALIBRAGE : présentation d'un outil d'assistance au choix et à la rédaction de textes pour l'enseignement (4.6a, publication)

On trouvera ici de larges extraits d'un article paru dans la Revue québécoise de linguistique, vol. 25, n° 1, 1996, (UQAM), Montréal. Auteurs : François Daoust, Université du Québec à Montréal; Léo Laroche et Lise Ouellet, ministère de l'Éducation du Québec. Des corrections mineures ont été apportées à la publication de 1996. La version intégrale de l'article peut être consultée en ligne à l'adresse http://www.ling.uqam.ca/sato/publications/bibliographie/sato-calibrage.html

1 Introduction

La lecture est au coeur des apprentissages fondamentaux de l'école. On souhaite non seulement que les élèves sachent lire et qu'ils aiment la lecture, mais aussi qu'ils puissent développer des stratégies de lecture efficaces afin de comprendre tout texte qu'ils doivent lire ou qu'ils ont le goût de lire. L'enseignement et l'évaluation de la lecture représentent donc des défis constants pour les enseignants[1] du primaire et du secondaire. Parmi leurs préoccupations, le choix des textes figure en tête de liste puisqu'il est le moteur de tout apprentissage de la lecture.

SATO-CALIBRAGE est un outil informatique qui veut répondre à ce besoin en offrant une assistance pour le choix et la rédaction de textes. L'une de ses caractéristiques est de donner un indice à un texte pour le situer sur un continuum, établi de la première année du primaire à la cinquième année du secondaire.

Après avoir présenté la problématique du projet, nous expliquerons la méthodologie suivie et nous ferons une description du prototype. Quelques exemples d'applications pédagogiques seront ensuite décrits.

Le présent article, de nature descriptive, présente sommairement nos travaux. Un rapport de recherche, cf. Daoust, Laroche, Ouellet & coll. (1993), expose en détail l'historique du projet, ainsi que ses dimensions linguistique et statistique.

2 Problématique

Dans le monde de l'enseignement, trois responsabilités professionnelles requièrent un éclairage particulier face aux textes : la planification de l'enseignement,

l'évaluation de la lecture et la rédaction de texte. Nous décrirons rapidement ces situations et nous verrons, dans la suite de l'article, comment un outil informatisé peut faciliter la tâche du choix de texte.

Plusieurs enseignants planifient leurs cours à l'aide du matériel didactique offert sur le marché. D'autres préfèrent présenter aux élèves des textes venant de sources diversifiées. Ils doivent alors se demander si ces textes sont accessibles aux élèves. Dans d'autres occasions, la planification de la lecture ne se fera pas en fonction d'une thématique mais plutôt à partir d'objectifs d'apprentissage déterminés ; par exemple, trouver le sens d'un mot inconnu d'après le contexte, comprendre les liens qu'établissent certains mots de relation ou comprendre le sens de phrases très longues. Dans tous ces cas, au moment de l'**enseignement**, on a besoin de savoir si le texte est adapté à son groupe d'élèves et quels sont les défis qu'il présente.

Choisir le ou les textes qui serviront à l'**évaluation** de la lecture est une tâche lourde de conséquence. Le texte choisi est en effet le point de départ de la définition de la tâche et de l'élaboration du questionnaire. Il arrive parfois qu'une épreuve de lecture n'atteigne pas ses objectifs parce que le texte est trop facile ou trop difficile. On veut donc un texte qui soit du bon niveau. On veut aussi repérer les difficultés du texte, soit pour les atténuer, ou soit au contraire, pour vérifier si l'élève est capable de surmonter les obstacles.

Les textes utilisés pour l'enseignement ou pour l'évaluation peuvent venir de sources externes. On peut aussi choisir de rédiger des textes. Toutefois, au moment de la **rédaction**, les mêmes questions se posent : on veut un texte ni trop facile ni trop difficile ou on veut rédiger un texte qui présente des défis particuliers pour que les élèves développent des stratégies de lecture précises. Comment peut-on affronter ce problème? Bien sûr, l'intuition et l'expérience viendront à la rescousse, mais est-ce objectif et suffisant? Est-on assuré d'avoir le texte que l'on recherche? Spontanément, le recours aux formules de lisibilité vient à l'esprit.

Il existe un certain nombre d'indices pour mesurer la lisibilité d'un texte[2]. Cependant, la plupart de ces mesures ont été conçues pour des textes rédigés en langue anglaise. Voilà pourquoi nous avons voulu développer un instrument de

mesure de la lisibilité qui soit adapté au milieu scolaire francophone du Québec. Pour ce faire, nous avons opté pour une démarche expérimentale basée sur une analyse comparative de larges corpus de textes. Ces corpus se veulent représentatifs du matériel fourni aux élèves québécois dans les cours de français langue maternelle au primaire et au secondaire.

Aussi, dès le début de nos travaux en 1989, il est apparu nécessaire d'établir une collaboration étroite entre le ministère de l'Éducation, le Centre d'ATO de l'UQAM et les milieux scolaires. On a donc mis sur pied un comité d'appui composé de conseillers pédagogiques de français (primaire et secondaire), de responsables d'élaboration d'épreuves (ministère de l'Éducation et Banque d'Instruments de Mesure --BIM) ainsi que de quelques personnes du milieu collégial et universitaire.

3 Méthodologie

Pour réaliser notre recherche, nous nous sommes appuyés sur certaines hypothèses touchant la nature du discours écrit. Pour valider nos hypothèses, nous avons mis en place un dispositif linguistique. Finalement, des traitements statistiques nous ont permis d'élaborer un indice rendant compte de la lisibilité des textes.

3.1. Le cadre expérimental

Comme tout projet en analyse de texte, notre démarche est fondée sur un certain nombre d'hypothèses sur la nature du discours. Ainsi, par exemple, on peut considérer que les textes fournis aux élèves de première année devraient, au-delà des variations individuelles propres à chaque texte, partager des caractéristiques communes qui les destinent à leur utilisation dans un contexte d'apprentissage ou d'évaluation. En d'autres termes, nous posons en postulat l'hypothèse générale de cohérence du discours social, plus spécifiquement ici, du discours produit dans le cadre de l'institution scolaire et destiné à un public cible composé d'enfants en situation d'apprentissage. D'un point de vue socio-linguistique donc, les textes constituant le matériel scolaire s'inscrivent dans un cadre institutionnel déterminé avec des acteurs sociaux historiquement définis. Les indices produits au cours de notre recherche seront donc marqués par ce contexte social. Cela n'exclut pas que

l'on puisse utiliser nos indices dans un cadre social différent. Il faudra cependant considérer ces différences dans l'évaluation des résultats.

En nous appuyant sur l'hypothèse générale de cohérence du discours scolaire, nous allons étudier le fonctionnement discursif en observant un ensemble de textes individuels. La question de la représentativité des données, à savoir ici les textes fournis aux élèves, est donc une des premières questions à poser dans une approche expérimentale. Cette représentativité implique des hypothèses sur l'objet à analyser, sur sa cohérence et sa variabilité. La constitution du corpus utilisé pour cette recherche traduit bien ces préoccupations. Pour chaque texte inclus dans ce corpus, un certain nombre de renseignements ont été recueillis: la classe d'enseignement où le texte est utilisé, sa provenance et, dans certains cas, le type de discours. Des données statistiques sur une quinzaine de variables linguistiques ont ensuite permis d'épurer le corpus disponible en rejetant les textes atypiques. Par exemple, un texte où il n'y a pas de points ou presque est probablement un poème. Enfin, les cas litigieux ont été soumis à un groupe d'experts qui ont confirmé ou révisé le classement des textes par rapport aux classes d'enseignement.

3.1.1 Constitution du corpus

Les textes qui composent notre corpus proviennent du matériel didactique approuvé par le ministère de l'Éducation du Québec (MEQ). Il peut aussi s'agir de textes utilisés pour l'évaluation. Dans les deux cas, il était donc possible d'attribuer une classe d'appartenance à chaque texte. Au début, le corpus contenait des textes qui correspondaient à tous les types de textes imposés dans les programmes d'études. Par la suite, les poèmes, les chansons, les comptines, les charades, les faits divers, les lettres d'invitation, les contrats et les extraits de pièce de théâtre ont été exclus et ce, pour deux raisons: d'une part, ces types de textes étaient marginaux dans le corpus et, d'autre part, leurs formes linguistiques les démarquaient nettement de l'ensemble. Nous avons donc décidé de nous concentrer sur des textes homogènes, les plus couramment utilisés dans l'enseignement du français au Québec; nous avons cherché à constituer un corpus d'environ 50 textes par classe. Le tableau suivant présente la composition du corpus utilisé dans l'élaboration de l'indice

SATO-CALIBRAGE.

Tableau 1
Description du corpus

Classe	Ordre d'enseignement	
	PRIMAIRE	SECONDAIRE
Première année	65	63
Deuxième année	82	43
Troisième année	63	49
Quatrième année	60	57
Cinquième année	60	67
Sixième année	70	-
TOTAL	400	279

Au total, le corpus compte 679 textes.

Le MEQ dispose de grilles permettant d'évaluer la pertinence d'un texte pour une classe d'enseignement donnée ou, à tout le moins, pour un cycle d'enseignement. Parmi les éléments de cette grille, on retrouve des caractéristiques qu'il est possible de détecter à l'aide d'un programme informatique, par exemple la longueur du texte, la longueur des phrases, etc. Notre objectif était donc, dans un premier temps, d'automatiser et de valider certains éléments de cette grille d'évaluation utilisée dans le milieu scolaire. Dans un deuxième temps, l'objectif visé était d'élaborer un protocole expérimental afin d'enrichir le modèle interprétatif.

3.1.2 Élaboration du protocole expérimental

Pour réaliser ce protocole, nous avons retenu le logiciel SATO qui nous permettait de dépister rapidement une variété d'indices textuels. Cet outil informatique représente le texte sur un plan composé de deux axes. On a d'abord un axe lexical qui dresse la liste du vocabulaire utilisé dans le texte. On a ensuite un axe syntagmatique qui restitue la linéarité du texte qui se donne en fait comme une suite d'occurrences des lexèmes.

Destiné à soutenir des activités d'analyse, le logiciel offre aussi la possibilité d'annoter et de catégoriser le texte, permettant ainsi de marquer le dépistage de

processus discursifs; ces marques peuvent s'inscrire sur l'axe lexical ou sur l'axe syntagmatique. Nous utilisons le terme *propriété* pour désigner une classe d'annotations ou de catégories permettant de marquer des lexèmes ou des occurrences.

Les dispositifs expérimentaux utilisés dans ce cadre ont pris la forme de scénarios de commandes qui aboutissent à la production de données quantitatives correspondant à divers indices ou variables: nombre ou proportion de phrases utilisant telle ou telle construction, fréquence d'utilisation de tel lexème ou telle classe de lexèmes. Appliqués sur chacun des textes du corpus, ces scénarios produisent finalement une très grande quantité de données. Il peut arriver que des données puissent être interprétées directement par des spécialistes. Le plus souvent cependant, pour être évaluées correctement, les données doivent être examinées à travers une modélisation mathématique (cf. 3.3).

3.2 Le dispositif linguistique

Nous désignons par dispositif linguistique, l'ensemble des ressources linguistiques déployées à l'intérieur du projet SATO-CALIBRAGE. Ces ressources sont les suivantes: 1) bases de données lexicales ; 2) procédures pour repérer des noms propres ; 3) procédures pour identifier les verbes conjugués ; 4) procédures de dépistage des locutions grammaticales ; 5) et, s'appuyant sur les dispositifs précédents, procédures permettant le dépistage de phrases potentiellement complexes.

3.2.1 Bases de données lexicales

Les bases de données lexicales prennent la forme de dictionnaires. Ce sont des fichiers qui contiennent des informations sur des formes lexicales. En consultant ces dictionnaires, on peut annoter le lexique d'un texte en transférant sur une propriété lexicale les renseignements se trouvant dans le dictionnaire. Deux bases de données sont ici utilisées. La première[3] contient la catégorie grammaticale hors contexte d'environ un demi million de formes lexicales. Quant à la seconde liste, elle a été développée à l'intérieur du projet; il s'agit d'un dictionnaire de mots

familiers aux élèves de sixième année du primaire constitué à partir des formes lexicales rencontrées dans le corpus utilisé pour ce projet.

Le dictionnaire des mots connus par les élèves de sixième année a été constitué en faisant valider le lexique de l'ensemble du corpus par des enseignants de sixième année. Le corpus s'étant enrichi au cours des années, cette validation a dû être faite deux fois pour tenir compte des nouveaux mots. Dans chacun des cas, la validation a été effectuée par un groupe de cinq enseignants d'expérience provenant de régions différentes du Québec et œuvrant dans des milieux sociaux différents. On a accepté comme connus les lexèmes jugés familiers par au moins quatre personnes. La consigne donnée aux enseignants demandait de considérer connu un mot qu'au moins les trois quarts des élèves connaissent à l'oral. Dans certains cas, des vérifications ont été faites auprès des élèves eux-mêmes, afin de vérifier leur connaissance de certains mots.

Notons que la validation des mots a été effectuée sur les formes fléchies des lexèmes, c'est-à-dire dans la forme où ils se présentent dans le texte. Par la suite, nous avons élaboré des dispositifs de fléchissement permettant d'ajouter les flexions régulières des mots connus. En ce qui concerne les verbes, le problème de l'extension de la couverture est plus délicat. Il n'est pas question évidemment d'accepter toutes les formes conjuguées des verbes. Finalement, nous avons convenu de conjuguer les verbes aux temps simples selon les prescriptions du programme de sixième année en écriture[4]. Nous poursuivons notre réflexion sur les critères à retenir pour sélectionner les verbes à conjuguer à partir des formes déjà authentifiées comme connues. Par exemple, si seulement le participe passé a été reconnu, est-ce que l'on peut conclure automatiquement que le verbe est connu dans ses formes conjuguées?

3.2.2 Repérage des noms propres

Même si notre corpus est volumineux, la liste des mots soumis à l'évaluation du milieu n'est pas exhaustive. On verra plus loin que l'on peut compléter la liste par un dictionnaire personnel. Par ailleurs, les noms propres, qu'il n'est pas possible évidemment de rassembler dans une base de données lexicale, peuvent être

considérés connus en raison de leur contexte d'utilisation. Pour faciliter l'identification des noms propres, un scénario a été bâti dans le but de dresser pour chaque texte une liste de noms propres potentiels. Cette liste peut, au choix de l'utilisateur, être présentée pour fins de validation.

Cette liste de noms propres se limite aux mots qui ne font pas déjà partie de la liste des mots connus à titre de nom commun par exemple. On exclut aussi les articles, les pronoms, les adverbes, les prépositions et les conjonctions. Finalement, on dénombre l'utilisation des lexèmes de la liste restante en comptant les graphies qui débutent par une majuscule et en notant les cas où cette utilisation ne fait pas suite à une ponctuation forte telle le point. Dans ce dernier cas, il y a de bonnes chances que le mot soit un nom propre.

Les noms propres ainsi identifiés sont ajoutés à la liste des mots connus. De même, on y ajoute les nombres, ponctuations et séparateurs. La figure 1 présente la moyenne des pourcentages de mots inconnus par classe d'enseignement. Bien que la courbe indique un fléchissement au premier cycle du secondaire, on constate que le pourcentage de mots inconnus croît avec la classe d'enseignement.

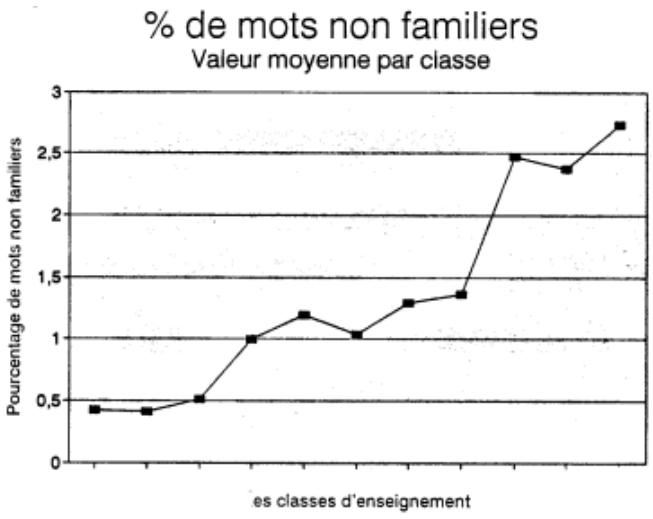


Figure 1

Pourcentage moyen de mots inconnus par classe d'enseignement

3.2.3 Désambiguïsation des verbes conjugués

Il nous est aussi apparu que le décompte du nombre relatif de propositions dans le texte était susceptible de nous donner des indications sur sa complexité. Le dépistage en contexte des verbes conjugués s'est donc avéré nécessaire. Nous avons déjà eu l'occasion de présenter de façon détaillée notre stratégie de désambiguïsation des verbes, cf. Daoust & Dupuis (1996, 1994). Nous nous contenterons donc ici d'un rappel.

À l'aide de SATO, il est possible de décrire, sous forme de patrons de fouille, les contextes désambigüisants. Du même coup, on peut associer à ces patrons des actions de désambigüisation catégorielle. Nous adoptons donc une stratégie de *grammaires locales*, cf. Silberztein (1989). La solution développée ici comporte deux étapes incorporées dans une seule procédure. On a d'abord l'élagage ou la suppression des catégories grammaticales *indésirables*. On a ensuite l'ajout d'une propriété permettant de visualiser le résultat de la règle et de retracer le contexte de son application. Avec le logiciel, on a construit un dispositif permettant d'évaluer la productivité des règles et de voir par quels moyens rendre la grammaire d'émondage plus efficace. On peut comparer toutes les applications réussies de telle ou telle règle ou, à l'inverse, tous les contextes où aucune règle de désambiguïsation ne s'est appliquée. Cela permet d'examiner tous les cas semblables disséminés dans un texte et de rectifier les règles déjà existantes. Cet examen peut aussi conduire à l'ajout de nouvelles règles pour augmenter l'efficacité du système. Ce dispositif, utilisé en phase de validation, a été supprimé dans le module final. Voici un exemple de règle et sa traduction dans une commande au logiciel.

Règle:	Une forme, qui peut être soit un nom soit un verbe conjugué, n'est pas un verbe conjugué si elle est précédée d'une forme qui est strictement une préposition. La préposition peut être suivie facultativement d'un article ou d'un déterminant et d'adjectifs non ambigus.
Exemple:	La femelle construit habituellement son nid sous un tas de larges branches...
Commande:	contexte appliquer ** $*gramr==prép*. **

$*gramr=(art$,dét$)*-*. **
$*gramr=v_conj*gramr=nomc*syntaxe:-v_conj*&*règle:+d1

Explication: La commande *contexte appliquer* effectue le repérage des contextes qui satisfont aux contraintes définies par les filtres dont la définition suit. Le *$* indique qu'il n'y a aucune contrainte sur les caractères du mot ; *gramr* contient la catégorie grammaticale hors-contexte; *==prép* signifie que le mot doit être une préposition non-ambiguë ; *. est un opérateur de proximité qui indique que le filtre suivant est immédiatement adjacent. Le deuxième filtre accepte tout mot qui est un article ou un déterminant ; *- indique que la position peut être vide. Le dernier filtre indique que le mot possède la double catégorie verbe conjugué et nom commun; la propriété *syntaxe* est la projection de *gramr* sur les occurrences ; l'opérateur *:-* indique que l'on veut enlever la catégorie *nomc* à la propriété *syntaxe* ; l'opérateur **&* indique que la catégorie verbale doit être le pôle de la concordance ; finalement, **règle:+d1* indique que l'étiquette *d1* qui identifie la règle doit être apposée sur le lexème désambiguïsé.

La couverture des règles de désambiguïsation pourrait être élargie puisque nous nous sommes limités aux règles les plus productives. On pourrait aussi faire appel à des algorithmes probabilistes, ce qui est de plus en plus la tendance comme en témoigne un numéro récent de la revue de l'ATALA, cf. Habert (1995): chaînes de Markov, N-gram etc. Cependant, comme les textes à analyser sont courts, le nombre de validations manuelles à effectuer, le cas échéant, ne constitue pas un handicap majeur.

La figure 2 à la page suivante permet de visualiser la moyenne des pourcentages par texte de verbes conjugués selon les années d'enseignement. Il apparaît donc que le nombre relatif de verbes conjugués dans un texte est un indice de facilité. Cela correspond probablement à un grand nombre de phrases courtes à construction simple du type sujet-verbe-complément.

Une première série d'analyses statistiques nous a révélé que la difficulté croissante des textes en fonction de la classe d'enseignement semblait subir un fléchissement au niveau du secondaire. En particulier, nos variables ne semblaient pas suffisantes pour caractériser la fin du secondaire. C'est alors que nous avons entrepris, cf. Daoust (1993) de vérifier si certains marqueurs pouvaient rendre compte de structures argumentaires davantage présentes dans les textes des classes les plus avancées. Nous nous sommes donc intéressés aux adverbes, aux prépositions et aux

conjonctions. Comme plusieurs items de ces catégories prennent la forme de locutions, nous avons dû d'abord développer une procédure de reconnaissance de ces locutions. Faisant l'économie d'une analyse grammaticale complète des phrases, nous avons adopté encore une fois une stratégie de grammaires locales. Ce faisant, nous avons choisi d'écarter des expressions trop ambiguës comme *en fait*.

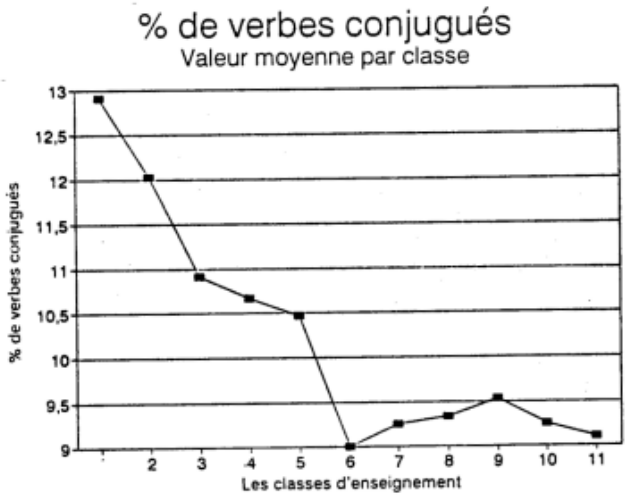


Figure 2
Pourcentage moyen de verbes conjugués par classe d'enseignement

3.2.4 Dépistage des locutions grammaticales

Les lexèmes simples et les locutions ainsi dépistées ont ensuite fait l'objet de diverses manipulations statistiques. Ces analyses nous ont conduit à définir une nouvelle variable basée sur le décompte des lexèmes suivants: *alors que, à l'instant, à présent, au-delà, au-dessous, au-dessus, au-devant, certes, dont, guère, parmi, particulièrement, séparément, toutefois, d'ailleurs, en effet, en vertu de, le long de, tel, telle, telles, tels, vous*. La présence du pronom *vous* peut apparaître étonnante à première vue. Le mot lui-même n'est pas difficile mais il annonce probablement une forme conjuguée plus difficile. On verra plus tard dans la présentation de l'indice que le pronom *tu* a aussi été retenu par les analyses statistiques. L'usage relatif du *tu* diminue avec la classe d'enseignement. À l'inverse du *vous*, il s'agit donc d'un indice de facilité. L'usage contrasté du *tu* et du *vous* est sans doute relié à des considérations psycho-linguistiques, le tutoiement étant souvent relié à un univers familier plus caractéristique des premières classes du

primaire. Les figures 3 et 4 confirment que le pourcentage d'utilisation de ces mots augmente avec la classe d'enseignement alors que le pourcentage d'utilisation du *tu* diminue.

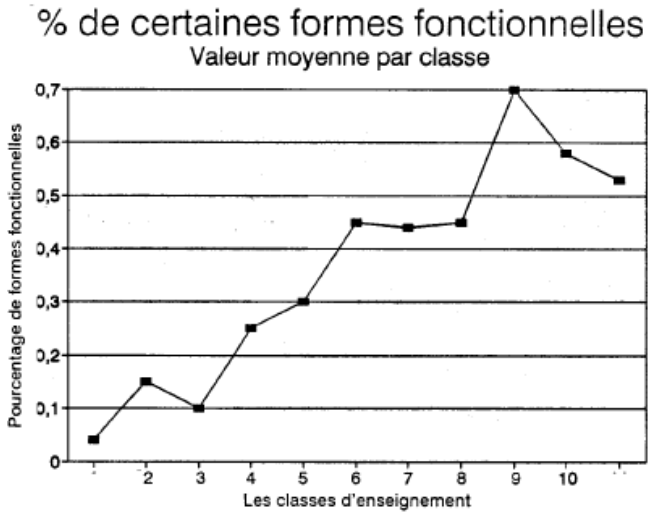


Figure 3
Pourcentage moyen de certaines formes fonctionnelles

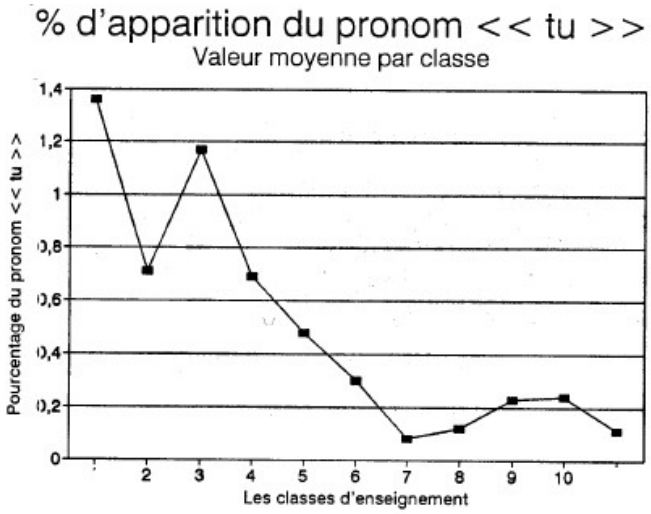


Figure 4
Pourcentage moyen du pronom *tu* par classe d'enseignement

3.2.5 Repérage des phrases complexes

Le dernier élément de notre dispositif linguistique consiste à repérer certaines constructions de phrase susceptibles de contenir des éléments de complexité. Nous

nous sommes contentés ici de repérer des phrases qui contenaient des éléments déjà dépistés par les opérations précédentes. De multiples suggestions nous ont été faites par le comité d'appui. Beaucoup n'ont pas résisté à l'analyse statistique qui, par nature, ne retient pas les situations trop peu fréquentes. Dans le rapport qualitatif produit par le logiciel, nous avons quand même utilisé des diagnostics que l'analyse statistique n'a pas retenus. De façon intuitive à tout le moins, et de l'avis des membres du comité d'appui, ces diagnostics peuvent suggérer des reformulations. C'est le cas par exemple des phrases qui possèdent plus de trois verbes conjugués ou celles qui possèdent des *qui*, *que*, *dont*, etc.

Comme on le verra en 3.3.2, la première variable en importance de l'indice SATO-CALIBRAGE est un indice simple de complexité de phrase, à savoir le pourcentage de phrases de plus de 30 mots (ponctuation incluse). Plusieurs seuils de longueur ont été utilisés mais c'est ce seuil de 30 mots que l'analyse statistique a fait ressortir (figure 5).

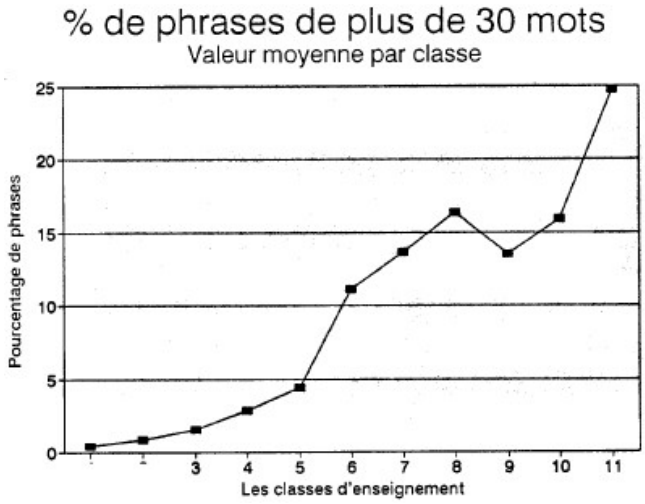


Figure 5
Pourcentage moyen de phrases de plus de 30 mots par classe d'enseignement

Le deuxième variable la plus importante de l'indice vient renforcer ce critère concernant la longueur des phrases. Il s'agit du pourcentage de points. Plus il y a de points dans le texte et plus il est facile. En première année, on a, en moyenne, presqu'un point à tous les 10 mots (ponctuations incluses!). En cinquième année du secondaire, on a un point à tous les 25 mots en moyenne (figure 6).

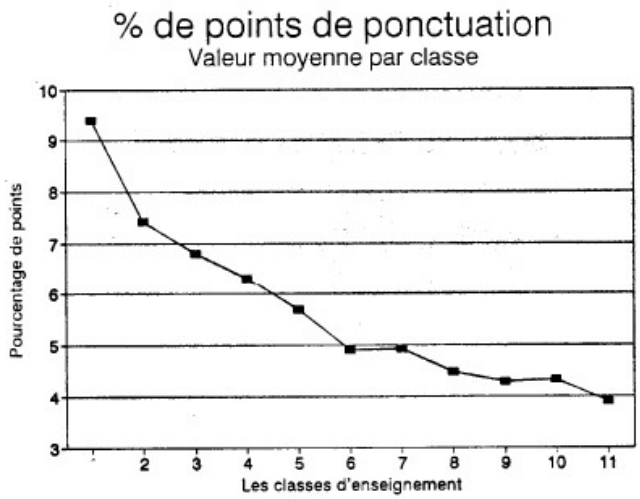


Figure 6
Pourcentage moyen de points par classe d'enseignement

Parmi les autres variables retenues par l'analyse statistique, plusieurs rejoignent l'intuition. Le nombre de phrases croît avec la classe d'enseignement mais avec une retombée étonnante en cinquième année du secondaire. Le point d'exclamation est un élément facilitant caractéristique des trois premières années du primaire. Les pronoms relatifs sont presqu'absents en première année. Leur nombre croît ensuite régulièrement mais connaît une baisse significative au premier cycle du secondaire. Ce même phénomène de fléchissement se produit pour d'autres variables, dont le pourcentage de mots de 9 lettres et plus qui, autrement, croît avec la classe d'enseignement.

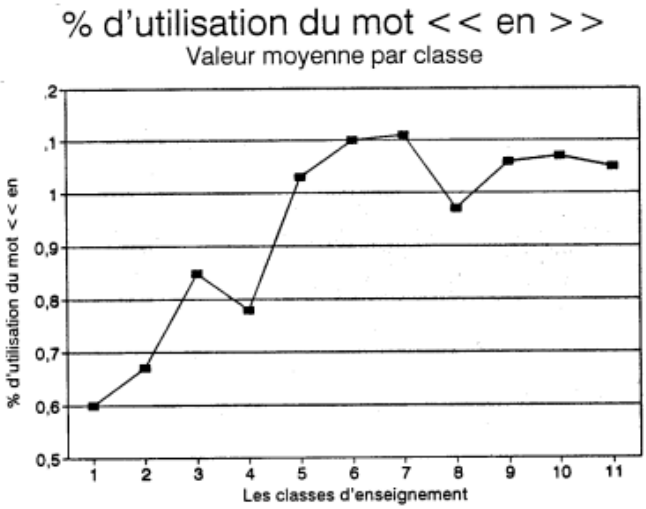


Figure 7

Pourcentage moyen du mot *en* par classe d'enseignement

Deux variables ont été retenues alors que l'on ne s'y attendait pas. Il s'agit des pourcentages d'utilisation des lexèmes *en* et *l'* respectivement. Même si l'importance statistique de ces deux variables n'est pas très grande (cf.tableau 2), elles apparaissent toutes deux comme des éléments de complexité. Il faudrait donc vérifier les contextes pour voir si leur utilisation plus fréquente dans les classes avancées correspond à des constructions de phrases déterminées. Le graphique de *en* semble indiquer une certaine stabilité de son usage à partir de la cinquième année. C'est donc surtout la faible utilisation du *en* dans les premières du primaire qui est digne d'attention (figure 7). Le graphique du *l'* nous laisse plus perplexe. Il faudra sans doute y revenir (figure 8).

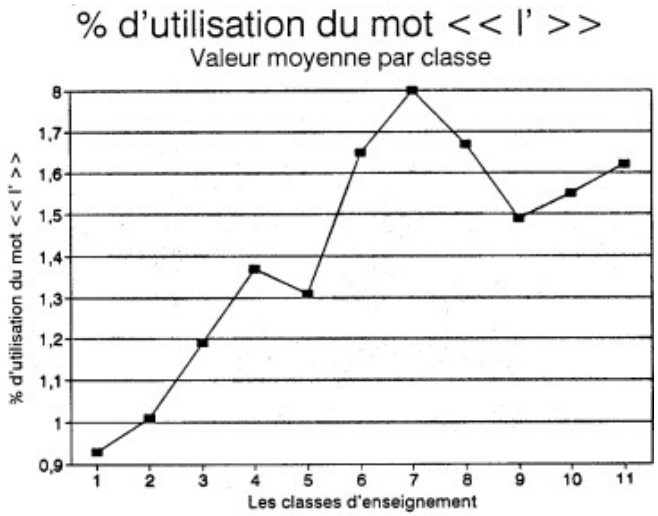


Figure 8
Pourcentage moyen du mot *l'* par classe d'enseignement

3.3 Le dispositif mathématique

L'analyse mathématique a deux objectifs. Il s'agit d'abord d'évaluer l'ampleur et la pertinence de la variation des indices. Il s'agit ensuite de voir comment les divers indices partiels peuvent, en se combinant, produire des indices complexes susceptibles de révéler des régularités appréhendées ou insoupçonnées.

Nous désignons, par dispositif mathématique, l'ensemble des méthodes quantitatives utilisées pour interpréter les indices fournis par SATO. Ces méthodes mathématiques sont utilisées à deux fins. D'abord, on s'en sert pour déterminer les

variables qui varient de façon significative en fonction de la classe d'enseignement. Ainsi, on peut confirmer ou infirmer des hypothèses concernant divers fonctionnements discursifs. Ensuite, on s'en sert pour combiner les indices primitifs significatifs afin de construire des fonctions aptes à prédire le rattachement à une classe d'enseignement.

Dans notre projet, nous avons fait appel à quatre types de modèles mathématiques. D'abord, puisque nous visons à trouver des indices permettant de distinguer les textes selon leur rattachement à des classes d'enseignement, nous avons recours à des tests d'hypothèses (test du Chi-2 en particulier) pour réaliser une première sélection des indices.

En ce qui concerne la constitution des indices basés sur les termes fonctionnels, nous avons voulu réduire le nombre de variables. Pour ce faire, nous avons utilisé deux techniques. Dans la première, nous avons soumis les termes fonctionnels retenus à l'analyse discriminante et avons conservé les termes gardés par l'analyse. Dans la deuxième technique, nous avons d'abord soumis l'ensemble des termes retenus à un algorithme de classement destiné à grouper ceux qui ont des distributions similaires par rapport aux classes d'enseignement, cf. Cucumel (1993), pour une présentation de méthodes de classification.

L'interprétation des groupes ainsi constitués a permis d'éliminer ceux dont le comportement semblait atypique. Elle a aussi permis de garder les autres groupes sous la forme d'indices composites.

Finalement, nous avons élaboré des fonctions prédictives permettant de classer un texte par rapport à une classe d'enseignement. Pour cela, nous avons utilisé les régressions simples et multiples et l'analyse discriminante.

Au terme de ces différentes analyses, il a été possible de définir un indice de difficulté, cf. Laroche (1993), basé sur la réalité du système scolaire québécois. Nous l'avons appelé l'indice SATO-CALIBRAGE.

3.3.1 Élaboration d'un indice

Pour être en mesure de proposer un indice susceptible d'indiquer le niveau de complexité d'un texte, les travaux statistiques sur le corpus se sont déroulés en deux temps. Au préalable, un ensemble de renseignements quantifiés a été produit à l'aide du logiciel SATO. Une première série d'analyses a consisté à produire des compilations univariées: statistiques descriptives, régressions simples, corrélations de Pearson. Dans un deuxième temps, des analyses multivariées ont permis de déterminer les sous-ensembles de variables qui peuvent le mieux expliquer le rattachement d'un texte à une classe d'enseignement.

Rappelons notre objectif. Il s'agit, sur la base des *portraits quantifiés* de chacun des textes du corpus disponible, d'établir un lien entre les variables (indices de difficulté-facilité) et la classe d'enseignement auquel est destiné le texte. Pour ce faire, la régression simple et la corrélation de Pearson ont permis de mesurer l'importance du lien qui s'établit entre deux séries de valeurs. La première série de valeurs représente la classe d'enseignement. Il s'agit d'un nombre entre 1 et 11 correspondant aux six années du primaire et aux cinq années du secondaire. La deuxième série de valeurs représente les résultats calculés à partir du texte pour chacune des variables disponibles.

Les résultats obtenus à la suite des analyses univariées ont permis de retenir les variables les plus fortement liées à la classe d'enseignement. Nous avons pu constater une assez grande linéarité des résultats obtenus au regard de la classe d'enseignement, principalement pour les textes utilisés dans les classes du primaire. Plus de 120 variables ont été utilisées pour réaliser les travaux de cette première étape de nos analyses statistiques. Après ces compilations, 45 variables furent identifiées comme étant assez fortement reliées à la classe d'enseignement.

La seconde étape de l'analyse consiste à examiner le comportement de l'ensemble des variables retenues auparavant afin de déterminer les sous-ensembles qui peuvent le mieux expliquer le rattachement d'un texte à la classe d'enseignement. On sait que plusieurs variables mesurent des aspects semblables de la complexité d'un texte; il s'agit de diminuer cette redondance.

Des analyses factorielles ont permis de regrouper les variables et de les situer par

rapport au rattachement à la classe d'enseignement. Un *facteur* principal a ainsi pu être identifié. Il a été possible de situer certaines variables sur un axe décrivant leur plus ou moins grande *complexité* mesurée par leur proximité à la classe d'enseignement. Cette phase de l'analyse a permis de retenir une trentaine de variables susceptibles de mieux décrire la lisibilité d'un texte.

Comme nous l'indiquions plus haut, parmi les variables disponibles pour ces analyses, plusieurs sont fortement corrélées entre elles, indiquant qu'il y a redondance. Des analyses réalisées à l'aide de la régression multiple tentent justement de diminuer ce phénomène en déterminant le jeu de caractéristiques le plus relié à la complexité des textes mesurée par leur rattachement à la classe d'enseignement. Les résultats obtenus par ces analyses ont rendu possible la fabrication d'un indice de calibrage.

3.3.2 Description de l'indice de calibrage

L'indice SATO-CALIBRAGE existe sous deux versions. La première, et la plus performante, tient compte de la longueur des textes. Plus un texte est long et plus il est susceptible d'appartenir à une classe avancée. Un deuxième indice a aussi été constitué sur la base d'une exclusion volontaire des variables tenant compte de la longueur des textes. Cet indice est moins performant mais est plus susceptible d'être utilisé sur des textes d'une autre nature, tels des romans ou autres textes longs.

Le tableau 2 donne la liste des variables constituant l'indice sensible à la longueur du texte. Cette liste a été produite par un algorithme de régression multiple qui permet d'éliminer les variables redondantes en ne gardant que celles qui sont les plus explicatives de la variance. Les 14 variables conservées permettent d'expliquer 74.2% de la variance. La variance mesure la dispersion de la variable *classe d'enseignement* autour de sa moyenne. Lorsque cette dispersion est calculée autour de la droite de régression (l'indice SATO-CALIBRAGE), elle diminue des trois quarts. C'est donc dire que l'on a pu *expliquer* ou prédire la classe d'enseignement avec une efficacité de près de 75% en utilisant 1es mesures produites par l'analyse des textes.

Le tableau 2 présente l'analyse de la régression multiple hiérarchique sur les variables prédictives de la classe d'enseignement. L'analyse de la régression multiple a été incluse dans cette étude sur la lisibilité afin de vérifier s'il est possible d'obtenir une équation de prédiction qui permettrait de décrire la relation linéaire entre des variables indépendantes et le rattachement des textes à la classe d'enseignement. La valeur de la régression multiple à la seizième étape du calcul est de 0,742. Cela signifie que près de 75% de la variance des valeurs indiquant la classe de rattachement des textes est expliquée par une combinaison linéaire des quatorze variables faisant partie de l'équation de régression.

TABLEAU 2
Analyse de régression sur les variables prédictives de la classe d'enseignement.

Variable	**Variance expliquée**	**Description de la variable**
v1	30,4	% de phrases de plus de 30 mots
v2	44,4	% de points (.)
v3	57,0	% de mots inconnus (non familiers)
	67,7	Nombre total de mots
v4	69,1	% de formes fonctionnelles difficiles + vous
v5	70,0	% de tu
v6	70,8	Nombre de phrases
--	70,7	RETRAIT de la variable «Nombre total de mots»
v7	72,0	% de points d'exclamation (!)
v8	72,6	% de pronoms relatifs
v9	73,2	% de mots de 9 lettres et plus
v10	73,5	% de «en»
v11	73,8	% de «l'»
v12	74,0	% de verbes conjugués
v13	74,1	% d'adjectifs
v14	74,2	% de phrases contenant plusieurs mots non familiers

L'équation de régression construite à partir de ces variables est la suivante:
3.613 + 0.054**v1** - 0.245**v2** + 0.781**v3** + 0.562**v4** - 0.196**v5** + 0.014**v6** - 0.228**v7** + 0.340**v8** + 0.037**v9** + 0.306**v10** + 0.224**v11** 0.081**v12** + 0.048**v13** - 0.018**v14**

Règle générale, un facteur positif dans l'équation indique que la variable est un

indice de difficulté alors qu'un facteur négatif est la marque d'un indice de facilité. L'application de cette équation sur un texte donné nous fournit un estimé de la classe d'enseignement auquel il devrait être destiné, compte tenu de l'analyse des caractéristiques de notre corpus de 679 textes.

La figure 9 représente la courbe de la moyenne par classe de l'indice SATO-CALIBRAGE en fonction du niveau d'enseignement.

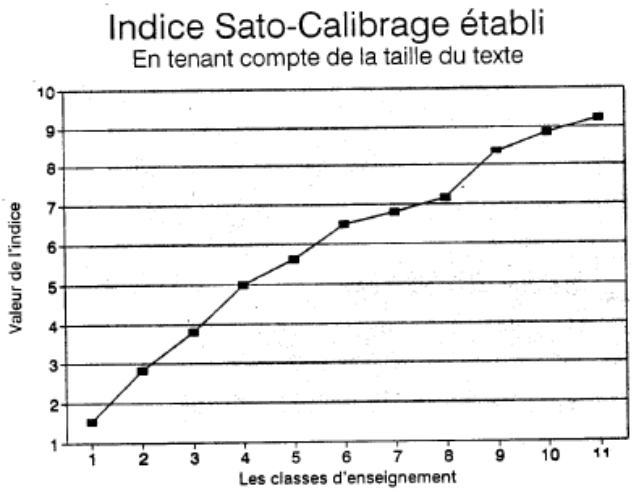


Figure 9

Indice SATO-CALIBRAGE tenant compte de la taille du texte

On peut constater que l'indice est très corrélé avec le niveau d'enseignement. On voit aussi qu'il est généralement linéaire malgré un affaissement progressif de la courbe après la quatrième année. Le rythme d'augmentation de la difficulté des textes a donc tendance à être moins forte au fur et à mesure de l'avancement scolaire. On trouvera dans le Guide de l'usager, cf. Daoust, Laroche & Ouellet (1996), des graphiques sur la répartition par classe d'enseignement de chacune des variables constituant l'indice.

4 Description du prototype

Les divers traitements réalisés par l'application sont les suivants: 1) la *génération* du texte soumis à SATO; 2) son analyse suivi de la production du rapport sommaire de calibrage; 3) et finalement, la production d'un rapport qualitatif.

Le prototype utilise la version courante du logiciel SATO. Ainsi, nous pouvons profiter des améliorations au produit au fur et à mesure de son développement. Un module d'interface a aussi été développé pour faciliter la configuration de l'outil et lancer les traitements.

Ayant choisi le ou les textes à calibrer, les analyses suivantes sont déclenchées.

1- Génération. Il y a tout d'abord *génération* du texte afin d'en produire une version en format interne à SATO. Cette opération consiste à lire le texte linéaire en format caractère afin d'en produire une représentation en deux dimensions: lexique et occurrences.

2- Analyse et rapport sommaire. Le module d'analyse de SATO réalise les traitements suivants.

a) Lecture des commandes de configuration.

b) Catégorisation grammaticale.

La catégorisation grammaticale s'opère par la consultation d'un dictionnaire SATO et par une analyse morphologique des formes particulières telles les nombres.

c) Repérage des mots familiers.

- Création de la propriété *connu*:
  oui: Mots outils les plus simples, noms propres, nombres, etc.;
  p6: Primaire sixième année (liste validée);
  6a: Primaire sixième année (liste personnelle).
- Consultation du dictionnaire des mots connus (6e année).
- Consultation du dictionnaire spécialisé (disciplinaire) des mots connus.
- Attribution de la valeur *oui* aux noms propres, délimiteurs, ponctuation, nombres, articles...
- Consultation du dictionnaire personnel des mots connus.

d) Dépistage des noms propres correspondant à des mots inconnus.

Le dépistage des noms propres a pour objectif principal de marquer ces mots comme connus. En effet, le contexte permet généralement au locuteur de comprendre qu'il s'agit d'un nom de personne, de lieu, etc. Dans cette application, il n'est pas nécessaire de valider les noms propres qui correspondent déjà à une forme

réputée connue. Le dépistage des noms propres est réalisé en comptant le nombre de fois où un mot (excluant les articles, pronoms, etc.) est en majuscule, et le nombre de fois où le mot en majuscule n'est pas précédé d'une ponctuation forte. Une décision est alors suggérée.

e) Validation manuelle des candidats noms propres.

Si cette option a été choisie lors de la configuration du prototype, la liste des candidats noms propres sera présentée pour validation. On peut alors confirmer ou infirmer la décision de SATO et, si nécessaire, voir la forme en contexte.

f) Catégorisation en contexte des verbes conjugués.

SATO-CALIBRAGE fait le décompte du nombre de propositions dans le texte. On identifie la proposition par le verbe conjugué. Comme plusieurs formes lexicales possèdent une catégorie grammaticale ambiguë, par exemple des verbes qui sont homographes avec des noms, des adjectifs, etc., on doit identifier les *vrais* verbes. Les étapes de l'algorithme de dépistage sont les suivantes:

- identification des locutions grammaticales figées;
- application de patrons de concordance avec catégorisation en contexte.

g) Validation manuelle des verbes encore ambigus.

Même si les règles de grammaires locales permettent de lever la majorité des ambiguïtés sur le verbe, il reste en général un certain nombre d'ambiguïtés. Si l'option de validation a été choisie lors de la configuration du prototype, les verbes encore ambigus sont présentés pour une catégorisation manuelle.

h) Identification des indices de complexité.

À partir du marquage déjà réalisé, il est possible de dépister diverses configurations susceptibles de représenter des difficultés. Les contextes dépistés sont alors marqués par une annotation inscrite dans la propriété contextuelle DIAGNOSTIC. Voici la liste des diagnostics dépistés sur les phrases.

4 verbes:	la phrase possède au moins quatre verbes conjugués;
31 mots et Plus:	La phrase contient plus de 30 mots;
Prorel-Con:	La phrase possède un mot qui, au dictionnaire, peut être un

	pronom relatif (*qui*, *que*, *dont*, etc.). La phrase est affichée même si, en contexte, le mot agit comme conjonction.
2 mots inconnus:	La phrase contient au moins deux mots inconnus.

i) Production des données quantitatives utilisées pour le calcul de l'indice SATO-CALIBRAGE (rapport sommaire). Le calcul de l'indice SATO-CALIBRAGE se fait par un programme externe à partir des données numériques produites par SATO.

3- Rapport qualitatif. Généralement, on fait suivre la production de l'indice d'un rapport qualitatif de calibrage. Ce rapport contient les éléments suivants :

- le lexique des noms, des adjectifs et des verbes apparaissant plus d'une fois ;
- la longueur des mots, des phrases et des paragraphes et l'indice de Gunning[5] ;
- la répartition des lexèmes par rapport aux listes de mots connus ;
- la liste des mots identifiés comme inconnus; l'utilisateur peut alors inscrire des lexèmes dans son dictionnaire personnel des mots connus ;
- la liste des mots longs (9 lettres ou plus) ;
- la liste des phrases susceptibles de contenir des éléments de complexité (cf. étape h).

Il est aussi possible de compléter l'analyse du texte en faisant appel à des scénarios spécialisés. En voici quelques exemples.

Lex_Tot	Lexique total trié par ordre alphabétique;
Lex_Det	Lexique des déterminants;
Lex_Lien	Lexique des mots de liaison;
Lex_Pp	Lexique des pronoms personnels;
Phr_Adj	Phrases contenant des adjectifs qualificatifs (hors contexte);
Phr_Ffd	Phrases contenant des formes fonctionnelles difficiles;
Phr_2Pp	Phrases contenant au moins deux pronoms personnels;
Phr_Prel	Phrases contenant des pronoms relatifs (hors contexte);
Phr_Conj	Phrases contenant des verbes conjugués;
Phr_Inc	Phrases contenant des mots inconnus;
Phr_9Let	Phrases contenant des mots de 9 lettres et plus;
Phr_Lon	Phrases dépassant une longueur fournie par l'utilisateur;
Phr_En	Phrases contenant *en*;
Phr_L	Phrases contenant *l'*;
Phr_Toi	Phrases contenant *toi*;

Phr_Tu Phrases contenant *tu*;
Phr_Vous Phrases contenant *vous*;
Phr_Excl Phrases contenant'!';
Phr_Mot Phrases contenant un mot fourni par l'utilisateur;
Phr_Son Phrases contenant une syllabe fournie par l'utilisateur;
Syn_Decr Statistique descriptive de la propriété Syntaxe.

L'application SATO-CALIBRAGE étant *ouverte* il est possible de la compléter et de la personnaliser à volonté. Ainsi, un utilisateur pourrait s'initier au logiciel SATO et ajouter une variété d'analyseurs pouvant fournir des avis spécialisés. Par exemple, on pourrait vouloir développer des analyses spécifiques pour identifier des difficultés qui correspondent à des populations dont la langue maternelle n'est pas le français.

5 Quelques illustrations

[Remarque Cette section n'est pas reproduite ici mais pourra être consultée dans la version Web de l'article]

6 Conclusion

[Remarque Cette section n'est pas reproduite ici mais pourra être consultée dans la version Web de l'article]

Notes.

[1] La féminisation des titres présente dans la version originale de cet article n'a pas été retenue par la rédaction.
[2] Pour une présentation de la problématique de la lisibilité, cf. Gélinas-Chebat & coll. (1993). Pour une brève recension des indices les plus connus, cf. Laroche (1990).
[3] Cette base de données, appelée couramment *la BDL*, a été développée au départ par Luc Dupuy dans le cadre du projet SACAO (Système d'analyse de contenu assistée par ordinateur), Programme Actions spontanées, FCAR 1989-1991; ce projet était dirigé par Jules Duchastel alors qu'il était directeur du Centre d'ATO de l'UQAM. La version de la BDL utilisée jusqu'à maintenant ne contient que les catégories majeures sans trait de genre, nombre ou temps et personne. La nouvelle version de la BDL contient ces traits ainsi que le lemme des formes fléchies. Il serait sans doute intéressant, pour une expérimentation future, de vérifier la prédictivité de ces nouvelles variables sur la classe d'appartenance des textes.
[4] Le programme du MEQ prescrit «la formation des temps simples (radical et

terminaison) à l'indicatif présent, à l'imparfait, au futur simple, au conditionnel présent, à l'infinitif présent, au participe passé, au subjonctif présent (quand la forme est différente de celle de l'indicatif présent) et au passé simple (3e personne du singulier et du pluriel)».

[5] L'indice GUNNING se calcule par la formule suivante: (longueur moyenne des phrases + % de mots de 9 lettres et plus) x 0,4.

Références

[*Remarque. On trouvera les références bibliographiques de l'article dans la bibliographie du chapitre*]

Post-scriptum

Pour illustrer notre propos, nous avons soumis le présent article à SATO-CALIBRAGE.

[Remarque Cette section n'est pas reproduite ici mais pourra être consultée dans la version Web de l'article]

Le développement commandité de SATO-calibrage s'est déroulé de 1989 à 1993 et a permis de livrer une version de l'application dans l'environnement PC-DOS de l'époque. Il avait été prévu du départ de bonifier le système par l'ajout de nouveaux corpus d'apprentissage tenant compte de l'évolution des programmes d'enseignement et du biais disciplinaire. Malheureusement, le désengagement de l'état québécois suite à la crise des finances publiques a coupé court à ces projets. Malgré tout, grâce à une commandite du ministère de l'Éducation en 1998, nous avons entrepris la conversion vers la version Web de SATO. L'application fait maintenant partie intégrante de l'interface de SATO.

Depuis son développement initial, SATO-calibrage a été utilisé dans diverses situations éloignées du contexte initial décrit dans l'article publié dans la Revue québécoise de linguistique. Ainsi, dans une de nos dernières collaborations actives avec le ministère de l'Éducation, une enseignante a proposé d'utiliser SATO-calibrage pour évaluer la production écrite d'élèves dans une classe de français langue seconde du réseau anglophone d'enseignement. SATO-calibrage est alors apparu comme une mesure de compétence linguistique des élèves et comme un diagnostic des difficultés spécifiques des élèves, concernant, par exemple, le biais engendré par l'influence de la langue maternelle. Moins un

élève maitrise la langue et plus l'indice SATO-calibrage est élevé. En effet, s'il maitrise mal la ponctuation, il risque de produire des phrases très longues. S'il ne maitrise pas l'orthographe d'usage, le nombre de mots absents du dictionnaire des mots connus sera élevé. La lecture du lexique de la composition de l'élève est susceptible de montrer des erreurs qui varient selon la langue maternelle de l'élève. La variété des contextes d'utilisation de SATO-calibrage nous a plusieurs fois étonnés : évaluation d'articles de journaux, d'affidavits, de messages informatifs destinées à divers publics. Avec la linguiste Claire Gélinas Chebat, nous avons évalué la lisibilité (l'illisibilité!) de formulaires d'impôts au bénéfice du vérificateur général du Québec.

SATO-Calibrage a été une occasion de mettre au point des dispositifs d'analyse linguistique comme les règles de désambigüisation syntaxique évoquées dans l'article de la RQL et sur lesquelles nous reviendrons à la section suivante consacrée au projet AlexATO. Il en est de même de la base de données lexicales que nous utilisons en analyse de discours.

La procédure de génération de la base de données lexicales n'ayant pas fait l'objet de publication, il convient ici de la décrire brièvement.



Procédure de génération de la base de données lexicales (4.6b, notice technique)

La base de données lexicales (BDL) est un dictionnaire de formes fléchies généré à partir de corpus de lemmes en format SATO. L'objectif de ce dictionnaire est d'abord de fournir des catégories grammaticales et des lemmes pour la catégorisation hors contexte des formes lexicales de corpus textuels. Mais, il permet aussi de propager d'autres catégories, par exemple, le degré de familiarité d'un mot, sa dérivation en tant que titre *féminisable*, etc.

La BDL a été développée essentiellement pour supporter des pratiques d'analyse lexicale ou contextuelle qui prennent appui sur les fonctions potentielles, grammaticales ou autres, des formes lexicales. La BDL n'a pas d'objectif d'exhaustivité. Mis à part les lemmes verbaux extraits du Bescherelle, les lemmes utilisés pour générer la BDL sont largement issus de formes rencontrées dans des corpus ayant été utilisés en analyse. Des entrées de dictionnaires de langue ont aussi été utilisées à une certaine époque. Mais on a réalisé que la surabondance de mots ne faisant plus partie de l'usage courant avait pour effet d'alourdir les

dictionnaires et d'introduire des usages catégoriels rarement employés dans les univers discursifs faisant partie de nos applications régulières d'analyse. On a donc procédé à un élagage de ces entrées pour ne conserver que les lemmes réputés connus par un public universitaire de lettres françaises.

L'application des dictionnaires générés à partir de la BDL sur le lexique des corpus est intégrée à des scénarios qui appliquent également des règles exploitant la morphologie des formes lexicales, par exemple, pour les nombres. D'autre scénarios peuvent être utilisés pour le dépistage des noms propres par exemple. Par conséquent, l'usager peut valider le résultat de l'application des scénarios sur le lexique du corpus et en compléter les entrées. L'application des dictionnaires est donc une étape dans une chaine de traitement facultative et plus ou moins complexe selon les besoins de l'analyse : dépistage des marqueurs d'argumentation, des formes non grammaticales, des noms propres, des mots étrangers, des locutions grammaticales et terminologiques, etc. Dans ce contexte, il nous est apparu qu'une base de données lexicales centrée sur un vocabulaire de base riche, mais quand même d'usage général, était l'outil le plus pertinent pour nos objectifs d'analyse de corpus. Cela dit, la stratégie mise en place pour la génération de la BDL pouvant être déployée sur une variété de bases de lemmes, il est facile de constituer des BDL spécialisées pour des fins particulières. Puisque les bases de lemmes sont d'abord considérées comme des corpus, l'enjeu ici n'est pas tant leur contenu, dicté par des circonstances et des finalités particulières, que les stratégies et les outils pour les transformer en base de données lexicales et en dictionnaires pouvant être déployés sur le lexique des corpus.

Deux types de ressources sont mises à contribution pour la génération de la base de données lexicales.

Il y a d'abord des listes de lemmes qui se présentent comme des corpus annotés par des propriétés, forme simplifiée de traits telle que gérée par le logiciel SATO. Pour la version de la BDL documentée ici, les corpus utilisés sont les suivants:

- Adjectifs et noms communs : *adjncoi.sat* ;
- Formes fonctionnelles : *fonc.sat* ;
- Verbes infinitifs : *vinf.sat* ;

- Verbes défectifs : *vinfdef.sat*.

Le deuxième type de ressources consiste en un ensemble de scénarios de commandes SATO permettant de dériver les lemmes et de les ajouter à la base de données lexicales sous forme d'entrées dans un fichier séquentiel indexé. Dans ce fichier, les blocs d'information sont coiffés d'une clé qui correspond à la graphie d'une forme fléchie d'un lemme, le lemme étant lui-même une entrée du dictionnaire. Sous cette clé, on retrouve un certain nombre de champs qui contiennent les informations associées à la forme graphique.

Avant d'expliciter l'articulation entre ces deux types de ressources, il convient de préciser en quoi notre stratégie de génération de dictionnaires diffère de celle développée par le LADL, le laboratoire de Maurice Gross, à peu près dans les mêmes années. Pour ce faire, nous nous référerons à cet article de 1990 écrit par Blandine Courtois qui décrit le dictionnaire DELAS des mots simples du français et le DELAF qui contient les formes fléchies.

D'abord, nous nous écartons un peu de la définition de *mot simple* donnée pour le DELAS. Cette définition, qualifiée de *purement formelle*, exclut de la base toute chaine contenant un séparateur, trait d'union, apostrophe, espace blanc, etc. Dans notre cas, nous admettrons des formes contenant des séparateurs, y compris l'espace, pourvu qu'elles soient dérivables par simple suffixation. Ainsi, *arc-en-ciel* (*arcs-en-ciel*) ne sera pas admis alors que *presse-papier* (presse-papiers) et *presqu'ile* (*presqu'iles*) le seraient.

La deuxième différence par rapport à la stratégie de dérivation du LADL, c'est l'absence de l'utilisation de codes morphologiques, sauf pour les verbes où on utilise les numéros de table du Bescherelle. Comme le définit Courtois, le code morphologique « fournit, en un format condensé, la représentation complète de la morphologie de tout mot variable » (Courtois 1990:14). Plutôt que d'adopter une stratégie de classes flexionnelles, il nous est apparu plus simple de suivre le fonctionnement descriptif des grammaires qui distinguent les règles générales et les exceptions. Notre approche consiste donc à appliquer les règles générales de la grammaire aux catégories appropriées, selon leur terminaison, et à appliquer

ensuite une liste d'exceptions selon les terminaisons. Ainsi, on peut ajouter des lemmes dans les corpus sans se soucier de leur classe morphologique, pour peu qu'on se soit assuré qu'aucune règle particulière ne s'applique, ce qui est généralement le cas.

Considérant nos objectifs et nos moyens modestes, cette façon de faire facilite l'entretien de la base, ce qui absolument essentiel. Il est, en effet, relativement aisé de suivre la logique d'application des scénarios de commandes pour identifier les règles pertinentes à des formes données. Et il est facile de modifier les scénarios qui sont en mode texte explicite. Dans le cas des verbes, l'accès au Bescherelle étant généralisé, il n'est pas difficile de s'y référer. Et comme les nouveaux verbes du lexique français sont généralement réguliers, l'entretien du corpus des verbes ne pose pas de problèmes particuliers. On a constaté cependant que les numéros de table du Bescherelle peuvent varier d'une édition à l'autre. On envisage donc dans l'avenir d'utiliser le verbe modèle comme étiquette plutôt que le numéro de la table dans le Bescherelle pour faciliter l'entretien des procédures.

Une autre différence de notre façon de faire par rapport à celle du DELAS concerne l'unicité des entrées de même forme graphique. Dans le DELAS, l'entrée pour *moule* est unique, mais contient deux codes morphologiques (*N1* et *N21*) selon qu'il s'agisse du *moule* et de la *moule.* Dans notre cas, nous aurons deux entrées qui se distinguent non pas par leur forme graphique, mais par la propriété associée : *moule*Nom=Nomcomms* et *moule*Nom=Nomcomfs*. Les scénarios de dérivation vont générer *moules*Nom=Nomcommp* et *moules*Nom=Nomcomfp*. Comme il s'agit d'homographes, on aura dans le dictionnaire des formes fléchies l'entrée *moules*Nom=(Nomcommp, Nomcomfp)*.

Concernant la génération des formes fléchies, notre stratégie suit le schéma classique consistant à retrouver le radical du mot en supprimant, si nécessaire, des caractères terminaux du lemme et à ajouter au radical le suffixe requis pour générer une flexion. S'il y a une différence par rapport à la stratégie du LADL, c'est que chaque scénario est dédié à une et une seule catégorie flexionnelle. Par exemple, plutôt que d'avoir un scénario de règles pour conjuguer toutes les formes d'une

classe de verbes, on aura un scénario pour générer un *mode-temps-personne-nombre* pour l'ensemble des classes de verbes. Comme chaque scénario sait à quoi il est destiné, il pourra ajouter à toutes les formes qu'il conjugue les bons traits de *mode-temps-personne-nombre* sans qu'il soit nécessaire de consigner les traits avec chacun des suffixes appliqués.

Aussi, la forme finale du dictionnaire est différente de celle du DELAS puisqu'elle n'est pas conservée en mode texte. Cela dit, la BDL pourra être exportée dans différents formats textuels, notamment le format TEI-ISO de structures de traits.

À titre d'illustration, voici le début du corpus du fichier SATO des lemmes nominaux et adjectivaux.

```
alphabet fr

propriété Adj symbolique pour lexique Adjdémfp Adjdémfs Adjdémmp
Adjdémms Adjexcfp Adjexcfs Adjexcmp Adjexcms Adjindfp Adjindfs Adjindmp
Adjindms Adjintfp Adjintfs Adjintmp Adjintms Adjnumfp Adjnumfs Adjnummp
Adjnumms Adjposfp Adjposfs Adjposmp Adjposms Adjquafp Adjquafs Adjquamp
Adjquams Adjrelfp Adjrelfs Adjrelmp Adjrelms Vparpasfp Vparpasfs
Vparpasmp Vparpasms

propriété Nom symbolique pour lexique Nomcomfp Nomcomfs Nomcommp
Nomcomms Prodémfp Prodémfs Prodémmp Prodémms Proindfp Proindfs Proindmp
Proindms Prointfp Prointfs Prointmp Prointms Properfp Properfs Propermp
Properms Proposfp Proposfs Proposmp Proposms Prorelfp Prorelfs Prorelmp
Prorelms Fém

propriété Statut symbolique pour lexique p6 Qué Fém
Titre Noms communs et adjectifs qualificatifs

abaissable*Adj=Adjquams
abaissant*Adj=Adjquams
abaisse*Nom=Nomcomfs
abaissement*Nom=Nomcomms
abandon*Nom=Nomcomms*Statut=p6
abandonnique*Adj=Adjquams
abasourdissant*Adj=Adjquams
abasourdissement*Nom=Nomcomms
abat*Nom=Nomcomms*Statut=p6
abâtardissement*Nom=Nomcomms
abatis*Nom=Nomcomms
abattage*Nom=Nomcomms*Statut=p6
abattement*Nom=Nomcomms
abatteur*Nom=Nomcomms*Statut=Fém
```

Comme on le voit, le fichier est composé des entrées non marquées accompagnées de propriétés qui en défissent les traits. Ainsi, la propriété *Adj* donne la catégorie des adjectifs en termes de nombre et de genre. La propriété *Nom* a la même fonction pour les noms. La propriété *Statut* fournit des informations

complémentaires indiquant si l'entrée fait partie des mots connus en sixième année validés pour SATO-calibrage. On indique aussi pour les noms agissant à titre de fonctions, si ce titre est *féminisable*. On indique aussi s'il s'agit d'une entrée correspondant spécifiquement au français du Québec.

Sur ce fichier soumis à SATO, on applique le scénario de génération des formes fléchies. Ce scénario de commandes SATO définit les diverses propriétés : *Gramr, Lemme, Suffixe*. Il appelle ensuite, tour à tour des scénarios généraux de suffixation et des scénarios d'exceptions. À partir de ces informations, le scénario procède à la *g*énération des formes fléchies qui sont ajoutées au dictionnaire par le mécanisme de suffixation.

Voici, par exemple, le scénario général de suffixations des formes plurielles.

```
* !Description: Procédure pour ajouter les suffixes pluriels
* !Description: à un lexique de formes au masculin
* !Auteurs: François Daoust, Fernande Dupuis (Janvier 1995)

* Terminaison
a,b,c,d,e,é,è,ê,ë,f,g,h,i,ï,j,k,l,m,n,o,ô,ö,p,q,r,t,u,û,ü,v,w,y
Propriété attribuer mp = valeur s       pour **
$(a,b,c,d,e,é,è,ê,ë,f,g,h,i,ï,j,k,l,m,n,o,ô,ö,p,q,r,t,u,û,ü,v,w,y)
* Terminaison l
Propriété attribuer mp = valeur 1ux     pour $al

* Terminaison s
Propriété attribuer mp = valeur 0       pour $s

* Terminaison u
Propriété attribuer mp = valeur x       pour $(a,e)u

* Terminaison x
Propriété attribuer mp = valeur 0       pour $x

* Terminaison z
Propriété attribuer mp = valeur 0       pour $z
```

Le scénario est composé d'une série de commandes d'affectation de valeurs à la propriété *mp* des suffixes masculins pluriels. Par exemple, les mots qui se terminent par *al* (filtre SATO *$al*) recevront la valeur de suffixation *1ux* indiquant qu'au moment de l'ajout d'entrées au dictionnaire, SATO devra enlever le dernier caractère à la forme non marquée pour ensuite lui ajouter le suffixe *ux*. Pour les exceptions, s'il y en a, on procèdera de la même façon en écrasant le suffixe général par le suffixe approprié pour des mots particuliers, par exemple *Propriété attribuer mp = valeur (s,1ux) pour (final,idéal)*.

L'ajout des formes fléchies au dictionnaire se réalise grâce à des commandes SATO du type.

```
Dictionnaire indexé attribuer bdl champ Adj Propriété Adj pour $*mp~nil
Ajouter suffixe mp
```

La commande ajoute au champ *Adj* du dictionnaire indexé *bdl* la valeur de la propriété *Adj* pour les entrées du corpus qui ont une valeur de suffixation au masculin pluriel dans la propriété *mp*.

Le projet SATO-calibrage illustre bien la notion de *lecture électronique*. L'analyse de corpus a permis la mise au point de dispositifs d'analyse que l'on déploie sur de nouveaux textes dans le cadre d'une chaine de traitement totalement documentée et transparente. L'interprétation des mesures et avis produits par la procédure d'analyse peut donc être contextualisée et nuancée de façon appropriée.

4.7 Projet AlexATO : développement d'un modèle de traitement coopératif

Le projet ALEX était une commandite privée de la compagnie ALEX destinée à valoriser les machines à traitement parallèle *VOLVOX* constituées d'un ensemble de processeurs autonomes appelés *transputers*. Dans le cadre de ce projet, au cours des années 1992 et 1993, une équipe s'est constituée afin de porter le logiciel SATO sur ordinateur parallèle et d'expérimenter des fonctionnalités nouvelles susceptibles de profiter de cette architecture d'ordinateurs à processeurs multiples. Dirigée par Jules Duchastel, l'équipe regroupait plusieurs chercheurs qui avaient développé une expertise dans l'utilisation de SATO à diverses fins d'analyse textuelle : Josiane Ayoub (philosophie), Gilles Bourque (sociologie), François Daoust (Centre ATO) et Monique Lemieux (linguistique) à l'UQAM; Suzanne Bertrand-Gastaldy, à l'Université de Montréal. François Daoust agira à titre de chef de projet pour tout le volet développement.

Le transfert de SATO sur les ordinateurs VOLVOX visait à augmenter de façon considérable les capacités de traitement de SATO, implanté sur les ordinateurs séquentiels de l'époque, d'abord en termes de volume de textes, mais aussi en termes de complexité. L'objectif du projet était aussi d'expérimenter des fonctionnalités nouvelles nécessitant une puissance de

calcul plus importante. Il s'agissait, en particulier, d'un gestionnaire de segments textuels et de modules de désambigüisation catégorielle.

L'utilisation des processeurs parallèles ne signifiait pas que tout le logiciel devait fonctionner sur ces processeurs. En particulier, l'interface utilisateur devait continuer à utiliser un ordinateur séquentiel conventionnel. L'objectif du projet visait donc à déterminer les processus de traitement (modules) impliqués par la *parallélisation* de SATO, le mode d'échange entre les modules séquentiels et parallèles et les environnements de programmation capables de soutenir ces divers modules. Pour réaliser une implantation de SATO qui puisse bénéficier des ordinateurs parallèles, nous avions conçu un **modèle de traitement coopératif** qui visait à ne transférer sur les ordinateurs à traitement parallèle que les fonctions exigeant le plus de ressources de calcul. Dans la perspective d'une utilisation pleinement intégrée d'un SATO parallèle, on devait envisager la transformation du SATO mono-poste, tel qu'on le connaissait à l'époque, vers un système multi-requêtes et multi-utilisateurs.

Sur la base du modèle de traitement qui vient d'être décrit, notre objectif était de rendre disponibles de nouvelles fonctionnalités de traitement exploitant la puissance du traitement en parallèle. D'une part, nous voulions appliquer un modèle de réseau de neurones (Bégin et Proulx, 1996) aux problèmes de catégorisation linguistique et socio-sémantique de l'analyse textuelle. D'autre part, on voulait développer un modèle de gestion de segments dynamiques catégorisés étendant les capacités algorithmiques de SATO.

En 1992, nous pensions donc déjà à ajouter au modèle *lexèmes/occurrences* de SATO une couche nouvelle permettant de gérer de façon autonome des segments textuels représentés comme des structures de référence aux objets SATO existants. Le segment, tel que défini dans le devis de l'époque était constitué de deux parties logiques : la référence (numéro d'ordre des premier et dernier mots du segment), et la catégorisation afférente à la structure.

Ces segments textuels devaient permettre d'ajouter une nouvelle dimension aux processus de catégorisation. Comme le présentait le devis de 1992, ces processus de catégorisation constituent une des activités centrales, avec la comparaison, de l'analyse de texte par ordinateur. Or, la segmentation est intimement liée à la catégorisation. En segmentant, on distingue des unités et la distinction est, la plupart du temps, marquée par une activité symbolique de *nominalisation* (applications de l'espace des segments à des espaces catégoriels) ou de *coréférence* (applications de l'espace des segments à l'espace des segments). La modélisation en termes de segments devait

permettre de fournir un cadre cohérent pour la rétention informatique des structures syntagmatiques et textuelles (macro-structure, structure argumentaire, etc.). Comme nous l'écrivions à l'époque, « la généralisation prévisible des normes de marquage de type SGML va poser avec acuité la question de la représentation informatique, à des fins de traitement et d'analyse, des segments marqués. SGML, en effet, fournit des outils de balisage du texte, mais ne constitue pas un modèle de traitement des unités marquées. » Depuis, XML a remplacé SGML, mais la pertinence du propos demeure la même, près de 20 ans plus tard.

La généralité du mode de représentation que constitue le concept de segment textuel dynamique, c'est-à-dire constitué en cours d'analyse, pose le problème de son implantation efficace sur ordinateur. Compte-tenu de l'exigence de calcul nécessaire pour supporter le nouveau modèle, le traitement parallèle était vu comme une partie de la solution au problème de la *représentation spatiale* des segments en termes de graphes construits sur le texte.



Les limites de l'architecture VOLVOX (4.7a, notice technique)

La réalisation de nos objectifs de recherche s'est butée très rapidement aux limites importantes de l'architecture des ordinateurs VOLVOX et à la faiblesse des outils de programmation qui nous étaient fournis.

Les ordinateurs VOLVOX sont constitués d'un réseau de *transputers* qui constituent autant d'ordinateurs autonomes qui communiquent entre eux par des liens sériels. L'architecture de ces ordinateurs contenait une faiblesse importante pour des applications telles SATO, à savoir le caractère très limité de l'accès à la mémoire de masse, en particulier les fichiers sur disque. À cet accès limité s'ajoutait le problème de l'optimisation de la communication entres les *transputers*. En effet, dans l'architecture des ordinateurs VOLVOX, il n'y avait qu'un *transputer* qui avait accès aux fonctions d'entrée-sortie en acheminant ses requêtes à l'ordinateur hôte. Pour accéder à ce *transputer* appelée *racine* ou *transputer* primaire, les autres *transputers* devaient utiliser des canaux de communication sérielle. Donc, toute charge d'entrée-sortie se traduit à la fois par une charge de communication entre les *transputers* et par une surcharge de la racine qui était la seule voie d'accès à la mémoire de masse et aux fonctions d'affichage. Donc, des

tâches exigeant beaucoup d'entrées-sorties étaient susceptibles d'être constamment en attente.

Dans la configuration qui nous avait été fournie, le seul serveur utilisable pour accéder au VOLVOX était le SUN sous Unix (Solaris). Nous avions un compilateur PASCAL destiné à dans l'environnement VOLVOX. Mais, il s'agissait d'un PASCAL de bas niveau offrant peu de facilités de programmation. Les modules, une fois compilés sur le SUN devaient être transférés sur les VOLVOX accompagnés d'une configuration définissant les canaux de communication entre *transputers*. Un fichier de configuration devait définir les processeurs utilisés et les liens physiques qui devaient exister entre eux (partie matérielle), ainsi que les tâches qui devaient être exécutées, leur placement, et les caractéristiques de leur environnement d'exécution : mémoire utilisée, canaux logiques de communication entre processus (partie logicielle). Dans l'environnement logiciel dont nous disposions, seuls les processus situés sur des *transputers* voisins --dans la configuration matérielle définie par l'utilisateur-- pouvaient communiquer directement, les liens logiques devant nécessairement avoir un support physique.

Bref, tout cela était très compliqué et, finalement, très inefficace.

Malgré tous nos efforts pour surmonter les limites techniques de l'environnement VOLVOX, on n'a pas abouti à une implantation vraiment opérationnelle. D'ailleurs l'utilisation de ces ordinateurs a finalement été abandonnée par l'Université à la fin de la commandite de recherche et l'entreprise commanditaire a finalement disparu. Cela dit, les modèles que nous avons développés à l'occasion de ce projet ont pu être repris des années plus tard dans le projet d'infrastructure développé pour la Chaire MCD de Jules Duchastel.

Du côté applicatif, le projet aura permis de réaliser une première modélisation des segments dynamiques, modélisation qui sera reprise et développée dans le contexte de la présente recherche doctorale.

Au cours de ce projet, avec notre collègue linguiste Fernande Dupuis, nous avons formalisé une classe de problèmes de catégorisation sur laquelle nous avons bâti un protocole formel

d'expérimentation et de validation ayant fait l'objet de publications (Daoust et Dupuis 1996). L'expérimentation a porté sur la catégorisation grammaticale en contexte à l'aide du modèle *connexionniste* **EIDOS** de nos collègues de psychologie (Bégin et Proulx, 1996).

En bref, le protocole peut être résumé dans les termes suivants.

Nous disposons de dictionnaires permettant d'associer des catégories à des lexèmes (mots hors contexte). En rapportant la catégorisation hors contexte sur les mots en contexte, on obtient donc des séquences de mots (occurrences) catégorisés. On applique à ces séquences un ensemble de règles déterministes de type *grammaires locales* (Silberztein 1989). Ces règles de désambigüisation, accompagnées d'un mécanisme de trace, font appel au dispositif de patrons de concordances SATO. On soumet ensuite ces contextes désambigüisés, autour d'une catégorie pivot, au réseau connexionniste en phase d'apprentissage afin de stabiliser ses paramètres. Par la suite, on soumet au système des contextes non désambigüisés afin que le réseau lève les ambigüités. Ce protocole a l'avantage ici de permettre la comparaison, sur un même corpus, de deux types de modèles, un modèle à base de règles et un modèle de type *réseau de neurones*. Ce protocole expérimental a aussi été repris pour comparer le même modèle à base de règles en SATO et le modèle probabiliste de Brill (1992) : voir Prévost, Heiden, Dupuis, 2000.

Pour notre protocole expérimental, nous avons choisi de nous centrer sur l'analyse de la position verbale. Parmi l'ensemble des règles du dispositif linguistique, nous avons choisi une des règles les plus productives et dont la représentation, en termes de séquence de catégories, était la plus concise. L'implantation du modèle EIDOS dans le cadre du projet VOLVOX prenait comme modèle d'expérimentation la reconnaissance d'images à partir d'échantillons brouillés. Nous nous sommes glissés dans ce modèle expérimental en simulant des images traduisant des contextes du corpus. Ainsi, par exemple, un de nos fichiers de test comprenait l'équivalent de 10 235 images, chaque image comptant 7 lignes de 39 pixels pour un total de 71 645 lignes de données numérisés et 2 794 155 pixels. Chaque pixel représente la présence (ou l'absence) d'une catégorie. Chaque ligne représente les catégories associées à un mot dans une séquence consécutive de 7 mots.

Ces images correspondent à des descriptions catégorielles (catégories grammaticales hors contexte) déterminées à partir d'une typologie linguistique préalable des problèmes d'ambigüité sur les verbes. Nous utilisons cette typologie pour valider les résultats obtenus par le modèle réseau de neurones. En d'autres mots, au centre de nos images simulées, on a une

ligne dont le seul pixel allumé correspond à une catégorie verbale vérifiée. Les trois lignes avant et après cette position verbale décrivent les catégories des trois positions avant et après le verbe. Ce premier fichier a servi à l'entrainement des modèles associatifs, statistiques ou réseau de neurones. Nous avons ensuite produit un deuxième fichier de données avec une ambigüité catégorielle en position centrale. Le résultat prédit par le modèle associatif entrainé est comparé au résultat obtenu par le modèle déterministe à bases de règles.



Un protocole pour la mise au au point d'algorithmes de désambiguïsation catégorielle (4.7b, publication)

En attendant de pouvoir disposer du programme basé sur le modèle de réseau de neurones, nous avons mis au point le protocole expérimental en utilisant un modèle associatif simple, mais suffisant pour valider le fonctionnement du protocole. C'est cette chaine de traitement qui est expliquée dans Daoust et Dupuis (1996). Nous reproduisons ici, avec de légères modifications, la section de l'article qui présente ce protocole.

Nous pourrions présenter ainsi la classe de problèmes qui nous intéresse. Nous partons de l'hypothèse de la disponibilité d'un savoir lexical sous forme catégorielle. En règle générale, ce savoir est polycatégoriel, c'est-à-dire qu'un lexème peut avoir plusieurs catégories dont la pertinence va varier selon le contexte d'utilisation. Cette disponibilité est très courante en analyse de texte par ordinateur et un logiciel comme SATO est justement conçu pour rendre explicite et facilement manipulable ce savoir lexical.

Quelques exemples suffiront à l'illustrer cette situation. Lorsque l'on veut indexer des segments textuels, on dispose généralement d'un thésaurus de concepts qui renvoie à un certain nombre d'items lexicaux. Cette relation est rarement biunivoque. Un forme lexicale peut renvoyer à plusieurs termes ou à aucun terme selon le contexte d'utilisation du lexème. En analyse de discours, on retrouve une situation analogue. Prenons comme exemple la grille sociologique Bourque-Duchastel-Beauchemin (1994) [6]. Un lexème comme « école » peut renvoyer à

plusieurs catégories de la grille : Éducation, Travaux publics (l'école en tant que bâtiment), etc.

Enfin, on peut citer un problème syntaxique très connu, la catégorisation grammaticale : « ferme », par exemple, peut être un nom, un verbe, un adjectif ou un adverbe dépendant de son contexte d'utilisation. Si certains lexèmes sont polycatégoriels, nous parlerons alors d'ambigüités, plusieurs lexèmes sont monocatégoriels : par exemple, « fermait » ne peut être qu'un verbe. C'est ce problème de catégorisation grammaticale que nous allons utiliser pour construire notre protocole expérimental.

Dans beaucoup de cas, c'est l'analyse des contextes où sont employés les lexèmes qui permet de lever l'ambiguïté, c'est-à-dire de sélectionner, parmi les catégories lexicales, celle ou celles qui s'avèrent pertinentes. En d'autres mots, le savoir lexical est généralement accompagné de contraintes d'utilisation. Les contraintes qui nous intéressent ici sont dites « locales », c'est-à-dire qu'elles concernent le contexte formé par les mots qui sont dans l'entourage immédiat. On se limitera aussi aux contraintes qui sont bâties autour de systèmes catégoriels.

(...)

Nous voulons nous servir des contextes faisant appel à des lexèmes monocatégoriels pour tenter de révéler des associations entre la catégorie d'une position cible et les catégories du contexte immédiat. En d'autres termes, nous nous intéressons à cette classe de problèmes d'ambiguïtés pour lesquels l'analyse des contextes immédiats permet une levée, ne serait-ce que partielle, de l'ambiguïté catégorielle du mot en position cible. Plusieurs modèles peuvent être utilisés afin de trouver, dans les catégories portées par les divers lexèmes du contexte, un mécanisme pour confirmer ou éliminer des catégories du lexème à désambiguïser [7].

Le mécanisme le plus courant est celui de la règle, dont les règles de grammaires sont une bonne illustration. Ainsi, par exemple, on dira : si un lexème, qui ne peut

être qu'une préposition, est suivi d'un lexème ambigu entre un nom et un verbe, alors ce dernier ne peut pas être un verbe conjugué. Dans la phrase *la femelle construit son nid sous un tas de branches*, le mot *branches* ne peut pas être une forme conjuguée du verbe *brancher*.

On peut aussi concevoir des dispositifs statistiques. Dans l'exemple précédent, une chaîne de Markov dont les probabilités auraient été estimées par échantillonnage à partir d'un ensemble de contextes non-ambigus aurait aussi permis de conclure que *branches* n'est pas un verbe. De la même façon, certains types de réseaux de neurones auraient produit même effet. Les modèles statistiques, markoviens, ou les modèles à base de réseaux de neurones procèdent par apprentissage. Les probabilités, mesures d'association ou poids neuronaux, sont déterminées suite à l'analyse d'un ensemble de contextes désambiguïsants.

Généralement, l'apprentissage est réalisé sur la base de l'analyse d'un corpus échantillon dont on a levé manuellement toutes les ambiguïtés. La construction de tels corpus est une lourde tâche. De plus nous ne croyons pas qu'il soit nécessaire de lever toutes les ambiguïtés pour être en mesure de révéler des associations pertinentes. Par exemple, dans le protocole d'apprentissage utilisé par l'équipe de Robert Proulx avec le modèle EIDOS, la découverte des prototypes est possible même si l'apprentissage a été opéré à partir d'échantillons bruités.

Le protocole expérimental que nous avons élaboré vise donc à nous fournir les éléments de contrôle nécessaires pour valider une variété de modèles en jouant sur divers paramètres du modèle. Le problème qui a été choisi pour valider le protocole est celui de la catégorisation grammaticale. Il s'agit d'un problème classique pour lequel il existe aussi des solutions « classiques » à base de règles. Nous avons choisi de nous concentrer sur l'ambiguïté verbale. Ce choix tient à des considérations pratiques, à savoir le besoin de compter le nombre de propositions pour évaluer la complexité d'une phrase. Il tient aussi à des considérations théoriques sur l'importance du verbe dans l'analyse de la phrase.

Le problème peut donc être résumé ainsi. Considérant les catégories grammaticales des lexèmes qui précèdent et qui suivent un lexème pouvant être un verbe, quels modèles peut-on construire pour déterminer la catégorie effective de ce lexème ambigu.

Notre protocole expérimental peut être schématisé de la façon suivante.

1 - Constitution d'un corpus témoin.

Nous disposions déjà de corpus validés et représentatifs. Nous avons utilisé des corpus de textes fournis à des élèves de diverses classes du primaire et du secondaire. Ce corpus, élaboré dans le cadre du projet SATO-CALIBRAGE mené avec le Ministère de l'Éducation du Québec, a l'avantage de nous fournir plusieurs textes dont le niveau est gradué [8]. Il est donc possible de choisir, dans un premier temps à tout le moins, des textes considérés faciles dans lesquels on est susceptible de trouver les structures syntaxiques les plus fréquentes de la langue.

2 - Mise au point d'un dispositif classique à base de règles.

Le savoir linguistique entourant le problème choisi est suffisamment balisé pour qu'il soit possible de construire un système inspiré des grammaires locales de Silberztein (1989) [9]. De plus, nous disposons avec SATO d'un système informatique capable de mettre en oeuvre cette stratégie, de l'appliquer sur notre corpus et d'en valider la performance.

3 - Extraction du corpus témoin de contextes dont les catégories cibles ne sont pas ambiguës.

Autour du projet SATO, nous avons construit une base de données lexicales capable d'effectuer la catégorisation du lexique de notre corpus [10]. De plus, par SATO, il est très facile de repérer les contextes possédant les caractéristiques requises. Les quatre catégories cibles qui ont été retenues pour fins de test sont le verbe conjugué, le nom commun, l'adjectif et l'adverbe.

4 - Soumission des contextes au modèle associatif (phase d'apprentissage).

Nous nous proposions de tester une variété de modèles statistiques ou neuronaux. En pratique, les contraintes de temps ont fait en sorte qu'un seul modèle a pu être testé, celui de la corrélation simple (Pearson) qui se rapproche d'un réseau de Kohonen [ll].

5 - Extraction du corpus témoin des contextes dont les catégories cibles sont ambigües et pour lesquels nous disposons d'une règle de désambigüisation.

Cette étape est semblable à l'étape trois. La différence tient aux critères de sélection des contextes qui doivent posséder en position cible des lexèmes possédant plus d'une catégorie grammaticale. Aussi, nous sélectionnons des contextes dont le lexème en position cible est ambigu (polycatégoriel). De plus, nous choisissons, parmi ces contextes, ceux pour lesquels l'ambigüité peut être levée par une règle. On s'épargne ainsi un lourd travail de catégorisation manuelle et on obtient des contextes pour lesquels on dispose déjà d'un premier dispositif algorithmique éprouvé.

6 - Soumission des contextes au modèle associatif (phase de prédiction).

Lors de cette étape, l'information catégorielle des contextes doit être utilisée pour prédire la meilleure catégorie en position cible. Ainsi, dans notre expérimentation test, on calcule

$$\hat{E} = V \times C$$

où $\hat{E}$ représente le vecteur des catégories cibles estimé par le produit de la matrice de corrélation (V) avec le vecteur représentant le contexte ambigu (C). La valeur la plus forte dans le vecteur $\hat{E}$ sélectionne la variable catégorielle correspondante. Dans cette première expérimentation, nous n'avons pas construit d'intervalles de confiance susceptibles de produire une zone d'*indécidabilité*.

7 - Comparaison de la catégorie prédite par le modèle associatif avec la catégorie prédite par la règle de grammaire.

Dans la mesure où les contextes ambigus sélectionnés ont été choisis parce que l'on disposait déjà d'un dispositif de désambiguïsation, il était possible de

comparer l'efficacité du dispositif associatif par rapport à une règle linguistiquement fondée. Nous évitons ainsi de comparer nos résultats avec des décisions humaines pouvant faire appel à des motivations qui dépassent la grille catégorielle fournie au dispositif associatif. On est donc davantage en mesure de faire la distinction entre la performance du système catégoriel et la performance du dispositif associatif.

8 - Reprise de l'expérimentation en faisant varier les paramètres.

Ce protocole expérimental a été conçu pour comparer diverses méthodes et paramètres. Il est donc possible de faire varier les modèles mais aussi les données. Ainsi, on peut tester l'apprentissage en fournissant des jeux de données avec plus ou moins d'ambigüités sur les contextes.

Notes.

[6]Voir Bourque, Duchastel et Beauchemin (1994).

[7] Pour une présentation des modèles linguistiques parmi les plus connus, voir Fujisaki et al. (1989, 1991), Milne (1988) et Smith (1991).

[8] Le projet SATO-CALIBRAGE est mené en collaboration avec Léo Laroche et Lise Ouellet du ministère de l'Éducation. Le *Cahier de recherche* no. 3, publié au Centre ATO-CI, décrit ce projet de manière exhaustive.

[9] Voir Silberztein (1989).

[10]Ce dictionnaire, appelé couramment «la BDL» (base de données lexicales), a été développé au départ par Luc Dupuy dans le cadre du projet SACAO (Système d'analyse de contenu assistée par ordinateur, Programme Actions spontanées, FCAR 1989-91) dirigé par Jules Duchastel alors qu'il était directeur du Centre d'ATO.

L'originalité du protocole utilisé dans le cadre du projet AlexATO peut se résumer ainsi.

1. On exploite le fait que l'ambigüité catégorielle du lexique n'est pas systématique. On se sert donc des entrées non ambigües pour repérer dans le corpus des contextes qui ne portent pas d'ambigüité en position centrale.

2. Pour les positions à gauche et à droite de la catégorie non-ambigüe, on utilise, en phase d'apprentissage, des contextes naturellement bruités. Donc, on ne lève pas manuellement l'ambigüité catégorielle portée par l'entrée lexicale de ces positions. L'ajustement des paramètres du réseau de neurones ou du modèle probabiliste se fera en présence des données réelles.

3. On teste le résultat de l'apprentissage en fournissant des contextes réels portant aussi une ambigüité catégorielle en position centrale. On choisit des contextes ambigus pour lesquelles on dispose de règles déterministes permettant de lever l'ambigüité et d'obtenir un cadre comparatif pour évaluer la performance du modèle associatif.

4. On utilise SATO pour créer les jeux de données, pour appliquer les règles déterministes et comparer les résultats produits par les divers modèles.

Une expérimentation sur nos données avec le modèle EIDOS a finalement pu être effectuée par l'équipe Proulx. Le résultat global de l'application du modèle a été de 90%, c'est-à-dire que dans 90% des images, le modèle a donné le même résultat que la règle linguistique réputée exacte à 100%. La fin de la commandite de recherche a aussi mis fin à la poursuite de l'expérimentation. Mais les acquis de la méthode restent intacts.

De même, plusieurs idées développées au niveau informatique seront reprises dans l'implantation actuelle de SATO suite au projet ATO-MCD : modèle client-serveur, grappes de traitement et architecture à couches multiples.

4.8 Projet Visibilité : le pari d'une architecture Web

Le projet *Visibilité, analyse de texte et documentation sur Internet*, s'est déroulé de 1996 à 1998 et rassemblait un ensemble de chercheurs et d'étudiants autour du Centre ATO. Le projet avait pour but de sortir l'ATO des officines de recherche et de créer un pôle de visibilité de l'expertise en analyse de texte par ordinateur (ATO) en mettant à profit les possibilités offertes par le réseau Internet. Plus particulièrement, il s'agissait de...

1. Développer une *méthodologie intégrée* d'analyse de textes par ordinateur, à partir d'applications variées ;

2. Développer un *environnement informatique* permettant d'accéder, au moyen de l'Internet, à des corpus, fiches méthodologiques, logiciels et applications adhérant aux normes HTML et SGML ;

3. Favoriser *l'information et la formation en ATO* en rendant accessibles, dans un environnement informatisé, des outils, des méthodes et des applications susceptibles de servir au mieux les intervenants dans les industries de la langue ;

4. Accroître la collaboration interuniversitaire et internationale au niveau de la francophonie, particulièrement en France, afin de soutenir l'ATO.

À plusieurs égards, le projet Visibilité reprenait la problématique du projet SACAO avec une insistance particulière sur les besoins en formation.

> L'expérience des chercheurs du centre leur a appris qu'un des freins à la diffusion des outils en analyse de texte par ordinateur tient à l'**absence d'une formation adéquate** dans le domaine. Ainsi, la vision la plus répandue chez la plupart des utilisateurs potentiels est qu'il s'agit d'un domaine très complexe réservé à des spécialistes. Le pendant de cette vision est que plusieurs sont à la recherche du logiciel miracle, *pas compliqué*, convivial et qui va, pour ainsi dire, faire l'analyse à la place de l'analyste. C'est ainsi qu'une vision naïve et un peu magique du domaine cohabite avec l'idée de son inaccessibilité.
>
> La réalité est toute autre. D'une part, il n'y a aucune magie reliée à l'analyse de texte par ordinateur. L'informatique est là pour appuyer une démarche analytique dont le contrôle relève en dernière instance de l'analyste lui-même. Pour maîtriser cette démarche, il y a une formation de base à acquérir. (http://www.ling.uqam.ca/visib/apropos/projet.htm)

Cependant le contexte informatique était déjà très différent de celui qui prévalait à l'époque du projet SACAO, presque dix ans auparavant. C'est ainsi qu'avec *Visibilité*, nous avons amorcé le virage vers Internet, tant sur le plan de la formation que sur celui de l'implantation informatique.

> Les outils de l'Internet nous fournissent aujourd'hui des moyens nouveaux de visibilité. De même, les normes SGML et HTML nous fournissent les outils de normalisation dont on a besoin pour assurer un marquage cohérent et facilement communicable de l'information textuelle et documentaire en général. Les possibilités offertes par le marquage HTML nous permettent aussi de réaliser des passerelles vers divers logiciels en fournissant une interface à manipulation directe : boutons, menus, formulaires, etc. Le projet VISIBILITÉ comporte donc deux volets : un premier

volet, plus spécifiquement informatique, et un second volet, plus spécifiquement méthodologique. (http://www.ling.uqam.ca/visib/apropos/projet.htm)

Le volet informatique consistait à réaliser une implantation client-serveur de SATO, dans le cadre du Web et de la normalisation SGML.

Le Web constitue, on le sait, l'une des ressources les plus spectaculaires et les plus populaires du réseau Internet. C'est aussi une application du principe de marquage normalisé des documents. Donc, en plus de constituer un outil de diffusion remarquable, le Web avec son protocole de marquage HTML est lui-même un objet de recherche pour les industries de la langue. HTML a d'abord été conçu comme un format de diffusion de documents hypertextuels. Cependant, il est aussi possible d'utiliser le protocole comme outil d'interface entre un programme et l'utilisateur dans le contexte d'une architecture client-serveur. Nous avons déjà commencé à prototyper un telle interface pour le logiciel SATO. Ce premier effort nous a convaincu de la faisabilité et de la pertinence d'une telle approche.

Deux situations pourraient bénéficier d'une telle stratégie d'implantation avec des retombées certaines pour les industries de la langue. En effet, cette approche permettrait une exploitation économiquement rentable des gros corpus institutionnels: banques de lois et procédures, de jurisprudence, articles scientifiques, journaux, etc. Outre l'accès à ces données à travers les systèmes classiques de bases de données textuelles, il serait possible d'offrir au client l'accès à des analyseurs plus complexes supportés par un logiciel générique tel SATO.

La deuxième situation qui rendrait profitable une telle approche a trait à la formation et à la diffusion des produits et méthodes en analyse de texte par ordinateur. Or, pour être efficace, cette diffusion doit être encadrée tout en bénéficiant d'une visibilité maximale. Une implantation d'outils comme SATO dans le cadre du Web vise donc directement cet objectif de diffusion-formation. Il faut permettre au plus grand nombre d'acquérir cette culture en ayant accès en mode démonstration aux outils et aux méthodes du domaine. Par l'Internet, il est possible d'entrevoir un accès contrôlé à des outils d'analyse de texte, à des corpus, à des bases de données lexicales, à des fiches méthodologiques, etc. (http://www.ling.uqam.ca/visib/apropos/projet.htm)

Le volet méthodologique s'est traduit par la production de pages écrans permettant une diffusion cohérente des méthodes d'ATO dans un contexte hypertextuel. Ainsi, le site Web de Visibilité contient une base terminologique en ATO, un ensemble de fiches méthodologiques, des tutoriels, les manuels des logiciels, des démonstrations et l'accès en ligne à SATO-Internet, une bibliographie des articles des chercheurs, le plus souvent consultables en ligne et une application industrielle, *ICATeL, Indexation et Conversion SGML Automatiques pour le traitement documentaire de Textes de Loi*
(http://www.ling.uqam.ca/sato/activites/icatel/accueil.htm).

Le site Web du projet Visibilité est toujours en ligne et est fourni comme ressource en appui à l'utilisation de SATO. Certes, son contenu aurait intérêt à être mis à jour. Malgré tout, on y trouve beaucoup d'informations qui demeurent pertinentes, notamment plusieurs articles écrits par les chercheurs qui ont œuvré au sein du Centre ATO. Pour ce qui est du choix informatique de miser sur les protocoles Internet pour sortir SATO des limites de DOS sans avoir à entretenir des interfaces Windows et Mac, on doit noter qu'il s'agissait à l'époque d'une proposition avant-gardiste qui a nécessité le développement de solutions nouvelles. Aujourd'hui, les protocoles Web et les ressources pour les exploiter se sont beaucoup développées. Malgré tout, les solutions que nous avons dû inventer à l'époque ont tenu la route et ont pu évoluer jusqu'à ce jour.

4.9 Projet ATO-MCD : une infrastructure robuste pour l'ATO

Jules Duchastel, professeur de sociologie de l'UQAM et membre fondateur du Centre ATO en 2003, obtient en 2001 une chaire de recherche du Canada. C'est dans ce cadre qu'il obtient une subvention du Fonds canadien pour l'innovation (FCI) pour un projet intitulé *Base de données réseau en analyse du discours politique, infrastructure de recherche pour la Chaire de recherche du Canada Mondialisation, démocratie et nouvelles régulations politiques,* projet dont la mise en place se déroulera de 2002 à 2005. François Daoust y agit comme chef de projet pour le développement d'une infrastructure XML-Web d'analyse de texte par ordinateur.

En effet, SATO est identifié comme le logiciel central d'analyse textuelle dans le cadre du programme de recherches de la Chaire de recherche du Canada en Mondialisation, Citoyenneté et Démocratie (MCD). Dans le cadre du projet d'infrastructure associé à la création de la Chaire, la consolidation du système SATO se retrouve au centre du volet logiciel du projet et déterminera aussi la stratégie de déploiement de l'équipement informatique destiné à le supporter. Il s'agit de faire évoluer le logiciel en faisant éclater un certain nombre de limites d'utilisation tributaires de son développement dans le cadre des contraintes matérielles et logicielles très contraignantes des ordinateurs et systèmes d'exploitation des années antérieures. Cette évolution constitue un réel défi, surtout dans le contexte de l'application de la compatibilité ascendante qui caractérise de développement du logiciel. Il faut s'assurer en effet que le travail passé et en cours sur les corpus, les lexiques et les procédures, puisse se poursuivre sans rupture avec le passage aux nouvelles normes. Cette section présente les composantes de cette évolution et les solutions qui ont été trouvées pour son implantation harmonieuse.



SATO-XML **: une plateforme Internet ouverte pour l'analyse de texte assistée par ordinateur** (4.9a, publication)

Une communication sur la plateforme SATO-XML développée dans le cadre du projet d'infrastructure pour la chaire MCD a été présentée lors des JADT de 2004 (Duchastel, Daoust et Della Faille, 2004). Nous en reproduisons ici quelques extraits qui présentent l'architecture informatique et donnent des exemples visuels du fonctionnement du SATO-Web tel qu'il existe toujours au moment d'écrire ces lignes.

---

1. Introduction

Le développement d'une infrastructure de recherche au profit de la communauté des chercheurs en analyse de texte vise à rendre accessibles sur Internet des *corpus vivants*, c'est-à-dire analysables en ligne en fonction des stratégies spécifiques de chaque chercheur. Nous présentons ici une architecture développée autour du logiciel SATO (Système d'analyse de texte par ordinateur ; Daoust, 1996) mais qui permet également de rassembler divers modules d'analyse statistique, linguistique,

etc.

(...)

2. Contexte et principes méthodologiques

C'est au printemps de 2001 qu'ont débuté les travaux de développement d'une infrastructure de recherche élargie en analyse de texte par ordinateur dans le cadre de la Chaire de recherche du Canada en Mondialisation, citoyenneté et démocratie. Le projet vise à intégrer des acquis développés au cours des années mais qui restent encore trop dispersés (Duchastel, 1993 ; Duchastel et Armony, 1996).

(...)

Au niveau des données, le projet vise l'accueil, la conservation et l'exploitation scientifique de corpus de textes numérisés provenant de la communauté canadienne et internationale des chercheurs en analyse du discours. L'exportation et l'importation des données selon un format XML apparaît comme une condition pour faciliter la conservation, l'échange et le traitement des corpus et des données lexicales. XML, rappelons-le, est un langage général de balisage des documents électroniques qui permet de publier, conserver, annoter et transformer des textes selon un protocole indépendant des formats propriétaires. Faisant l'objet de concertations (The TEI Consortium, 2001), les protocoles de balisage XML facilitent le transfert des données et des résultats entre logiciels. Signalons que si le projet vise tout particulièrement les données textuelles en langue française portant sur le discours politique, la plateforme est extensible aux autres domaines de recherche en sciences sociales et en lettres.

Au niveau des traitements informatiques, l'objectif est de fournir un environnement flexible, entièrement accessible via Internet, et permettant au chercheur de déployer ses propres stratégies d'annotation, d'exploration et d'analyse de corpus collectifs ou personnels. Au cœur de la plateforme logicielle, on retrouve le logiciel SATO, augmenté de fonctionnalités permettant l'accueil et l'exportation de données en format XML.

Cette technologie permet d'envisager un véritable travail coopératif jumelant un

espace de travail personnel avec des ressources partagées : corpus, bases de données lexicales, documentation et guides méthodologiques. Il sera dès lors envisageable de transformer les collaborations fondées sur le partage de résultats en projets de recherche coopératifs durables voués à la co-exploitation de la base de données et au partage des corpus, lexiques et des savoirs socio-sémantiques. Puisque ce poste de travail électronique utilise une technologie Web standard, il est facilement modifiable et documentable par des tutoriels, manuels, bulles d'aides et guides méthodologiques. Il est également aisé d'implémenter des versions multilingues.

Au niveau matériel, le projet privilégie une approche souple faisant appel à de l'équipement standard rassemblé en îlots de traitement rassemblant plusieurs ordinateurs en réseau. La plateforme logicielle peut donc être déployée dans une variété de configurations allant de l'ordinateur personnel à un réseau élaboré d'ordinateurs se partageant les données et les traitements.

Pour comprendre les motifs à la base de cette stratégie de développement, il faut rappeler les grandes étapes d'évolution des technologies informatiques. On a connu la période des ordinateurs centraux basés, d'une part, sur un traitement centralisé et, d'autre part, sur un accès décentralisé aux données et aux traitements par le biais des terminaux accessibles par modem. Par la suite, on a assisté au triomphe de la micro-informatique qui a démocratisé l'accès au traitement des données sur le poste de travail de l'utilisateur devenu plus puissant que les ordinateurs centraux d'autrefois et à un coût qui dépasse à peine celui des terminaux de jadis. La troisième *révolution* informatique a trait à la généralisation de la réseautique via Internet et l'intégration multimédia et hypertextuelle que permet le Web et le langage HTML.

HTML est un dialecte issu de la norme SGML. Après avoir connu un développement accéléré et un peu anarchique d'HTML avec la concurrence effrénée dans le développement des navigateurs, le W3C qui arbitre le développement du Web a décidé d'arrêter l'évolution d'HTML pour promouvoir XML, un langage de balisage issu d'une simplification de SGML et qui intègre la

notion d'*extensibilité*. Ce retour à plus de rigueur dans la normalisation des formats des données a pour toile de fond l'impératif de l'échange des données sur Internet dans la perspectives de services Web permettant à des ordinateurs d'échanger des données en vue de les traiter.

Par ailleurs, l'ordinateur personnel est devenu à lui seul un véritable centre de calcul dont l'entretien dépasse souvent les capacités de l'utilisateur, en particulier en termes de mises à jour des logiciels et des chaînes de traitement. De plus, dans le domaine de la recherche, nous faisons face à des produits en évolution qui ne disposent pas toujours du même niveau de support que les logiciels commerciaux ou grand public. De là la nécessité d'aller vers des solutions mixtes qui concentrent des ressources de traitement accessibles par le Web et qui extensionnent le bureau de travail personnel du poste local vers des îlots de traitement distants. De là, aussi, la nécessité du travail coopératif, au-delà du simple échange de publications scientifiques, de telle sorte que l'on puisse échanger des données, en ce qui nous concerne les corpus de textes, les bases de données lexicales, les procédures informatiques et les méthodologies. L'accès à des traitements via le Web, et la normalisation XML des données à des fins d'échanges entre plateformes informatiques, sont donc des tendances en développement. Outre le projet ATO-MCD, citons, à titre d'exemples d'infrastructure Web dans le domaine de l'analyse des données textuelles à des fins de recherche et d'enseignement, les projets Tapor et Weblex.

Le portail Weblex de l'École normale supérieure Lettres et Sciences humaines de Lyon vise à fournir un accès par Internet à des outils d'analyse textuelle. Encore en développement, le portail permettra un accès aux outils lexicométriques développés depuis des années dans des équipes de recherche dont la tradition remonte au Centre de lexicologie politique de Saint-Cloud. Outre l'accès à des outils d'analyse quantitative des données textuelles aux fonctionnalités apparentées à celles de Lexico (Salem) et Hyperbase (Brunet), le logiciel Weblex entend fournir une édition hypertexte du document et un moteur de recherche très complet (Heiden, 2002). Le Centre ATO de l'UQAM collabore avec l'équipe de Lyon depuis plusieurs données et la convergence vers des protocoles XML devrait faciliter le

transfert des données et des traitements entre les deux groupes.

Au Canada, on retrouve un autre projet de développement d'un portail pour l'analyse textuelle. Il s'agit du Text-Analysis Portal for Research (TAPoR) : « TAPoR permettra l'établissement d'une infrastructure de chercheurs et de ressources informatiques pour l'analyse des textes à travers le pays par la mise sur pied de six centres régionaux afin de former un portail national pour l'analyse des textes » (Rockwell et coll., 2002).

Pour sa part, SATO, dans sa version HTML, est disponible depuis plusieurs années déjà en accès libre au Centre ATO de l'UQAM à l'adresse «http://www.ling.uqam.ca/ato». Tout comme la version DOS qui la précédait, SATO-HTML donne la priorité aux fonctions d'annotation et de catégorisation lexicale et contextuelle ainsi qu'aux stratégies d'analyse personnalisées (scénarios) accompagnées de mécanismes de trace de l'exploration. Comme la plupart des logiciels d'analyse textuelle, on retrouve dans SATO les fonctionnalités classiques de concordance et de fréquences lexicales mais augmentées de dispositifs de catégorisation. Au niveau des fonctions statistiques, seules les fonctions de base sont directement intégrées au logiciel. En contre-partie, le logiciel permet de produire à loisir des matrices d'occurrences destinées à être traitées par des analyseurs statistiques externes, par exemple des analyses factorielles des correspondance.

La section suivante décrit l'architecture du système et ses perspectives de développement futur.

3. Architecture de la plateforme SATO-XML

On pourrait qualifier le logiciel SATO de *tableur textuel*. Le système permet d'accueillir un corpus brut ou déjà annoté; il permet de l'annoter ou de changer l'annotation déjà présente, de catégoriser le corpus selon des grilles définies par l'analyste et une fois décrit, de l'exploiter de multiples manières. SATO permet de garder une trace complète du processus de description et d'analyse du corpus. Le logiciel offre aussi la possibilité de programmer des dispositifs de *lecture électronique* (Daoust, 2002) et, donc, d'établir des protocoles d'analyse

personnalisés et adaptés à chaque type de discours.

(...)

SATO fonctionne en mode client-serveur au moyen d'une interface HTML standard. Le logiciel est accompagné d'un environnement de gestion HTML permettant de définir des comptes d'usagers, d'ouvrir des sessions qui pourront être servies en parallèle. Le système permet de constituer des banques de textes ainsi que des librairies de scénarios et de dictionnaires. L'interface HTML est modifiable à loisir pour créer des applications particulières dans diverses langues. Cette interface permet de jumeler SATO avec d'autres logiciels, des pages HTML conventionnelles et d'utiliser toute la puissance des langages de scriptage comme Perl, PHP, Python, etc.

Les requêtes envoyées par l'utilisateur à partir de son navigateur Web, sont d'abord reçues par un programme général, une passerelle, qui gère le dialogue avec une application. Donc, la même passerelle qui dialogue avec SATO peut servir d'interface à tout autre programme qui lit un fichier de commandes et génère un fichier de résultats. Il est donc facile de rassembler autour de SATO une variété de modules informatiques qui seront déployés à la demande de l'utilisateur à partir de son navigateur Web. Ainsi, nous avons déjà mis au point une chaîne de traitement faisant appel au logiciel *Guidexpert* (Plante et coll., 2003) pour réaliser une description linguistique et sémantique de corpus. De même, nous prévoyons intégrer des logiciels statistiques et des systèmes de visualisation des résultats commandés par le chercheur dans son espace de travail privé à partir de son navigateur Web.

L'implantation du logiciel dans une architecture Web (SATO-HTML) a permis le développement d'une expertise dans le domaine des interfaces HTML et CGI (*common gateway interface)*. La nouvelle implantation SATO-XML a permis de produire une deuxième version de l'interface qui en augmente l'utilisabilité et qui supporte des interfaces multilingues. Aussi, toute la partie qui consiste à donner accès au *bureau de travail* de l'usager sur le serveur a été complétée et revue de façon à la distinguer de l'usage du logiciel SATO lui-même. D'autres

développements sont à prévoir afin d'exploiter les potentiels de filtrage et de transformation des textes en format XML.

Du point de vue interne au logiciel, la différence la plus importante entre SATO-XML et SATO-HTML sera le passage au jeu de caractères Unicode, ce qui implique des filtres de conversion permettant de récupérer les données antérieures. Aussi, l'abandon du code hérité de la version DOS sera l'occasion d'augmenter diverses limites de traitement : dimension maximale des corpus, nombre de propriétés, attributs d'affichage et d'hyperliens, etc. Du point de vue de la syntaxe externe des corpus importés et exportés, la nouveauté a trait à l'utilisation de formats XML s'ajoutant au format propriétaire défini avant l'apparition des normes XML et SGML.

On pourrait qualifier la phase actuelle de développement du logiciel de phase de consolidation permettant de passer aux nouvelles normes XML et Unicode. Ce passage se réalise dans le contexte d'une plateforme de type client-serveur basée sur une technologie Web standard facilitant le traitement coopératif entre logiciels indépendants s'échangeant des fichiers de données dans des formats standardisés. L'étape suivante consistera à ajouter un formalisme et des dispositifs de traitement permettant d'exploiter les relations structurelles tissées par le texte. Les relations les plus immédiates concernent la macrostructure de présentation du texte en sections emboîtées avec titres et renvois. Mais, elles concernent aussi les diverses constructions syntaxiques et stylistiques, les structures argumentaires, rhétoriques, dialogiques, et les divers liens marquant la cohérence textuelle. Ces dispositifs, étudiés par la linguistique textuelle (Adam, 1990), ainsi que la reconnaissance de la *macro-structure sémantique* des textes exigent des dispositifs informatiques de *catégorisation structurelle,* par analogie à la catégorisation simple que nous pratiquons actuellement. L'objectif à plus long terme est donc d'exploiter pleinement les relations entre les segments textuels dans un tracé de *lecture-explicitation* ou dans des analyses lexicales sensibles aux marques de structure.

4. Exemples d'utilisation de la plateforme

Les paragraphes qui suivent illustrent quelques moments d'une analyse réalisée à

l'aide de SATO-XML dans son état actuel de développement. Dans notre exemple, nous avons choisi les communiqués de presse en langue anglaise produits par trois groupes de défense des animaux : *World Wildlife Fund*, *Sea Shepherd* et *Greenpeace*. Ces communiqués concernent la levée du moratoire sur la pêche à la morue par l'Union Européenne (en décembre 2002) et l'annonce de la reprise de la pêche à la baleine par l'Islande (en août 2003). Comme ces communiqués ont été émis durant la même période par des groupes différents, mais s'adressant aux mêmes personnes (les membres des groupes, le public en général, les médias ainsi que les organisations mises en cause), ils permettent au chercheur de supposer les groupes assis autour d'une même table installée dans un espace délibératif à l'échelle mondiale, un espace où la production textuelle joue un rôle de premier plan.

Nous avons sélectionné un texte par groupe et par thème (baleines et poissons), soit six textes au total. Il existe pour l'utilisateur deux façons d'envoyer ses textes vers l'espace disque qui lui est réservé sur le serveur : soit à l'aide d'un formulaire disponible dans l'interface du bureau Web de SATO, soit par FTP (*File Transfer Protocol*). L'accès aux textes demeure privé, c'est-à-dire que ces derniers ne sont accessibles qu'à leur propriétaire qui pourra cependant décider de les partager en mode lecture avec d'autres membres de son groupe ou en autoriser le dépôt dans une librairie publique accessible à tous.

Les textes retenus résidant sur le serveur, nous pouvons créer un corpus à l'aide d'un formulaire HTML. Le contenu du corpus sera déterminé par une référence à chacun des six fichiers contenant les communiqués de presse. SATO en produira alors une représentation sous la forme d'un plan lexique-occurrences. Le système tiendra compte des annotations du chercheur distinguant, par exemple, les diverses parties constitutives des textes : auteurs, titres, sections, etc. L'image 1 illustre la procédure de soumission d'un corpus.

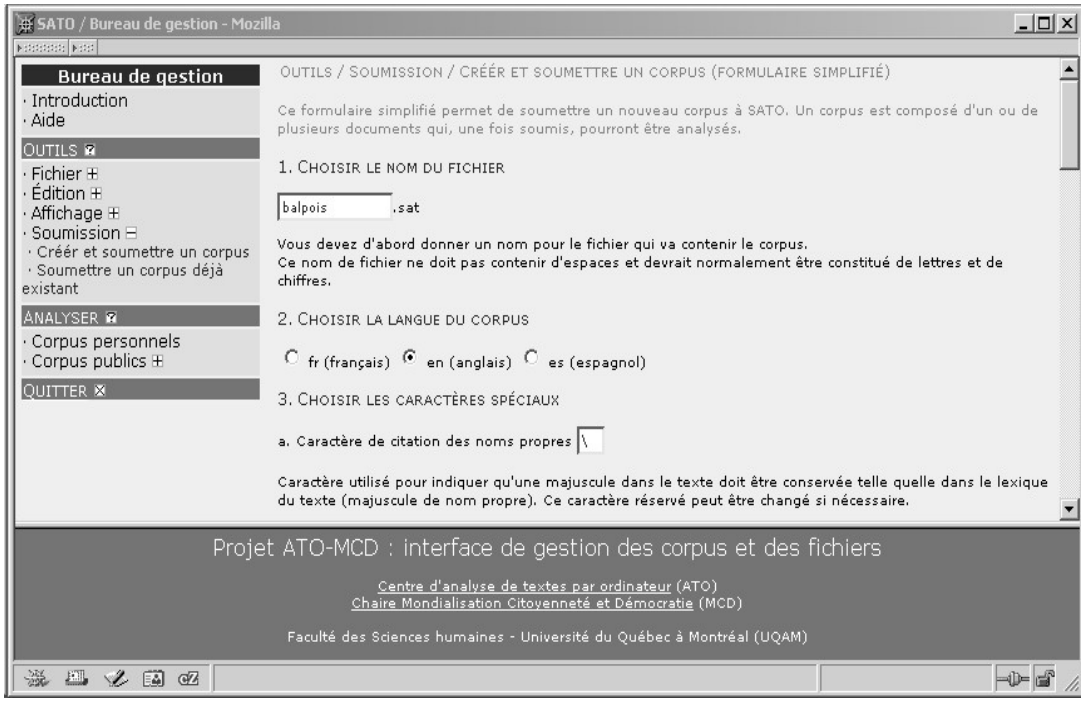


Image 1 : soumission d'un corpus

Cette photo d'écran donne un aperçu de l'interface du bureau sur le serveur. À gauche, se trouve le menu. Si on clique sur un item suivi d'un +, on développe les sous-items. Dans cet exemple, on a cliqué sur l'item *Soumission.* La section centrale de l'écran présente la partie supérieure du formulaire de soumission d'un corpus. La section du bas est la bannière d'identification. Suite à la soumission du formulaire, SATO génère le corpus et passe dans la section analyse du logiciel. Pour les sessions ultérieures, on entrera directement dans la section analyse en choisissant l'item *Corpus personnel* sur le bureau.

Une première manière d'explorer le corpus est d'afficher le lexique des formes lexicales. Dans l'illustration qui suit, nous présentons un lexique ventilé par organisme et par thème. L'image 2 présente l'interface de commande et le formulaire d'affichage du lexique. À gauche, on retrouve le menu de commandes de SATO. En cliquant sur l'item principal *lexique* suivi d'un +, on obtient le formulaire *Afficher* dans la section centrale de l'écran. Le champ *filtre* reçoit alors le patron *$*Freqtot<50>5* qui indique que seuls les items dont la fréquence totale est inférieure à 50 et supérieure à 5 seront retenus. Dans le champ *Tri*, la propriété *Fréqtot* est sélectionnée afin d'ordonner le lexique par la fréquence totale dans le corpus.

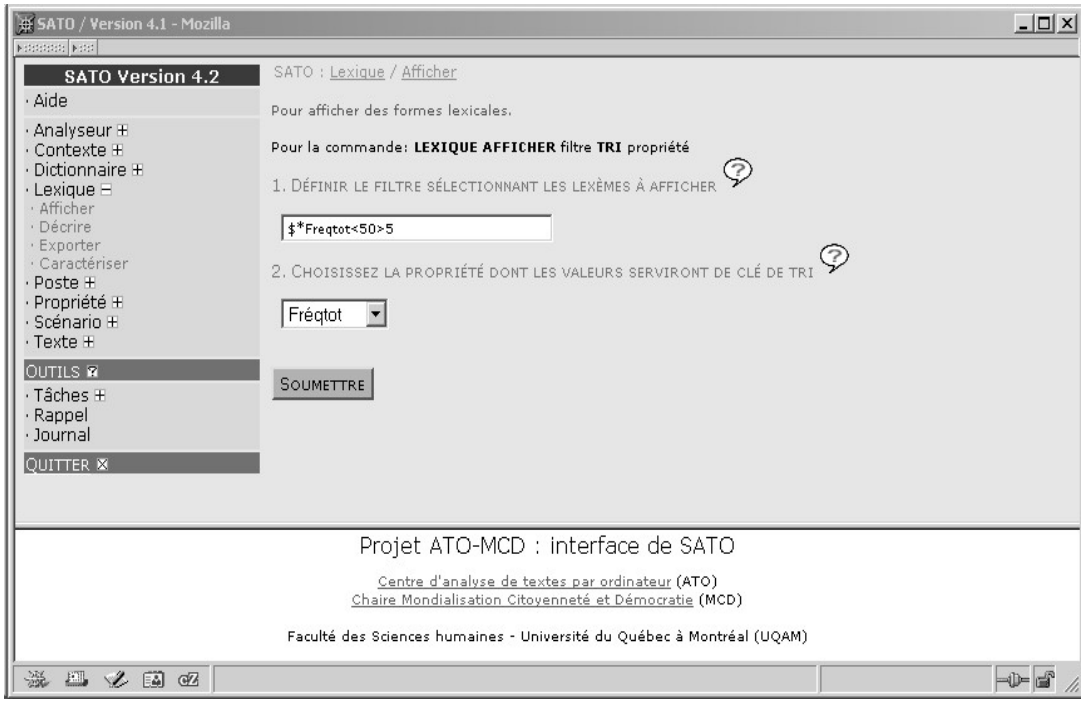


Image 2 : formulaire d'affichage du lexique

L'image 3 présente le résultat de la soumission du formulaire précédent. Outre la colonne indiquant la fréquence totale (Fréqtot), on peut voir une colonne pour chacun des groupes (WWF pour *World Wildlife Fund*, SEA pour *Sea shepherd* et GRE pour *Greenpeace*), ainsi que la distribution lexicale selon les deux thématiques concernant la pêche à la baleine (BALEINES) et la pêche à la morue (POISSONS). Dans la dernière colonne se trouve la forme lexicale. Dans la partie inférieure de la photo d'écran, on a le journal qui garde la trace de toutes les opérations effectuées durant la session de travail. De plus, la trace cumulative de chaque session est conservée dans le journal associé au corpus. On peut, par simple copier-coller de commandes reproduites dans le journal, construire des scénarios de commandes qui pourront être appliqués à loisir sur le même corpus ou tout autre corpus.

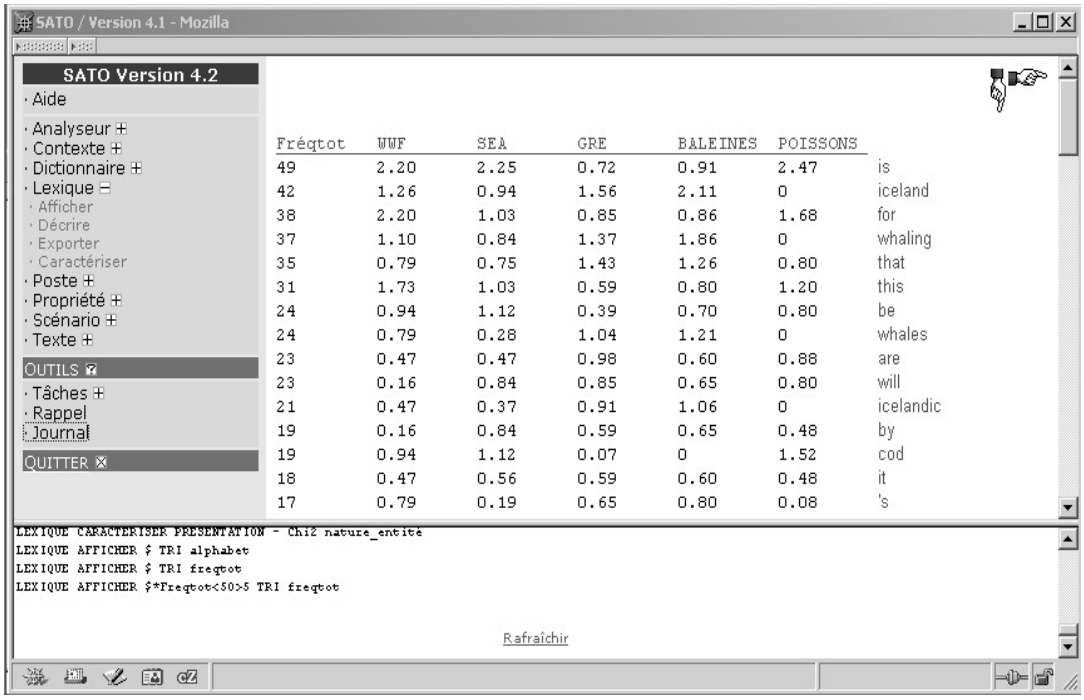


Image 3 :affichage du lexique et du journal

Si on clique sur une forme lexicale, on dévoile dans la fenêtre du bas un menu de catégorisation que nous retrouvons dans l'image 4. La partie droite de la photo d'écran révèle chacune des propriétés associées au mot retenu, ici le nom propre *Lieberman.* On y trouve notamment la propriété *nature_entité* ajoutée en cours d'analyse pour décrire la nature des acteurs sociaux : *technocrates*, *animal*, *protecteurs*, *public*, *autres en faveur des animaux*, *médias*, *scientifiques* et *pêcheurs*. La partie gauche de l'écran de catégorisation contient un menu permettant d'accéder aux contextes courts (KWIC) du mot cliqué, de le catégoriser, de sauvegarder les annotations, etc.

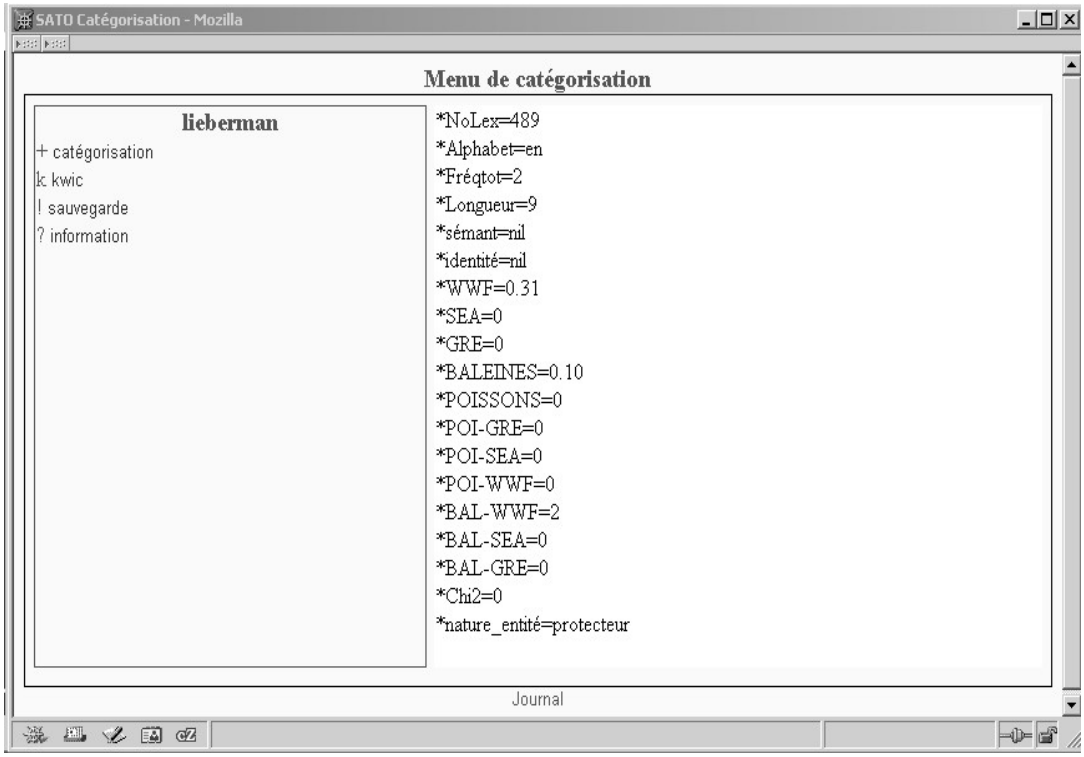


Image 4 : menu de catégorisation avec l'information

Cette catégorisation sémantique permet de visualiser la fréquence des différents types d'acteurs et leur répartition dans les divers textes. L'image 5 montre un affichage du texte avec mise en évidence des mots correspondant à des acteurs sociaux. À l'écran, les mots sont de couleur différente en fonction de chacune des valeurs de la propriété *nature_entité.* Ces valeurs décrivent les différents acteurs qui s'opposent et s'allient dans le discours des groupes de défense des animaux pour les deux thèmes choisis. Les catégories ont été établies à partir du lexique, mais elles peuvent aussi être désambiguïsées à la lecture du mot en contexte (KWIC). Par exemple, un même acteur peut être considéré, selon le contexte, comme un scientifique ou comme un protecteur.

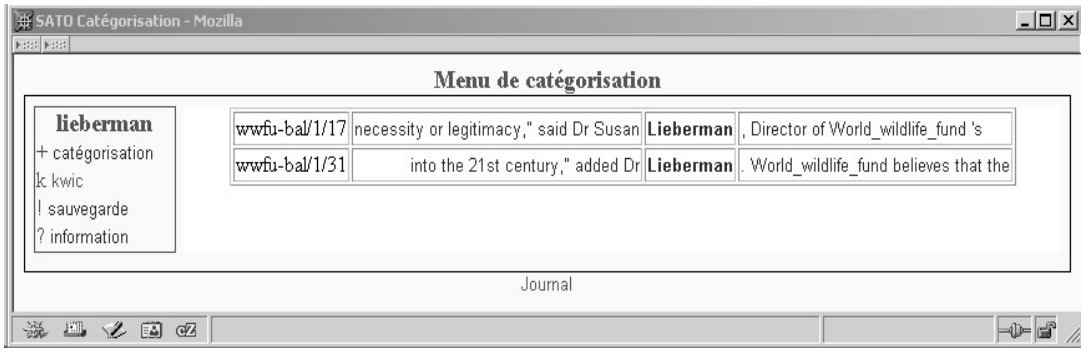


Image 5 : menu de catégorisation avec le KWIC

Un affichage du lexique des catégories d'acteurs (image 6), trié en fonction des fréquences cumulées par groupe et pondéré par la taille du texte, montre que les catégories d'acteurs qui distinguent le plus les groupes entre eux sont celles d'*autres en faveur des animaux* et de *public* (sur-représenté dans les textes de GRE), de *protecteurs* (sur-représenté dans les textes de WWF) et de *technocrates* (sous-représenté dans les textes de WWF).

Il apparaît, après l'affichage du texte catégorisé et coloré (non illustré ici) en fonction des différentes valeurs de la propriété *nature_entité*, que le discours de *Greenpeace* met en scène le plus grand nombre d'acteurs, correspondant à l'ensemble des valeurs de la propriété et répartis également dans le sous-corpus. Quant au *World Wildlife Fund*, reconnu comme le moins radical des trois groupes, il insiste dans son texte sur la morue, sur les *protecteurs* des animaux, la présence des autres acteurs n'étant que suggérée alors que le texte concernant les baleines mentionne le *public* (*community*, *people*, *consumers*). *Sea Shepherd* n'évoque, dans son texte sur les baleines, que les *pêcheurs* (*fleet*, *whalers*) confrontés à l'action directe du groupe. Le texte sur la morue est moins menaçant et moins direct et le nombre d'acteurs mentionnés s'accroît. L'opposition mise en évidence par le groupe se trouve cette fois entre le *public* et les *gouvernements*. Ces données confirment la radicalité reconnue du groupe qui n'hésite pas à saborder les baleiniers.



Image 6 : affichage du texte avec mise en couleur des mots catégorisés

Divers lexiques d'occurrences ou de co-occurrences peuvent être générés en fonction des critères de partition du corpus. Plusieurs tableaux ont été produits par des analyseurs statistiques simples appliqués sur le corpus. Les limites de cet article ne permettent pas de les reproduire ici. Il s'agissait plutôt d'illustrer quelques moments d'une analyse et de mettre en lumière les possibilités d'une plateforme Internet ouverte. On pourra, par ailleurs, consulter une démonstration en ligne sur le site Web de la chaire MCD et du Centre ATO de l'UQAM.

L'infrastructure matérielle requise pour le projet ATO-MCD consiste principalement en un ensemble de serveurs d'applications performants capables de supporter un traitement analytique sur de gros volumes de texte. Cette infrastructure doit aussi pouvoir accueillir, conserver et intégrer des nouveaux corpus à mesure que s'étendront les activités recherche de

la Chaire, rendre accessible et exploitable par Internet les outils de calculs appliqués à la base de données de discours politiques et permettre un accès élargi à cette puissance de calcul en offrant à la communauté des chercheurs de normaliser et traiter leurs propres corpus.

Afin de réduire les coûts d'infrastructure matérielle, mais aussi dans le but d'assurer une plus grande sécurité des l'infrastructure générale, nous avons opté pour une architecture en réseau permettant de répartir les données et les calculs sur des serveurs de puissance moyenne disposés en miroir dans des locaux séparés. En fait, il s'agit là du déploiement maximal d'une architecture de calcul permettant d'avoir tous les modules sur un ordinateur personnel, sur un serveur accessible depuis un réseau privé ou public, ou sur une grappe de traitement, qu'on a appelé îlot de traitement comportant un nombre quelconque d'ordinateurs distincts. Comme cette architecture fait partie de l'implantation actuelle de SATO et de son environnement de calcul, nous y reviendrons plus en détails au chapitre suivant sur l'implantation de SATO. Nous reviendrons aussi sur l'impact de la conversion vers Unicode et sur la stratégie que nous avons développée pour gérer l'interface multilingue de SATO.

L'architecture de calcul mise au point à l'occasion du projet ATO-MCD a permis de reprendre quelques idées du projet de parallélisation AlexATO pour l'implantation de notre architecture de *grappes de serveurs* ou *fermes de calcul (computer cluster).* Depuis ce temps, on dispose d'une puissance de calcul qui permet un libre accès aux outils de l'analyse textuelle de partout dans le monde. ATO-MCD aura aussi permis de parfaire les interfaces Web de SATO et d'augmenter certaines limites internes de traitement. Le projet nous a permis de préciser les protocoles permettant de fédérer des applications diverses, de partager des ressources et de gérer la langue de l'interface. Plusieurs de nos objectifs, cependant, nécessiteront, pour être réalisés correctement, de poursuivre le travail au-delà de la fermeture officielle du projet. C'est le cas,en particulier, de la normalisation XML des corpus et des résultats de l'analyse. Ce sera un des points majeurs de notre intervention au sein du réseau ATONET. La participation de François Daoust à la *Journée d'étude de l'ATALA* tenue sur le thème *Articuler les traitements sur corpus* (Daoust, 2005) marquera d'ailleurs le point de transition entre les deux projets.

4.10 Projet ATONET

Le réseau ATONET (www.atonet.net) se définit comme un *réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur.* ATONET s'est constitué formellement en 2005 grâce à la subvention du programme *Les textes, les documents visuels, le son et la technologie : Subventions de réseautage* du *Conseil de recherche en sciences humaines du Canada*. Cette subvention, dont le titulaire principal est Jules Duchastel, a fourni un cadre pour alimenter la collaboration déjà existante entre les chercheurs impliqués dans l'utilisation de l'ordinateur à des fins d'analyse textuelle, en particulier ceux qui gravitent autour des *Journées d'analyse statistique des données textuelles* (JADT) et de la revue électronique *Lexicometrica*. François Daoust a agi comme coordonnateur du réseau et principal développeur. ATONET a été subventionné jusqu'en 2008.

Liste des membres d'ATONET en 2010 (4.10a, remarque)

Initié par Jules Duchastel, titulaire de la chaire du Canada en mondialisation, citoyenneté et démocratie, le réseau s'est d'abord constitué autour des co-chercheurs Patrick Drouin, Yves Marcoux, André Salem et Jean-Marie Viprey, demandeurs d'une subvention auprès du Conseil de la recherche en sciences humaines du Canada.

Université du Québec à Montréal
- Jules Duchastel*, Chaire MCD
- François Daoust, Centre ATO

Université de Montréal
- Patrick Drouin*
- Yves Marcoux*

Université Paris-3
- André Salem*, Syled
- Serge Fleury, Syled
- William Martinez, Syled
- Maria Zimina-Poirot

Université Paris-10
- Benoît Habert, LIMSI

Université Paris-12
- Jean-Marc Leblanc, CEDITEC

Université de Franche-Comté
- Jean-Marie Viprey*, ATST

ENS-LSH, Lyon
- Serge Heiden, ICAR

CNRS
- Michel Jacobson, LACITO
- Ludovic Lebart*, Directeur de recherches
- Bénédicte Pincemin, Laboratoire de Linguistique Informatique (Université Paris 13)
- Max Reinert, Laboratoire Printemps

Université d'Oxford
- Lou Burnard*, co-éditeur des Guidelines for Electronic Text Encoding and Interchange

Université « La Sapienza » de Rome
- Sergio Bolasco*

Università della Calabria
- Michelangelo Misuraca

Université polytechnique de la Catalogne
- Monica Bécue*

Les chercheurs dont le nom est suivi d'un astérisque sont co-demandeurs de la subvention au Conseil de la recherche en sciences humaines du Canada.

(source : http://www.atonet.net/ visité en mars 2010)

ATONET répond à la nécessité de mettre en place un réseau d'échange visant à développer les conditions pour une mise en commun concrète des ressources et des méthodes en ATO, de telle sorte qu'elles puissent être utilisées par les chercheurs, les étudiants et les collaborateurs industriels sur leurs propres données. Les objectifs d'ATONET s'articulent de façon prioritaire autour de trois volets de convergence technologique : un volet *méthodes et expérimentation*, un volet *normalisation XML des formats de documents électroniques* et un volet *terminologie*. Certains de ces volets ont leur pendant au sein des groupes de travail de la revue Lexicometrica, (GADT, Groupe d'Analyse des Données Textuelles), en particulier le *GADT - Formats des données textuelles.*

Le **premier volet** d'ATONET, *méthodes et expérimentation*, concerne le partage des méthodes et des ressources en analyse de texte assistée par ordinateur. L'objectif du réseau est de se donner un cadre précis et concret pour évaluer la portée de ces méthodes et des logiciels qui les supportent en les appliquant à un même ensemble de données. C'est dans ce contexte qu'ont été menés des projets d'expérimentation contrôlée sur corpus permettant de comparer les méthodes d'analyse des membres du réseau. Ainsi, un corpus ayant déjà fait l'objet d'une analyse assistée par le logiciel SATO (Gélinas-Chebat et coll. 2004) a été soumis à une chaine

de traitement impliquant les principaux logiciels des membres du Réseau, mettant en lumière leur caractère complémentaire (Daoust et coll. 2006). Cette expérimentation montre que l'approche inductive qui vise à déployer des outils statistiques sans pré-constructions théoriques peut aussi être utilisée pour confirmer des indices lexicaux et discursifs que tentent d'appréhender l'approche hypothético déductive. Inversement, la construction d'hypothèses et de dispositifs visant à les valider permet d'aller au-delà des indices par l'établissement de grilles catégorielles qui mettent en évidence les articulations du discours. L'appareillage statistique déployé aux divers moments de l'analyse peut donc passer du statut de visualisation à celui de validation probabiliste d'hypothèses construites. Cette complémentarité méthodologique se retrouve aussi dans le déploiement successif des plans d'analyse allant de l'échelle micro (segments répétés, syntagmes et énoncés) à l'échelle macro (lexiques, et macro-structures textuelles).

Le volet *méthodes et expérimentation* prévoyait aussi le partage de corpus documentés permettant de procéder à des analyses avec divers outils et méthodes sur des données communes. Ce besoin de partage n'est pas nouveau mais n'a jamais pu être mis en place concrètement. Plusieurs raisons plus ou moins explicites ont fait en sorte que les espoirs de partage n'ont pas toujours abouti par le passé. Un des problèmes a trait à l'état du droit concernant la mise à disposition des corpus, d'où la nécessité de disposer d'un système de publication qui permette de gérer une diversité de droits de diffusion. Un autre problème est l'absence de modèle documentaire intégré permettant de gérer les divers niveaux de description d'un corpus et de donner accès à ces données par un système approprié de publication des métadonnées. C'est cette problématique qui est à l'origine du projet de dépôt de données sur lequel nous reviendrons plus loin.

Le **deuxième volet** prioritaire du réseau s'est inscrit comme une nécessité pratique pour l'expérimentation à savoir la possibilité de transférer les données d'un logiciel à l'autre et d'une méthode à l'autre sans perte des niveaux de description antérieure. Pour ce faire, il fallait convenir de formats d'échange de documents électroniques en vue de leur traitement par les divers outils logiciels développés au sein de la communauté des chercheurs en ATO. L'utilisation du langage de balisage XML s'impose naturellement pour cette tâche. C'est dans ce contexte que nous avons formulé une proposition de format d'échange basé sur les recommandations de la *Text Encoding Initiative*, (http://www.tei-c.org/) et des propositions de l'Organisation internationale de normalisation (ISO TC37/SC4 : http://tc37sc4.org/). Cette

proposition a été soumise en 2005 à un séminaire de travail au lac Sacacomie au nord de Montréal. C'est pourquoi on réfère à cette *proposition de représentation XML-TEI pour l'échange de corpus annotés*, par le vocable abrégé de *proposition Sacacomie (*Daoust et Marcoux, 2006).

Pour faire suite à l'adoption de ces propositions, nous avons développé des passerelles de conversion des données permettant de passer des *formats propriétaires* au format XML-TEI et inversement. Nous avions aussi convenu d'utiliser le principe d'un découpage en *mots* qui laisse au propriétaire du corpus la décision des modalités du découpage. Ce découpage permet de disposer d'un système référentiel pour l'annotation en place ou à distance sous forme de fichiers externes d'annotations. Ce premier pas étant franchi, nos efforts ont porté sur la définition de formats XML pour les résultats d'analyse produits par les divers modules de traitement des logiciels (segments textuels, tableaux lexicaux, graphes et données lexicographiques, etc.) avec la possibilité de rattacher les résultats aux corpus dans un espace intertextuel électronique.

Forts des acquis des deux premières années du réseau, nous avons élaboré un projet de recherche unificateur permettant de déployer les technologies émergentes autour d'un système de dépôt de données et de son référentiel. On entend ici par *référentiel* la portion de l'entrepôt de données qui concerne les métadonnées servant de références pour le traitement des données. Le système, s'il s'apparente aux *dépôts de données institutionnels* que l'on retrouve dans le monde de l'édition électronique, par exemple pour la publication des thèses universitaires, prend ici sa particularité en ce qu'il implique un réseau intertexte reliant les corpus et leurs traitements résultant en annotations, procéduriers et documents d'analyses. L'idée était de concevoir un système documentaire apte à gérer un espace articulant les corpus et les analyses avec leurs résultats, le tout sous forme de documents d'annotations et d'analyse. Cet intertexte implique non seulement le réseau dynamique des métadonnées, qui permet de relier les documents numériques (corpus, ressources, annotations et analyses), mais aussi les annotations elles-mêmes portant sur des niveaux particuliers du corpus analysé de différents points de vue et par différentes personnes. Plus particulièrement, l'objectif était de monter un prototype de dépôt de données s'appuyant sur des corpus documentés, annotés et analysés par divers outils.

En analyse de texte assistée par ordinateur, les chercheurs construisent des corpus raisonnés rassemblant des documents, ou extraits de documents, avec pour objectif de refléter, sous forme d'objets empiriques, des phénomènes discursifs à interpréter. Lorsqu'il s'agit de remettre ces pièces dans l'espace du débat scientifique, on doit à la fois rendre compte du corpus en tant qu'objet construit et des documents d'origine susceptibles d'être réutilisés par d'autres chercheurs qui voudront constituer leur propre corpus de recherche. Il s'agit là d'une difficulté réelle qui explique en partie le fait que les chercheurs hésitent à remettre leur corpus dans l'espace public. Cette absence de diffusion des corpus annotés est un frein à la recherche, car la *polémique scientifique* exige qu'on puisse revenir sur les sources, réutiliser les enrichissements analytiques (annotations, éditions critiques, documents d'analyse, etc.), les compléter, les discuter, etc.

Les questions théoriques en jeu ont trait à la construction-déconstruction des corpus en objets numériques riches pouvant rendre compte de la dimension documentaire selon divers niveaux de granularité tissés de relations multiples et diversifiées. La déconstruction consiste à décomposer les corpus, ou les collections de textes, en objets de nature plus atomique sans perte de richesse au niveau des relations qui les situent dans l'intertextualité et dans l'interdiscursivité. La (re)construction vise au contraire à rassembler ces pièces au sein de corpus raisonnés, objets de la démarche expérimentale du chercheur.

Plusieurs formalismes peuvent être utilisés pour représenter les diverses couches d'annotations sur un corpus, de même que les relations entre le corpus d'origine, les procédures de traitement, les résultats d'analyse et leurs interprétations. En particulier, on distingue l'annotation *in situ* (*embarquée*), qui s'ajoute directement au document d'origine, et l'annotation *externe* (*débarquée*) qui est dégagée du corps du texte et peut même constituer un document indépendant. XML est devenu le langage par excellence pour convenir d'une syntaxe concrète pour structurer les textes et les annotations sur les textes. XML, rappelons-le, est un langage général de balisage des documents électroniques qui permet de publier, conserver, annoter et transformer des textes selon un protocole indépendant des formats propriétaires. La conversion des données, des logiciels et des interfaces à la norme XML facilite grandement l'élaboration de chaines d'analyse textuelles réutilisables. La *Text Encoding Initiative* (*TEI* : http://www.tei-c.org/) a recours à XML pour proposer des façons de faire, des *schémas* permettant de nommer et d'organiser ces structurations. Il appartient ensuite à chaque communauté de choisir, parmi ces propositions, les formats les plus adaptés

à ses données et à l'objectif de recherche. C'est ainsi que les membres d'ATONET ont adopté une *proposition de représentation XML pour l'échange de corpus annotés.* Cette proposition, conforme aux recommandations du *TEI*, s'inspire également des travaux du comité de l'*Organisation internationale de normalisation* (*ISO*) dédié à la terminologie et autres ressources langagières (ISO TC37/SC4 : http://tc37sc4.org/). Cette première proposition de normalisation *XML-TEI* est complétée par une proposition plus élaborée basée sur le principe d'un découpage en *mots* qui permet de conserver la segmentation originale opérée par l'analyste du texte. Ce découpage permet ainsi de disposer d'un système référentiel pour l'annotation en place ou à distance au moyen de fichiers externes d'annotations.

L'adoption d'un format d'échange de corpus annotés ne suffit pas, cependant, à réaliser les conditions pour l'échange et le partage des ressources. Il faut encore documenter les conditions de production des documents et les règles de leur établissement en tant que ressources numériques. C'est là le domaine des métadonnées. À ce niveau aussi, on retrouve divers formalismes, exprimés de plus en plus en XML, et qui permettent à une communauté de décrire ses ressources. C'est le cas notamment du *Dublin Core*, de l'entête *TEI* et de *RDF*.

Le *Dublin Core*, maintenu par le Dublin Core Metadata Initiative (DCMI : http://dublincore.org/), est un ensemble restreint et standard de matadonnées conçues pour la description de documents numériques. L'ensemble contient 15 champs de base tous facultatifs et répétables (ISO Standard 15836-2003). Ce noyau de base peut être complété par des champs supplémentaires et des raffinements d'éléments existants. Il existe aussi des groupes d'intérêt qui s'appuient sur le Dublin Core pour en proposer des usages spécialisés. C'est le cas du *Open Language Archives Community* (OLAC : http://www.language-archives.org/) qui propose des raffinements supplémentaires ou des schémas précisant le format des valeurs des éléments. Par exemple, OLAC, s'inspirant des termes de relation du format bibliographique MARC, propose une liste de rôles permettant de préciser l'apport des divers contributeurs dans la constitution d'une ressource numérique. Un des avantages du Dublin Core est qu'il est directement supporté par le protocole de collecte des métadonnées qui est à la base du *Open Archive Initiative*, ce consortium qui propose un modèle permettant de fédérer les métadonnées des organismes qui consentent à les publier selon ce protocole. En s'appuyant sur ces protocoles standards, nous sommes donc certains de pouvoir faire connaître nos ressources. Il reste que nous devons, en tant que communauté spécifique, convenir d'une politique d'utilisation de ces champs qui soit adaptée à nos objectifs.

RDF (*Resource Description Framework*) est un protocole extensible de métadonnées qui émane du W3C. Il permet d'exprimer des relations qualifiant une ressource numérique identifiée par un *URI* (*Uniform Resource Identifier*). Ces relations réalisent une ontologie convenue au sein d'une communauté d'intérêt. Donc, comme pour le Dublin Core, l'adhésion à des normes de marquage des métadonnées telles RDF, s'il encadre nos pratiques, ne nous dispense pas de la nécessité de formaliser ces pratiques sous formes de politiques éditoriales. On a aussi l'entête *TEI* qui, précédant les données textuelles elles-mêmes, les décrivent et les documentent *de l'intérieur*, si on peut dire, en ce sens que l'entête TEI fait partie du document numérique.

Nos travaux récents (Daoust et coll. 2008) nous ont conduit à formuler une proposition précise et exemplifiée quant à l'articulation de ces divers formalismes. Ce modèle, suivant en cela une distinction déjà présente dans la TEI, définit comme objets numériques les textes individuels d'une part et les corpus, en tant que collection de textes, d'autre part. Chacun de ces objets devrait, à l'interne, se présenter comme un *document TEI*. Le modèle prévoit aussi que l'annotation analytique puisse prendre la forme de *documents externes d'annotation*, eux-mêmes en format TEI. Chacun des objets numériques est décrit par une fiche *Dublin Cor*e faisant appel, s'il y lieu, à des *schémas* précisant le format et la nature des métadonnées. Les liaisons entre les objets sont décrites à l'aide de *relations RDF* permettant notamment de relier les textes aux corpus et les documents d'annotation aux textes annotés. Certains résultats de traitement peuvent aussi prendre la forme de documents TEI mis en relation avec les ressources qui ont produit le résultat. Par exemple, un calcul de cooccurrences pourra prendre la forme d'un document comprenant les résultats du calcul, mais aussi les paramètres ayant servi à produire ces résultats. Les métadonnées seront aussi utilisées pour lier le résultat aux documents analysés. L'entête TEI des textes individuels et des corpus de textes permet de documenter les règles d'établissement et de codification de chaque document électronique. Cette articulation entre métadonnées externes et internes permet de respecter l'intégrité interne de chaque objet numérique, tout en favorisant des mises en relation autonomes d'objets atomiques susceptibles d'être recomposés en corpus particuliers pour des fins particulières d'analyse.

L'enjeu de cette modélisation est de constituer un réel espace public de partage des ressources pour l'ATO avec la capacité pour le chercheur d'accéder à cet espace pour se constituer des corpus de recherche et produire de nouvelles analyses qui pourront être versées à leur tour

dans cet espace public. Cela dit, au-delà du principe général de cohérence que permet le modèle, le défi technologique qu'on doit relever concerne la construction de chaines concrètes de traitement permettant de procéder, dans des termes accessibles aux chercheurs en sciences humaines, à ces constructions et déconstructions.

Le **troisième volet** d'ATONET est d'ordre terminologique. Chaque logiciel en ATO a sa terminologie spécifique désignant de façon différente des réalités qui peuvent être voisines. L'établissement d'un lexique de référence explicitant les termes du domaine est un outil très utile pour l'échange et l'apprentissage. Aussi, en particulier dans la tradition des JADT, l'échange en ATO met en contact de façon régulière des logiciels, des ressources linguistiques et des corpus dans plusieurs langues romanes (français, espagnol, italien et portugais) et en anglais. La production d'interfaces multilingues, ou la compréhension des interfaces monolingues, bénéficierait fortement d'un lexique comparatif permettant d'établir des équivalents terminologiques entre ces diverses langues. Le troisième volet du projet vise donc la mise en commun de ressources terminologiques : recension et comparaison des terminologies existantes au sein de nos équipes et mise en place d'une plateforme accessible par Internet permettant aux membres du réseau d'avoir accès à ces terminologies.

Dès la première année d'ATONET, on a pu mettre en place un modèle de fiche terminologique accessible et modifiable à partir du Web, système qui a été alimenté par la suite pour gérer une terminologie multilingue portant principalement sur les termes statistiques et les notions constitutives de l'analyse textuelle par ordinateur.

La fin des subventions pour le projet ATONET a marqué une pause dans la réalisation du projet de dépôt de données adapté à la constitution de corpus de recherche. Cependant, le travail de modélisation s'est poursuivi, notamment à l'occasion du projet d'*Encyclopédie virtuelle des révolutions.*

> Ce projet pionnier consiste à rendre disponible sur le réseau Internet les grands textes classiques en sciences sociales et humaines en matière de révolution, pour servir de matériau de base aux chercheurs en sciences humaines, enseignants, honnêtes hommes. (Ayoub 2007)

Dans ce projet, qui est encore en phase de démarrage, le modèle de données développé au sein d'ATONET occupe une place centrale du point de vue technologique. Il n'est donc pas

étonnant que l'exposé public de notre modèle de dépôt de données présenté aux JADT 2008 prenne comme exemple des données destinées à l'*Encyclopédie virtuelle des révolutions*.



Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche (4.10b, publication)

Dans une communication aux JADT 2008, nous présentions ainsi le projet de dépôt de données adapté à la constitution de corpus de recherche (Daoust et coll. 2008). Nous la reproduisons ici avec quelques corrections à la syntaxe XML des exemples.

Problématique

Introduction

Les questions théoriques en jeu ont trait à la construction-déconstruction des corpus en objets numériques riches pouvant rendre compte de la dimension documentaire selon divers niveaux de granularité tissés de relations multiples et diversifiées. La déconstruction consiste à décomposer les corpus, ou les collections de textes, en objets de nature plus atomique sans perte de richesse au niveau des relations qui les situent dans l'intertextualité et dans l'interdiscursivité. La (re)construction vise au contraire à rassembler ces pièces au sein de corpus raisonnés, objets de la démarche expérimentale du chercheur.

Plusieurs formalismes peuvent être utilisés pour représenter les diverses couches d'annotations sur un corpus, de même que les relations entre le corpus d'origine, les procédures de traitement, les résultats d'analyse et leurs interprétations. En particulier, on distingue l'annotation *in situ,* qui s'ajoute directement au document d'origine, et l'annotation *externe* qui est dégagée du corps du texte et peut même constituer un document indépendant. La notion d'organisation documentaire des corpus, et des documents qu'ils rassemblent, est donc aussi directement liée à la question de l'annotation analytique dans un contexte de débat public en constante évolution.

Un exemple complexe : les procès-verbaux du Comité d'instruction publique

Pour illustrer cette problématique, nous présentons un exemple que nous expliciterons davantage plus loin. Il s'agit d'une version électronique de l'édition nouvelle des *Procès Verbaux du Comité d'Instruction publique* des assemblées révolutionnaires en France dont les séances se sont tenues de 1791 à 1793 (Ayoub et Grenon 1997). Cette édition électronique se situe dans un projet plus vaste d'*Encyclopédie virtuelle des révolutions* dirigé par Josiane Ayoub à l'UQAM.

En format papier, l'édition Ayoub-Grenon des procès-verbaux compte 6354 pages. La particularité de cette édition, c'est qu'elle dévoile plusieurs *couches sédimentaires* de documents. En effet, les procès-verbaux avaient déjà fait l'objet d'une édition commentée par l'historien James Guillaume. Cette édition de 1889, s'étalant sur huit volumes, ajoute au procès-verbal de chacune des séances du Comité d'Instruction publique un ensemble d'annexes qui permettent d'éclairer les procès-verbaux. Des commentaires de liaison accompagnent ces annexes. On trouve aussi des centaines de notes ajoutées par Guillaume sur les procès-verbaux et les annexes. L'ouvrage contient également un index alphabétique et analytique des matières, un index des noms de lieux et de personnes. Enfin, on trouve des sommaires et des introductions.

L'édition de 1997 est aussi une édition augmentée qui ajoute un nouveau dispositif critique à l'édition de Guillaume. On y trouve donc de nouvelles introductions, de nouvelles notes et même de nouvelles annexes. Avec la mise en ligne de ces trois couches documentaires, impliquant une modélisation apte à rendre compte des multiples liens explicitement tressés entre les diverses pièces, il faut prévoir l'ajout de couches supplémentaires résultant du travail d'analyse futur des chercheurs. Il pourra s'agir de documents d'époque, mais aussi de nouveaux documents d'analyses et de multiples annotations sur les textes existants.

Si on peut, à la rigueur, considérer la collection comme un seul corpus, parce que constitué à des fins de publication sur papier, la déconstruction d'un tel *corpus* devient une nécessité pour permettre aux chercheurs de constituer leurs propres corpus raisonnés pouvant éventuellement, dans l'esprit de la recherche sur les révolutions, intégrer des pièces qui ne font pas partie de l'édition papier. Pour

illustrer quelques-uns des problèmes soulevés par cette déconstruction, examinons quelques morceaux choisis autour de la trentième séance du *Comité de l'Instruction publique* qui s'est tenue le 25 janvier 1792.

Voici d'abord un extrait du procès-verbal lui-même.

TRENTIÈME SÉANCE
Du 25 janvier 1792

M. Vaublanc a relu le projet de décret sur les pompes triomphales. Le Comité en a adopté la nouvelle rédaction[202].
M. De Bry a lu une analyse du plan de M. Talleyrand[203].
M. Para offre au Comité trois ouvrages de sa composition : des Éléments de physique, des Principes du calcul et de la géométrie, un Cours complet de physique, le tout composant sept volumes. Le Comité arrête que le président écrira à M. Para pour lui dire que le Comité reçoit son offre avec reconnaissance[204].
M. Lambert, ayant demandé à être autorisé à rendre à M. Métoyen le tableau en broderie qu'il avait présenté au Comité et qu'il redemandait, le Comité a approuvé que ce tableau fût rendu à la personne qui l'a présenté[205].

CONDORCET,
ARBOGAST, LACÉPÈDE,

Le procès-verbal est accompagné d'une annexe rassemblée par l'historien Guillaume.

PIÈCES ANNEXES

Les procès-verbaux de l'Assemblée législative contiennent les indications suivantes au sujet du projet sur les récompenses militaires :

Du jeudi 26 janvier, au matin

Un membre a demandé qu'on indiquât une séance pour entendre le rapport du Comité de l'instruction publique sur les récompenses nationales à accorder aux armées qui auront combattu pour la liberté et la constitution.
Ce rapport a été ajourné à la séance de samedi[206] au soir[207].

Du samedi 28 janvier, au matin

Un membre fait la motion que le rapport du Comité de l'instruction publique sur les récompenses à décerner aux guerriers qui auront bien mérité de la patrie et qui devait être fait dans la séance de la veille, soit entendu dans celle-ci. Cette proposition est mise aux voix et adoptée.
Le rapporteur de ce Comité présente un rapport et un projet de décret sur les récompenses à accorder aux guerriers qui auront bien servi la patrie.
L'Assemblée ajourne à vendredi la seconde lecture[208] et ordonne l'impression du rapport et du projet de décret[209].

Les renvois de notes font référence à des entrées dans le fascicule des notes. Les titres et le paragraphe introductif sont de Guillaume. Voici un extrait du fascicule des notes pour l'annexe de Guillaume à la trentième séance.

206. Il faut sans doute lire *vendredi* au lieu de *samedi,* comme on le verra par l'extrait ci-après du procès-verbal de la séance du 28 janvier (qui était un samedi).
207. Procès-verbal de l'Assemblée, t. IV. p. 301.
208. La seconde lecture n'a pas eu lieu.

209. Procès-verbal de l'Assemblée, t. IV. p. 335 — Un exemplaire imprimé du rapport de Vaublanc se trouve aux Archives nationales, AD VI, 80. (Communication de M. A. Aulard.)

Les formalismes à l'appui

Avant de proposer un modèle de dépôt de données, nous présentons un certain nombre de formalismes à la base de ce modèle. Le choix de ces formalismes découle de la nature de notre démarche, qui se rapproche de celle de la *Free Bank* (Salmon-Alt, Romary, Pierrel, 2004), bien qu'elle se situe davantage dans une perspective d'analyse de discours que dans une perspective TAL d'annotation linguistique. Dans le cas de la *Text Encoding Initiative* (TEI), son choix remonte à des travaux précédents (Daoust, Marcoux, 2006).

XML est devenu le langage par excellence pour convenir d'une syntaxe concrète pour structurer les textes et les annotations sur les textes. XML, rappelons-le, est un langage général de balisage des documents électroniques qui permet de publier, conserver, annoter et transformer des textes selon un protocole indépendant des formats propriétaires. La conversion des données, des logiciels et des interfaces à la norme XML facilite grandement l'élaboration de chaines d'analyse textuelles réutilisables. La *Text Encoding Initiative* (TEI) a recours à XML pour proposer des façons de faire, des *schémas* permettant de nommer et d'organiser ces structurations. Il appartient ensuite à chaque communauté de choisir, parmi ces propositions, les formats les plus adaptés à ses données et à ses objectifs de recherche. C'est ainsi que les membres d'ATONET ont adopté une *proposition de représentation XML pour l'échange de corpus* annotés (Daoust, Marcoux, 2006). Cette proposition, conforme aux recommandations du TEI, s'inspire également des travaux du comité de l'*Organisation internationale de normalisation* dédié à la terminologie et autres ressources langagières (ISO TC37/SC4). Cette première proposition de normalisation XML-TEI est complétée par une proposition plus élaborée basée sur le principe d'un découpage en *mots* qui permet de conserver la segmentation originale opérée par l'analyste du texte. Ce découpage permet ainsi de disposer d'un système référentiel pour l'annotation en place ou à distance au moyen de fichiers externes d'annotations. Il permet aussi de s'assurer que différents logiciels de lexicométrie puissent s'appuyer sur les mêmes unités lexicales, si

l’analyste le souhaite.

L'adoption d'un format d'échange de corpus annotés ne suffit pas, cependant, à réaliser les conditions pour l'échange et le partage des ressources. Il faut encore documenter les conditions de production des documents et les règles de leur établissement en tant que ressources numériques. C'est là le domaine des métadonnées, c'est-à-dire des données sur les données. À ce niveau aussi, on retrouve divers formalismes, exprimés de plus en plus en XML, et qui permettent à une communauté de décrire ses ressources. C'est le cas notamment du *Dublin Core*, de l'*entête TEI* et de *RDF*, que nous avons retenus. L’*Encoded Archival Description* (EAD), norme pour la représentation d’instruments de recherche de fonds d’archives historiques, aurait pu être considérée au lieu de l’entête TEI, puisqu’elle permet elle aussi la représentation de métadonnées descriptives. Cependant, comme le choix de la TEI était déjà établi pour la représentation des corpus, l’intégration des métadonnées s’avérait beaucoup simple avec l’entête TEI.

Le *Dublin Core*, maintenu par le Dublin Core Metadata Initiative (DCMI), est un ensemble restreint et standard de matadonnées conçues pour la description de documents numériques. L'ensemble contient 15 champs de base tous facultatifs et répétables (ISO Standard 15836-2003). Ce noyau de base peut être complété par des champs supplémentaires (*raffinements*). Il existe aussi des groupes d'intérêt qui s'appuient sur le Dublin Core pour en proposer des usages spécialisés. C'est le cas du *Open Language Archives Community* (OLAC) qui propose des raffinements supplémentaires ou des schémas précisant le format des valeurs des éléments. Par exemple, OLAC, s'inspirant des termes de relation du format bibliographique *MARC*, propose une liste de rôles permettant de préciser l'apport des divers contributeurs dans la constitution d'une ressource numérique. Un des gros avantages du Dublin Core, c'est qu'il est directement supporté par le protocole de collecte des métadonnées qui est à la base du *Open Archive Initiative*, ce consortium qui propose un modèle permettant de fédérer les métadonnées des organismes qui consentent à les publier selon ce protocole. En s'appuyant sur ces protocoles standards, on est donc certain de pouvoir faire connaitre nos ressources. Il reste qu'on doit, en tant que communauté spécifique, convenir d'une politique

d'utilisation de ces champs qui soit adaptée à nos objectifs.

RDF (*Resource Description Framework*) est un format extensible de métadonnées qui émane du *W3C*. Il permet d'exprimer des relations qualifiant une ressource numérique identifiée par un *URI* (*Uniform Resource Identifier*). Comme pour le Dublin Core, l'adhésion à des normes de marquage des métadonnées telles RDF, s'il encadre nos pratiques, ne nous dispense pas cependant de la nécessité de formaliser ces pratiques sous formes de politiques éditoriales.

Finalement, on a l'*entête TEI* qui, précédant les données textuelles elles-mêmes, les décrivent et les documentent *de l'intérieur*, si on peut dire, en ce sens que l'entête TEI est directement imbriquée dans le corpus considéré comme un seul document numérique. Le TEI propose aussi des éléments spécifiques (balises) qui permettent de pointer sur des documents, mais surtout des parties de documents, et de marquer les liens entre ces ressources.

Pour ce projet de dépôt de données adapté à la constitution de corpus de recherche, il faut réfléchir à la dualité nécessaire entre les formalismes de description des métadonnées (Dublin Core et RDF) d'une part, et, d'autre part, les formalismes de balisage de corpus de type TEI. La méthodologie que nous avons mise en place pour la recherche s'articule autour de deux dimensions interreliées. La première consiste à procéder à l'analyse de cas variés de corpus et de collections de textes. La deuxième dimension consiste en l'élaboration de chaines de traitement exploitant le modèle de données. Pour ce faire, nous avons identifié une plateforme logicielle ouverte qui offre toutes les possibilités de prototypage de nos modèles de données. Il s'agit de *Fedora* (*Flexible Extensible Digital Object Repository Architecture*), une plateforme Java développée dans un contexte universitaire (université Cornell et bibliothèque de l'université de Virginie). Basé sur un concept d'objets numériques pouvant être composés de plusieurs flux de données, Fedora est conçu sur le modèle du service Web fournissant divers *diffuseurs* (*disseminator*) permettant de rassembler et de mettre en forme des objets numériques locaux ou distants en réponse à une requête sur les fiches de métadonnées et sur les relations entre objets.

Ces deux dimensions méthodologiques se combinent dans une approche prototypale. D'une part, nous disposons d'un ensemble de formalismes de représentation des données et métadonnées qui doivent subir l'épreuve des traitements informatiques. D'autre part, nous avons accès à plusieurs situations réelles de corpus raisonnés et de collections de documents numériques qui traduisent des pratiques discursives et analytiques. L'enjeu est de voir jusqu'à quel point nos formalismes seront aptes à représenter nos données et aussi à tester la performance de Fedora comme logiciel de dépôt de données susceptible de supporter un espace de travail fonctionnel pour la gestion de corpus numériques à des fins d'analyse.

Un exemple complexe : les procès-verbaux du Comité d'instruction publique

Dans cette section, nous donnons un aperçu du modèle proposé, par le truchement d'un exemple. Nous reprenons l'exemple des *Procès Verbaux du Comité d'Instruction publique* pour illustrer l'utilisation des divers formalismes présentés à la section précédente.

La modélisation de ce type de collection pose d'emblée le problème du niveau de granularité qui sera retenu pour constituer des *objets numériques* accompagnés de métadonnées aussi précises que possible. L'objet physique que constitue le document papier correspond généralement à l'unité documentaire qui fait l'objet d'une entrée dans le catalogue des bibliothèques. Il est techniquement possible de produire une édition électronique qui reproduirait ce niveau de granularité. On aurait alors un document fortement structuré en termes de balises TEI afin de rendre compte de la multiplicité des unités textuelles et de leurs relations. La constitution d'un objet d'une telle complexité est une tâche difficile qui a pour inconvénient de figer un état de la collection alors que la dynamique de la recherche voudra au contraire la reconfigurer sans cesse en la spécialisant et en l'augmentant tout à la fois.

Il serait assez logique de considérer le procès-verbal d'une séance comme un objet numérique dont la fiche *Dublin Core* pourra nous donner les auteurs, la date, la référence, etc. Les notes produites par Guillaume cent ans plus tard constituent un

document distinct annotant le premier. On pourra aussi trouver des notes des auteurs de l'édition de 1997. Ces notes devraient aussi constituer un objet distinct avec une fiche *Dublin Core* distincte.

L'annexe au procès-verbal a été rassemblée par Guillaume cent ans après la tenue des séances du Comité. À ce titre, il s'agit d'un document distinct possédant sa propre fiche de métadonnées. Il en est de même des notes sur les annexes. Cependant, ces annexes sont elles-mêmes des objets composites. Ici, par exemple, on a deux extraits des procès-verbaux de l'Assemblée législative correspondant à deux sessions distinctes. Guillaume introduit et situe ces extraits dans le corps même de l'annexe, en plus d'y ajouter des renvois à des notes publiées à part dans l'édition 1997 des procès-verbaux. Dans la mesure où il s'agit ici d'extraits de procès-verbaux, dont l'intégral pourrait se retrouver dans l'édition électronique des procès-verbaux de l'Assemblée législative, on ne voit pas l'avantage de déconstruire l'annexe en unités plus atomiques. On pourra donc considérer ici un modèle composite avec une fiche de métadonnées indiquant qu'il s'agit d'extraits de procès-verbaux introduits par James Guillaume. Voici un exemple, en format libre, de fiche *Dublin Core* pour cette annexe.

Fiche *Dublin Core* de l'annexe (en format libre)

dc:title | Annexe au procès-verbal de la trentième séance des procès-verbaux du comité d'instruction publique de l'Assemblée législative : version électronique.
dc:description | Extrait du procès-verbal de l'Assemblée législative, France, 26 et 28 janvier 1792.
dc:creator | Ayoub, Josiane.
dc:contributor | Assemblée législative de France; Guillaume, James
dc:publisher | Université du Québec, Projet d'encyclopédie virtuelle des révolutions
dc:date | 2007-10-01
dc:identifier | http://corpus.ato.uqam.ca/corpus/evr/cip-legis-pv30-annexe.xml
dc:format | text/xml
dc:source | France. Assemblée nationale législative (1791-1792). Comité d'instruction publique. Procès-verbaux du Comité d'instruction publique de l'Assemblée législative, publiés et annotés par J. Guillaume. - Edition nouvelle présentée, mise à jour et augmentée par J. Ayoub et M. Grenon. Volume 2, Fascicule 1 (Séances, annexes et appendices), pp 354-355. Paris : L'Harmattan, 1997. ISBN: 2738457916.
dc:language | fr
dc:coverage | France 1792-01-26 1792-01-28
dc:rights | http://creativecommons.org/licenses/by-nc-sa/2.5/ca/
dc:relation |http://corpus.ato.uqam.ca/corpus/evr/cip-legis-pv30-notes.xml ; http://corpus.ato.uqam.ca/corpus/evr/cip-legis-pv30.xml

Les noms des divers champs de la fiche sont introduits par *dc:title, dc:description,* etc. En format XML, la fiche est plus précise en ce qu'elle permet d'indiquer des

formats qui décrivent davantage la sémantique et la syntaxe des entrées. Même si le champ *dc:relation* permet d'indiquer que des ressources supplémentaires peuvent être pertinentes, la nature de ces relations est mieux décrite par des relations RDF. Voici un exemple de définition RDF reliant l'annexe au procès-verbal.

Document *RDF* lié à l'annexe (en format XML)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:evr="http://www.evr.org/termes/"> <rdf:RDF>
 <rdf:Description rdf:about="http://corpus.ato.uqam.ca/corpus/evr/cip-legis-pv30-notes.xml">
 <annexe rdf:resource="http://corpus.ato.uqam.ca/corpus/evr/cip-legis-pv30.xml"/>
 <dc:creator rdf:resource="Ayoub, Josiane">
 </rdf:Description>
 </rdf:RDF>
```

Dans cet exemple de définition RDF, la relation a*nnexe* est définie dans l'espace de noms du projet EVR. On fait aussi appel à l'espace de nom du Dublin Core *xmlns:dc* pour montrer que les entrées du Dublin Core peuvent aussi s'exprimer dans un document RDF.

Imaginons qu'un chercheur veuille constituer un corpus regroupant le procès-verbal de la trentième session du CIP, les annexes ajoutées par James Guillaume ainsi que ses notes. La liste de ces documents devrait lui être fournie suite à une requête au moteur de recherche des métadonnées, à la manière d'une recherche dans le catalogue électronique d'une bibliothèque. Cochant les documents qu'il juge pertinents, le chercheur voudra que le système de dépôt de données assemble ces pièces sous forme d'un corpus analysable par ses outils d'analyse textuelle. Normalement, ce corpus devrait être produit en format TEI quitte à être soumis par la suite à un filtre de conversion vers le *format propriétaire* d'un logiciel particulier, comme celui développé par le réseau ATONET.

Voyons d'abord la transcription en *TEI minimal* du texte de l'annexe et du document de notes. Dans ce format, on utilise un balisage non hiérarchique faisant appel à des balises sans contenu qui servent d'éléments frontières (*milestone*) découpant le corpus en zones. Dans cet exemple, ces zones traduisent le marquage initial du document sous forme de noms de style dans le traitement de texte. Des balises vides sont aussi utilisées pour rendre compte des frontières physiques de l'édition papier en termes de page (<*pb/*>) et de ligne (<*lb/*>). Les paragraphes

sont encadrés par * * et les mots par *<w> </w>*. Ce découpage en mots, accompagné d'identifiants uniques, a pour objectif d'indiquer aux logiciels de textométrie les unités (token) qui devront servir aux comptages. Mais, ce découpage servira aussi de points d'ancrage pour référer à des parties du document par des pointeurs externes, par exemple pour accrocher le contenu d'une note au mot qui contient l'appel de la note. Comme pour tout document TEI, le contenu du texte est précédé d'un entête qui reprend des éléments de la fiche Dublin Core en plus de fournir divers renseignements sur le codage du texte. On notera que la transcription TEI de l'annexe ne fait pas mention des notes numérotées puisque celles-ci sont considérées comme une annotation externe. Les commentaires de liaison de Guillaume sont cependant marquées par le *milestone* portant l'attribut *unit="partie"* et qui permet de distinguer minimalement les parties du texte.

Annexe à la session 30 du CIP (en format TEI minimal)

<?xml version="1.0" encoding="utf-8"?> <TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt> <title>**Annexe au procès-verbal de la trentième séance des procès-verbaux du comité d'instruction publique de l'Assemblée législative : version électronique**</title>
<principal>**Ayoub, Josiane**</principal>
</titleStmt>
<publicationStmt>
<publisher>**Université du Québec, Projet d'encyclopédie virtuelle des révolutions**</publisher>
<pubPlace>**Québec, Canada**</pubPlace> <date>**2007-10-01**</date>
<availability> **http://creativecommons.org/licenses/byncsa/2.5/ca/** </availability>
</publicationStmt>
<notesStmt>
<note>**Extrait du procès verbal de l'Assemblée législative, France, 26 et 28 janvier 1792**</note>
<note>**Des notes sur le texte figurent dans un fichier séparé.**</note>
</notesStmt>
<sourceDesc> <bibl> **France. Assemblée nationale législative (1791-1792). Comité d'instruction publique. Procès-verbaux du Comité d'instruction publique de l'Assemblée législative, publiés et annotés par J. Guillaume. - Edition nouvelle présentée, mise à jour et augmentée par J. Ayoub et M. Grenon. Volume 2, Fascicule 1 (Séances, annexes et appendices), p 74. Paris : L'Harmattan, 1997. ISBN: 2738457916**</bibl> </sourceDesc>
</fileDesc>
<profileDesc> <langUsage> <language ident="fr">**Français**</language> </langUsage> </profileDesc>
<encodingDesc> <refsDecl> **Les balises «milestone n="valeur-de propriété" unit="nom-de-propriété"» concernent les mots qui suivent la balise jusqu'à l'apparition d'un nouveau milestone de même «unit». Les références de pagination utilisent les balises pb (début de page), lb(début de ligne) et w (word).** </refsDecl> </encodingDesc>
</teiHeader>
<text>
<body>
<pb n="cip-pv030a0/74"/>
<!-- *{ref:: 2-7384-5791-6, p.74} --> <lb n="1"/><milestone unit="partie" n="NoteInterne"/><w xml:id="w2">**Les**</w> <w xml:id="w3">**procès-verbaux**</w> <w xml:id="w4">**de**</w> <w xml:id="w5">**l'**</w><w xml:id="w6">**Assemblée**</w> <w xml:id="w7">**législative**</w> <w

```
xml:id="w8">contiennent</w> <w xml:id="w9">les</w> <w xml:id="w10">indications</w> <w
xml:id="w11">suivantes</w> <lb n="2"/><w xml:id="w13">au</w> <w xml:id="w14">sujet</w> <w
xml:id="w15">du</w> <w xml:id="w16">projet</w> <w xml:id="w17">sur</w> <w
xml:id="w18">les</w> <w xml:id="w19">récompenses</w> <w xml:id="w20">militaires</w> <w
xml:id="w21">:</w> </p>

<p><lb n="3"/><milestone unit="partie" n="SéanceDate"/><w xml:id="w23">Du</w> <w
xml:id="w24">jeudi</w> <w xml:id="w25">26</w> <w xml:id="w26">janvier</w><w
xml:id="w27">,</w> <w xml:id="w28">au</w> <w xml:id="w29">matin</w> </p>

<p><lb n="4"/><milestone unit="partie" n="Texte"/><w xml:id="w31">Un</w> <w
xml:id="w32">membre</w> <w xml:id="w33">a</w> <w xml:id="w34">demandé</w> <w
xml:id="w35">qu'</w><w xml:id="w36">on</w> <w xml:id="w37">indiquât</w> <w
xml:id="w38">une</w> <w xml:id="w39">séance</w> <w xml:id="w40">pour</w> <w
xml:id="w41">entendre</w> <w xml:id="w42">le</w> <w xml:id="w43">rapport</w> <w
xml:id="w44">du</w> <w xml:id="w45">Comité</w>
<lb n="5"/><w xml:id="w47">de</w> <w xml:id="w48">l'</w><w xml:id="w49">instruction</w> <w
xml:id="w50">publique</w> <w xml:id="w51">sur</w> <w xml:id="w52">les</w> <w
xml:id="w53">récompenses</w> <w xml:id="w54">nationales</w> <w xml:id="w55">à</w> <w
xml:id="w56">accorder</w> <w xml:id="w57">aux</w> <w xml:id="w58">armées</w>
<lb n="6"/><w xml:id="w60">qui</w> <w xml:id="w61">auront</w> <w xml:id="w62">combattu</w>
<w xml:id="w63">pour</w> <w xml:id="w64">la</w> <w xml:id="w65">liberté</w> <w
xml:id="w66">et</w> <w xml:id="w67">la</w> <w xml:id="w68">constitution</w><w
xml:id="w69">.</w> </p>

<p><lb n="7"/><w xml:id="w71">Ce</w> <w xml:id="w72">rapport</w> <w xml:id="w73">a</w> <w
xml:id="w74">été</w> <w xml:id="w75">ajourné</w> <w xml:id="w76">à</w> <w xml:id="w77">la</w>
<w xml:id="w78">séance</w> <w xml:id="w79">de</w> <w xml:id="w80">samedi</w> <w
xml:id="w81">au</w> <w xml:id="w82">soir</w><w xml:id="w83">.</w> </p> <!-- ... -->
  </body>
 </text>
</TEI>
```

Dans notre modèle, les notes sur l'annexe constituent un document autonome dont le lien logique avec le texte de l'annexe est inscrit dans les métadonnées sous la forme d'une relation RDF. Cependant, à l'intérieur même du document de notes, on devra retrouver des structures de pointage spécifiques permettant de lier le texte de chacune des notes avec le mot auquel il se raccroche. Mis à part ce mécanisme de pointage, le codage du document de notes suit le même modèle que l'annexe annotée. L'entête TEI contient la partie documentaire alors que le texte lui-même utilise des balises de type *milestone* pour rendre compte des frontières en parties logiques et physiques du texte. Le mécanisme de pointage se retrouve à la fin du document et consiste en éléments *<link/>* reliant, pour chacune des notes, le texte de la note, exprimé comme un empan (*range*) entre les premier et dernier mots. Ces deux liens sont encadrés par une balise *<LinkGrp>* qui donne des indications sur la sémantique d'utilisation de ces liens.

Notes de Guillaume sur l'annexe (en format TEI minimal)

<?xml version="1.0" encoding="utf-8"?> <TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt><title>**Notes sur l'annexe au procès-verbal de la trentième séance des procès-verbaux du comité d'instruction publique de l'Assemblée législative : version électronique**</title></titleStmt>
<publicationStmt> **Université du Québec, Projet d'encyclopédie virtuelle des révolutions** </publicationStmt>
<sourceDesc> <bibl>**France. Assemblée nationale législative (1791-1792). Comité d'instruction publique. Procès-verbaux du Comité d'instruction publique de l'Assemblée législative, publiés et annotés par J. Guillaume. - Edition nouvelle présentée, mise à jour et augmentée par J. Ayoub et M. Grenon. Volume 2, Fascicule 2 (Notes et index), pp 354-355. Paris : L'Harmattan, 1997. ISBN: 2738457916**</bibl>
</sourceDesc>
</fileDesc>
<encodingDesc> <refsDecl>
Les balises «milestone n="valeur-de propriété" unit="nom-de-propriété"» concernent les mots qui suivent la balise jusqu'à l'apparition d'un nouveau milestone de même «unit». Les références de pagination utilisent les balises pb (début de page), lb(début de ligne) et w (word).

milestone partie symbol "SéanceNo" "NoteNo" "NoteTxt"
</refsDecl> </encodingDesc>
</teiHeader>
<text>
<body>
<pb n="cip-pv030a0ng/354"/>
<!-- *{ref:: 2-7384-5791-6, p.354-355} --><lb n="1"/><milestone unit="partie" n="SéanceNo"/><w xml:id="w2">**30_e**</w> <w xml:id="w3">**séance**</w>

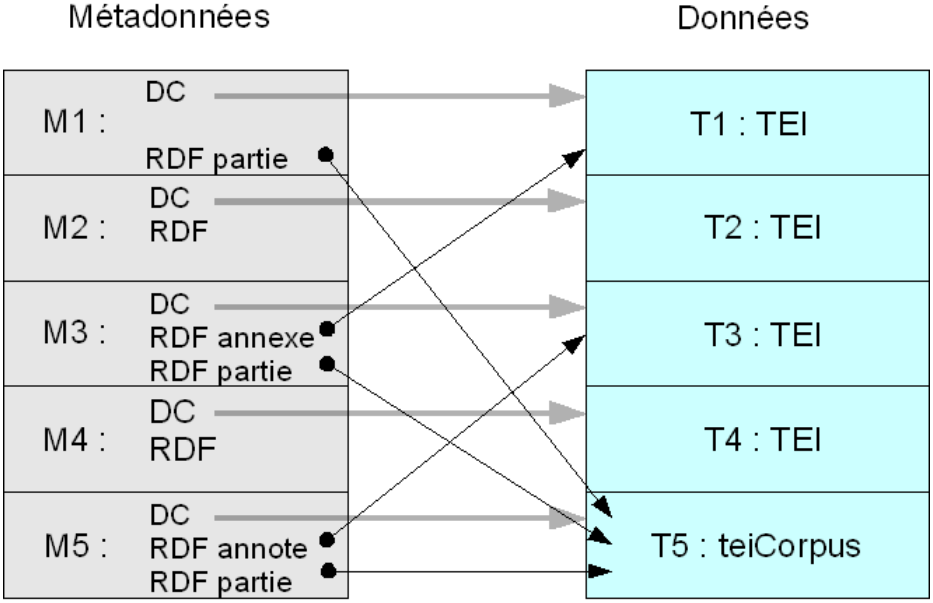
<!-- ... --> <lb n="4"/><milestone unit="partie" n="NoteNo"/><w xml:id="w43">**207**</w><w xml:id="w44">**.**</w> <milestone unit="partie" n="NoteTxt"/><w xml:id="w45">**Procès-verbal**</w> <w xml:id="w46">**de**</w> <w xml:id="w47">**l'**</w><w xml:id="w48">**Assemblée**</w><w xml:id="w49">**,**</w> <w xml:id="w50">**t**</w><w xml:id="w51">**.**</w> <w xml:id="w52">**IV**</w><w xml:id="w53">**.**</w> <w xml:id="w54">**p**</w><w xml:id="w55">**.**</w> <w xml:id="w56">**301**</w><w xml:id="w57">**.**</w>
<!-- ... -->

<linkGrp type="note-appel" targFunc="NoteTexte NoteAppel">
<!-- ... --> <link xml:id="n207" targets="range(#w43,#w57) cip-pv030a.xml#w83"/> <!-- ... -->
</linkGrp>
</body>
</text>
</TEI>

Ces deux documents pourront faire partie du corpus qui sera constitué suite à la requête du chercheur dans la base de métadonnées. La figure 1 donne une représentation schématique du rapport entre les métadonnées et les textes, textes individuels d'une part et corpus composites d'autre part. Dans ce schéma, on retrouve un bloc de métadonnées par document. Le Dublin Core (DC) renvoie à la description du contenu du document tandis que le RDF indique les relations entre documents. Une de ces relations consiste à indiquer dans quels corpus se retrouvent un document. Les documents eux-mêmes sont en format TEI. Les corpus utilisent le schéma teiCorpus qui permet de fédérer des documents TEI individuels. Un

mécanisme d'inclusion permet de faire référence au document individuel sans le dupliquer physiquement dans la base de données.

Figure 1 : Représentation des textes en objets numériques



Il existe plusieurs façons de représenter un corpus annoté en TEI. Prenons seulement le cas des notes. On peut procéder à la fusion du document contenant les notes avec le document annoté en insérant un élément *<note>* en lieu et place de l'appel de note. Il est aussi possible de conserver intégralement le document contenant les notes comme un texte autonome dans le corpus. C'est la voie que nous choisirons ici en s'appuyant sur la TEI qui définit un document de type corpus comme un *texte composite* rassemblant des textes individuels possédant leur propre entête. L'élément *<teiCorpus>* sera donc utilisé pour rassembler les éléments *<TEI>* employés pour baliser les documents individuels. Le mécanisme des pointeurs utilisé pour relier le texte des notes avec les mots annotés pourra ainsi être conservé. On notera cependant que les identificateurs d'éléments *xml:id* ont dû être normalisés pour s'assurer de leur unicité dans le corpus.

Corpus rassemblant les documents afférents à la session 30 du CIP

```
<?xml version="1.0" encoding="utf-8"?>
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader type="corpus">
  <fileDesc>
    <titleStmt>
      <title>Procès-verbal de la trentième séance des procès-verbaux du comité d'instruction publique de
l'Assemblée législative avec notes et annexes: version électronique
```

```
            </title>
           </titleStmt>
  <publicationStmt>
   <publisher>Université du Québec, Projet d'encyclopédie virtuelle des révolutions</publisher>
   <pubPlace>Québec, Canada</pubPlace> <date>2007-10-01</date>
   <availability><p>http://creativecommons.org/licenses/byncsa/2.5/ca/</p></availability>
  </publicationStmt>
  <notesStmt>
   <note>Extrait du procès verbal de l'Assemblée législative, France, 26 et 28 janvier 1792 avec l'annexe et les
notes de J. Guillaume</note>
  </notesStmt>
  <sourceDesc> <bibl> France. Assemblée nationale législative (1791-1792). Comité d'instruction publique.
Procès-verbaux du Comité d'instruction publique de l'Assemblée législative, publiés et annotés par J.
Guillaume. - Edition nouvelle présentée, mise à jour et augmentée par J. Ayoub et M. Grenon. Volume 2,
Fascicule 1 (Séances, annexes et appendices), p 74. Paris : L'Harmattan, 1997. ISBN: 2738457916</bibl>
</sourceDesc>
  </fileDesc>
  <profileDesc> <langUsage> <language ident="fr">Français</language> </langUsage> </profileDesc>
  <encodingDesc> <refsDecl> <p>Les balises «milestone n="valeur-de propriété" unit="nom-de-propriété"»
concernent les mots qui suivent la balise jusqu'à l'apparition d'un nouveau milestone de même «unit». Les
références de pagination utilisent les balises pb (début de page), lb(début de ligne) et w (word).</p>
</refsDecl> </encodingDesc>
  </teiHeader>
  <TEI>
   <teiHeader type="text">
    <fileDesc>
     <titleStmt> <title>Procès-verbal de la trentième séance des procès-verbaux du comité d'instruction
publique de l'Assemblée législative : version électronique</title>
     </titleStmt>
     <publicationStmt> <p>Idem</p> </publicationStmt>
     <sourceDesc> <p>Idem</p> </sourceDesc>
    </fileDesc>
   </teiHeader>
   <text>
    <body><p> Texte du procès-verbal </p></body>
   </text>
  </TEI>
  <TEI>
   <teiHeader type="text">
    <fileDesc>
     <titleStmt>
      <title>Annexe au procès-verbal de la trentième séance des procès-verbaux du comité d'instruction
publique de l'Assemblée législative : version électronique</title>
     </titleStmt>
     <publicationStmt> <p>Idem</p> </publicationStmt>
     <sourceDesc> <p>Idem</p> </sourceDesc>
    </fileDesc>
   </teiHeader>
<text> <body>
<p><lb n="7"/><w xml:id="cip-pv030a-w71">Ce</w> <w xml:id="cip-pv030a-w72">rapport</w> <w
xml:id="cip-pv030a-w73">a</w> <w xml:id="cip-pv030a-w74">été</w> <w xml:id="cip-pv030a-
w75">ajourné</w> <w xml:id="cip-pv030a-w76">à</w> <w xml:id="cip-pv030a-w77">la</w> <w
xml:id="cip-pv030a-w78">séance</w> <w xml:id="cip-pv030a-w79">de</w> <w xml:id="cip-pv030a-
w80">samedi</w> <w xml:id="cip-pv030a-w81">au</w> <w xml:id="cip-pv030a-w82">soir</w><w
xml:id="cip-pv030a-w83">.</w> </p>
</body>
</text>
  </TEI>
  <TEI>
```

```
   <teiHeader type="text">
    <fileDesc>
     <titleStmt><title>Notes sur l'annexe au procès-verbal de la trentième séance des procès-verbaux du comité
d'instruction publique de l'Assemblée législative : version électronique</title> </titleStmt>
     <publicationStmt> <p>Idem</p> </publicationStmt>
     <sourceDesc> <p>Idem</p> </sourceDesc>
    </fileDesc>
   </teiHeader>
   <text>
    <body>
     <!-- ... --> <p><lb n="4"/><milestone unit="partie" n="NoteNo"/><w xml:id="cip-pv030an-
w43">207</w><w xml:id="cip-pv030an-w44">.</w> <milestone unit="partie" n="NoteTxt"/><w
xml:id="cip-pv030an-w45">Procès-verbal</w> <w xml:id="cip-pv030an-w46">de</w> <w xml:id="cip-
pv030an-w47">l'</w><w xml:id="cip-pv030an-w48">Assemblée</w><w xml:id="wcip-pv030an-49">,</w>
<w xml:id="wcip-pv030an-50">t</w><w xml:id="wcip-pv030an-51">.</w> <w xml:id="cip-pv030an-
w52">IV</w><w xml:id="cip-pv030an-w53">.</w> <w xml:id="cip-pv030an-w54">p</w><w xml:id="cip-
pv030an-w55">.</w> <w xml:id="cip-pv030an-w56">301</w><w xml:id="cip-pv030an-w57">.</w>
</p><!-- ... -->
      <linkGrp type="note-appel" targFunc="NoteTexte NoteAppel">
       <link xml:id="n206" targets="range(#cip-pv030an-w7,#cip-pv030an-w41) #cip-pv030a-w80"/>
       <link xml:id="n207" targets="range(#cip-pv030an-w43,#cip-pv030an-w57) #cip-pv030a-w83"/>
        <!-- ... -->
      </linkGrp>
    </body>
   </text>
</TEI>
```

Ce modèle de données privilégie l'annotation externe non seulement pour l'apparat critique, mais aussi pour tout balisage analytique s'appuyant sur le document répertorié dans le référentiel de données. Aussi, comme pour la *FreeBank*, le découpage en mots simples (balise TEI *w*) est utilisé comme trame de base pour référer aux unités à annoter. Le chercheur qui voudra poursuivre l'analyse pourra, selon sa perspective de recherche, importer ou pas les documents d'annotation portant sur la sélection de documents sources pertinente à la constitution de son corpus de recherche.

[*Remarque. On trouvera les références bibliographiques de l'article dans la bibliographie du chapitre*]

4.11 Bibliographie du chapitre 4

ATONET. http://www.atonet.net

Adam, 1990. Adam J.-M. *Éléments de linguistique textuelle, Théorie et pratique de l'analyse textuelle*. Mardaga, Liège.

Ayoub, 2007. Ayoub J. *Encyclopédie virtuelle des révolutions.* http://corpus.ato.uqam.ca/forum/evr/ .

youb et renon, 1997. Boulad-youb J.; renon M. *Édition nouvelle, présentée, mise à jour et augmentée des procès-verbaux du comité d'instruction publique*. L'Harmattan, Paris, Montréal, ISBN 2738457916.

Bégin et Proulx, 1996. Bégin, J. & R. Proulx. Categorization in unsupervised neural networks : The EIDOS Model. IEEE Transactions on Neural Networks, 7 (1) : 147-154.

Bertrand-Gastaldy et coll., 1996. Bertrand-Gastaldy, S. ; Paquin, L.-C. ; Pagola, G. ; Daoust, F. Le traitement des textes primaires et secondaires pour la conception et le fonctionnement d'un prototype de système expert d'aide à l'analyse des jugements. In: Louisette Emirkanian et Lorne Bouchard. *Traitement automatique du français écrit*. Montréal, ACFAS: Collection Les cahiers scientifiques: 86, 1996, p. 241-276. http://www.ling.uqam.ca/sato/publications/bibliographie/Acfas_se.htm

Bertrand-Gastaldy, 1994. Gastaldy, S.; avec la collaboration de Gracia Pagola. *Le contrôle du vocabulaire et l'indexation assistés par ordinateur; une approche méthodologique pour l'utilisation de SATO*. [Montréal]: Université de Montréal. École de bibliothéconomie et des sciences de l'information; janvier 1994. 436 p. [mise à jour du rapport de 1992].

Bertrand-Gastaldy et Pagola, 1994b. Bertrand-Gastaldy, S.; Pagola, Gracia. *Le contrôle du vocabulaire et l'indexation assistés par ordinateur; un procédurier pour l'utilisation de SATO*. 100 p. [accepté pour publication par l'ASTED dans la Collection Clés en mains].

Bertrand-Gastaldy et coll., 1994c. Bertrand-Gastaldy, S.; Paquin L.-C.; Pagola, G.; Daoust, F. *Le traitement des textes primaires et secondaires pour la conception et le fonctionnement d'un prototype de système expert d'aide à l'analyse des jugements*. Colloque Traitement automatique du français écrit. 62e congrès de l'ACFAS, 16-20 mai 1994.

Bertrand-Gastaldy et coll., 1993a. Bertrand-Gastaldy, Suzanne; Daoust, François; Pagola, Gracia; Paquin, Louis-Claude. *Conception d'un prototype de système expert d'aide à l'analyse des jugements : rapport final présenté à SOQUIJ. Vol. 1 : synthèse des travaux*. [Montréal]: Université de Montréal. École de bibliothéconomie et des sciences de l'information / Université du Québec à Montréal. Centre de recherche en information et cognition ATO.CI; juillet 1993: 88 p. + annexes.

Bertrand-Gastaldy et coll., 1993b. Bertrand-Gastaldy, Suzanne; Daoust, François; Pagola, Gracia; Paquin, Louis-Claude, 1993. *Conception d'un prototype de système expert d'aide à l'analyse des jugements : rapport final présenté à SOQUIJ. Vol. 2 : documentation du système expert*. [Montréal]: Université de Montréal. École de bibliothéconomie et des sciences de l'information / Université du Québec à Montréal. Centre de recherche en information et cognition ATO.CI; juillet 1993: 91 p.

Bertrand-Gastaldy et coll., 1993c. Bertrand-Gastaldy, S.; Daoust, F.; Meunier, J.-G.; Pagola, G.; Paquin, L.-C. Prototype de système expert pour l'aide à l'analyse (tri, classification, indexation) des documents de jurisprudence. ICO93; *Actes du Colloque international en informatique cognitive des organisations/ International Conference on Cognitive and Computer Sciences for Organizations*, 4-7 mai 1993, Montréal: 503-507.

Bertrand-Gastaldy et coll., 1993d. Bertrand-Gastaldy, S.; Daoust, F.; Meunier, J.-G.; Pagola, G.; Paquin, L.-C., 1993. *Les traitements statistico-linguistiques et l'enquête cognitive comme moyens de reconstituer l'expertise des spécialistes en analyse documentaire: le cas de la jurisprudence*. Montréal: Université du Québec à Montréal, Centre de recherche en Cognition et Information ATO.CI. 30 p. (Cahiers de recherche; 1)

Bertrand-Gastaldy et coll., 1993e. Bertrand-Gasltady, S.; Daoust, F.; Meunier, J.-G.; Pagola, G; Paquin, L.-C., 1992. *Un prototype de système expert pour l'aide à l'analyse des jugements*. Congrès international Informatique et droit / Computers and Law, Montréal 1992, 30 septembre-3 octobre 1992.

Bertrand-Gastaldy et Pagola, 1992. Bertrand-Gasltady, S.; Pagola, G. L'analyse du contenu textuel en vue de la construction de thésaurus et de l'indexation assistées par ordinateur; applications possibles avec SATO (système d'analyse de textes par ordinateur). *Documentation et bibliothèques*; 38(2); avril-juin 1992: 75-89.

Bertrand-Gastaldy, 1992. Bertrand-Gasltady, S.; avec la collaboration de Gracia Pagola. *Le contrôle du vocabulaire et l'indexation assistés par ordinateur; une approche méthodologique pour l'utilisation de SATO*. [Montréal]: Université de Montréal. École de bibliothéconomie et des sciences de l'information; janvier 1992. 612 p. en pagination variée.

Brill, 1992. BRILL E. A simple rule-based part of speech tagger. In Proceedings of the Third Conference on Applied Computational Language (ACL) Processing, Trento.

Bourque, Duchastel et Beauchemin, 1994. Bourque, G.; Duchastel, J.; Beauchemin, J. *La société libérale duplessiste,* Montréal, Les Presses de l'Université de Montréal.

Brunet, 2004. Brunet, É. *Logiciel HYPERBASE (version 2.3)*, http://ancilla.unice.fr/~brunet/pub/hyperbase.html site visité le 7 janvier 2004.

Burnard et Bauman, 2007. Burnard L.; Bauman S. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. http://www.tei-c.org/release/doc/tei-p5-doc/html/

Coulon et Kayser, 1986. Coulon, Daniel; Kayser, Daniel. Informatique et langage naturel : Présentation générale des méthodes d'interprétation des textes écrits. *Technique et Science Informatiques*, Février, 1986, pp. 103-126.

Courtois, 1990. Courtois B. Un système de dictionnaires électroniques pour les mots simples du français, In: *Dictionnaires électroniques du français*, Éditions Blandine Courtois et Max Silberztein, Langue Française, n°87, Paris : Larousse, pp. 11-22.

Cucumel, 1993. Cucumel, G. Classification par partition et classification hiérarchique: deux méthodes complémentaires. *Cahier de recherche*, n° 3, Centre ATO-CI, UQAM, p. 83-96.

Daoust et coll., 2008. Daoust F.; Duchastel J.; Marcoux Y.; Rizkallah E. (2008). Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche, in *Actes des JADT-2008, vol. 1*, pp- 355-367, Presses universitaires de Lyon, 2008. ISBN 978-2-7297-0810-8 http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2008/pdf/daoust-duchastel-marcoux-rizkallah.pdf

Daoust et coll., 2006. Daoust F.; Dobrowolski, G.; Dufresne, M.; Gélinas-Chebat, C. Analyse exploratoire d'entrevues de groupe : quand ALCESTE, DTM, LEXICO et SATO se donnent

la main, in *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp- 313-326, Presses universitaires de Franche-Comté, 2006. ISBN 2.84867130.0
http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/028.pdf

Daoust et Marcoux, 2006. Daoust, F. et Marcoux, Y. Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés. In *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp- 327-340, Presses universitaires de Franche-Comté, 2006. ISBN 2.84867130.0
http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/029.pdf

Daoust et Gélinas-Chebat, 2005. Daoust, F. ; Gélinas-Chebat, C. *Lisibilité de documents de déclaration de revenus 2004 :analyse visuelle et textuelle.* Avis professionnel. Rapport privé commandité par le Bureau du Vérificateur général du Gouvernement du Québec.

Daoust, 2005. *Projet ATO-MCD : Une implantation des technologies Web pour le partage des corpus et des traitements.* Journée d'étude de l'ATALA, Paris, 12 février 2005.

Daoust, 2002. Daoust F. L'analyse de texte assistée par ordinateur, lunette de lecture des textes électroniques. *Communication présentée au colloque Publications et lectures numériques : problématiques et enjeux, 70ième congrès de l'ACFAS*, EBSI, Montréal. http://www.ebsi.umontreal.ca/rech/acfas2002/daoust.pdf

Daoust et Gélinas-Chebat, 2001. Daoust, F. ; Gélinas-Chebat, C. *Avis professionnel sur la lisibilité des documents sur l'impôt des particuliers*. Avis professionnel. Rapport privé commandité par le Bureau du Vérificateur général du Gouvernement du Québec et qui a servi à appuyer la recommandation 8.87 (tome 2 chap 8) du *Rapport du Vérificateur général du Québec*, Guy Breton, rapport déposé à l'Assemblée nationale pour l'année 2000-2001.

Daoust, 1996. Daoust, F. *SATO (version 4.0), Manuel de référence,* Centre d'analyse de texte par ordinateur (ATO), 1996.

Daoust, Laroche et Ouellet, 1996. Daoust, F.; Laroche, L.; Ouellet, L. SATO-CALIBRAGE : Présentation d'un outil d'assistance au choix et à la rédaction de textes pour l'enseignement,1996. *Revue québécoise de linguistique* , vol 25, no 1.

Daoust, Laroche et Ouellet, 1996b. Daoust, F.; Laroche, L.; Ouellet, L. *SATO-CALIBRAGE : Guide de l'usager,version 1.0,* 1996. Centre ATO, UQAM.

Daoust et Dupuis, 1996. Daoust, F.; Dupuis F. Analyse de texte et parallélisme, un protocole pour la mise au point d'algorithmes de désambigüisation catégorielle. In L. Émirkanian & L.H. Bouchard (réd.), *Traitement automatique du français*, Les Cahiers scientifiques de l'ACFAS, n° 86, p. 153-173. (Actes du colloque sur le traitement automatique du français écrit : développements théoriques et applications, 1996)

Daoust et Dupuis, 1994. Daoust, F.; Dupuis, F. Le dépistage en contexte des verbes conjugués à l'aide du logiciel SATO. *Revue ICO Québec*, vol 6, n° 1-2, p.106-113.

Daoust, 1993. Daoust, F. La méthode expérimentale en analyse de texte par ordinateur, 1993, In *Actes du colloque Les sciences du texte juridique: Le droit saisi par l'ordinateur*, sous la dir. de C. Thomasset, R. Côté et D. Bourcier. Textes présentés à un séminaire tenu à Val-Morin, Québec, du 5 au 7 octobre 1992 sous l'égide du Laboratoire Informatique, droit et linguistique du CNRS (France) et du Groupe de recherche Informatique et droit de

l'Université du Québec à Montréal. Cowansville: Les Éditions Yvon Blais Inc., 1993: 441-448.

Daoust, Laroche, Ouellet & coll. 1993. Daoust, F. ; Laroche, L. ; Ouellet, L. & coll. Le projet SATO-CALIBRAGE. *Cahier de recherche*, n° 3, Centre ATO-CI, UQAM.

Daoust, 1993. Daoust, F. Le dispositif mathématique. *Cahier de recherche*, n° 3, Centre ATO-CI, UQAM, p. 75-96.

Daoust, 1992a. Daoust, F. SATO (version 3.6), Manuel de référence, Centre d'analyse de texte par ordinateur (ATO), 1992.

Daoust, 1992b. Daoust, F. Le projet d'atelier cognitif et textuel: un moteur d'inférences original, *Actes de la Deuxième Conférence régionale de l'ACM SIGUCCS,* 1992, Université Laval, Québec, pp. 71-77.

Daoust, 1990. Daoust, F. L'informaticien, le lecteur et le texte, l'approche SATO. Gestion de l'information textuelle, revue ICO, 1990, pp. 55-60.

Daoust, 1987. Daoust, F. *SATO : Système d'Analyse de Textes par Ordinateur (version 3.4). Manuel de référence pour les micro-ordinateurs PC et PC compatibles*, Université du Québec à Montréal, Centre d'Analyse de Textes par Ordinateur, 1987, 81 pages.

Daoust, 1986. Daoust, F. *SATO (version 3.2), Manuel de référence,* Centre d'analyse de texte par ordinateur (ATO), Montréal, 1986.

Daoust, 1984. Daoust, F. *SATO (version 3.0), Guide d'utilisation (version préliminaire),* Service de l'informatique, Montréal, 1984.

Daoust, 1979. Daoust, F. *SATO,* Présentation du système SATO (version 3). *Colloque d'enseignement recherche de l'Université du Québec.* Mai 1979.

Daoust, 1976. Daoust, F. *SATO, Projet de consolidation informatique,* Service de l'informatique, UQAM, Montréal, décembre 1976.

Delisle et Vézina, 1997. Delisle, Cynthia; Vézina, Marie-Hélène, 1997. *ICATeL: Indexation et Conversion SGML Automatiques pour le traitement documentaire de Textes de Loi***.** http://www.ling.uqam.ca/sato/activites/icatel/accueil.htm

Dublin Core Metadata Initiative. Site Web :http://dublincore.org/

Duchastel, Daoust et Della Faille, 2004. Duchastel, J. ; Daoust, F. Della Faille, D. *S*ATO-XML: une plateforme Internet ouverte pour l'analyse de texte assistée par ordinateur. In *Le poids des mots*, Actes des JADT-2004, vol. 1, pp- 353-363, Presses universitaires de Louvain, 2004.

Duchastel, 2001. Duchastel J. *Présentation du projet ATO-MCD.* http://ato.chaire-mcd.ca/presentation/

Duchastel, Armony, 1996. Duchastel J. et Armony V. Textual Analysis in Canada: An Interdisciplinary Approach to Qualitative Data. *Current Sociology*, vol. 44, no 3, 259-278.

Duchastel, 1993. Duchastel J. Discours et informatique: des objets sociologiques? *Sociologie et sociétés*, vol. 25, no 2, 157-170.

Duchastel et Armony, 1991. Duchastel, J.; Armony, V. Étude d'un corpus de dossiers de la cour juvénile de Winnipeg à l'aide du système d'analyse de textes par ordinateur (SATO). In *Actes du colloque 'Journées internationales d'analyse statistique de données textuelles'*. Barcelone: Universitat Politècnica de Catalunya, 1991: 89-108. http://www.ling.uqam.ca/sato/publications/bibliographie/Jul13.htm

Duchastel et coll., 1989a. Beauchemin, J.; Daoust, F. Duchastel, J.; Dupuy, L.; Paquin, L.-C. The SACAO Project: using computation toward textual data analysis in the social science, *Advances in Computing and the Humanities*, JAI Press, Greenwich, Connecticut, vol. 3-4, 1989.

Duchastel et coll., 1989b. Duchastel, J.; Dupuy, L.; Paquin, L.-C.; Beauchemin. J.; Daoust, F. Système d'analyse de contenu assistée par ordinateur (SACAO). *Actes du colloque La description des langues naturelles en vue d'applications linguistiques*. Centre international de recherche sur le bilinguisme (CIRB), Québec, Université Laval, 1989, pp. 197-210. http://www.ling.uqam.ca/sato/publications/bibliographie/Jul15.htm

Encoded Archival Description. Version 2002 Official Site : http://www.loc.gov/ead/

Fedora. Site Web : http://www.fedora.info/ .

GADT. Groupe d'Analyse des Données Textuelles au sein de la revue Lexicometrica.
GADT - Ressources Textométriques. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/ressources-textometriques/
GADT - *Textométrie Multilingue* http://www.cavi.univ-paris3.fr/lexicometrica/jadt/textometrie-multilingue/
GADT - *Formats des données textuelles http://www.cavi.univ-paris3.fr/lexicometrica/jadt/format-texte/*

Gélinas-Chebat et coll., 2004). Gélinas-Chebat, C.; Daoust, F.; Dufresne, M.; Gallopel, K. et Lebel, M.-H. *Analyse exploratoire d'entrevues de groupe : les jeunes Français et le tabac.* In Purnelle, G., Fairon, C. et Dister, A. editeurs, I, *Actes des JADT-2004*, pages, 479-487.

Gélinas-Chebat, et coll., 1993. Gélinas-Chebat, C.; Préfontaine, C. ; Lecavalier, J. ; Chebat, J.C. Lisibilité - Intelligibilité de documents d'information. *Cahier de recherche*, n° 3, Centre ATO-CI, UQAM, p. 19-35.

Habert, 1995. Habert, B. Traitements probabilistes et corpus. *T.A.L.*, revue semestrielle de l'ATALA, vol. 36, n° 1-2.

Hamilton-Smith, 1970. Hamilton-Smith, N. Concord, User' Specification. Edinburgh Regional Computing Center, 1970.

Heiden, 2002. Heiden, S. *Weblex, Manuel Utilisateur, version 4.1 intermédiaire,* http://lexico.ens-lsh.fr/doc/weblex.pdf https://weblex.ens-lsh.fr/ sites visités le 7 janvier 2004.

ISO TC37/SC4. Site Web : http://tc37sc4.org/ .

JADT. http://www.jadt.org/

Laroche, 1993. Laroche, L. Analyses statistiques pour la constitution d'un indice SATO-CALIBRAGE. *Cahier de recherche*, n° 3. Centre ATO-CI, UQAM, p. 97-139.

Laroche, 1990. Laroche, L. Calibrage des textes et lisibilité. *Revue ICO Québec*, vol. 2, n° 3, p. 114-117.

Lexicometrica. http://www.cavi.univ-paris3.fr/lexicometrica/

MARC Standards. Site Web : http://www.loc.gov/marc/ .

McCarthy, et coll. 1962. McCarthy, John, *LISP 1.5 Programmer's Manual,* (with Abrahams, Edwards, Hart, and Levin), MIT Press, Cambridge, Mass.

Maingueneau, 1997. Maingueneau D. *L'Analyse du Discours, Nouvelle édition*. Hachette, Paris.

Meunier, Paquette et Daoust, 1975. Meunier, J.-G.; Paquette, M.; Daoust, F. *SATO (version 2), Manuel de l'usager*. Recherches et Théories no. 20.2, Département de philosophie, UQAM 1975.

Meunier, Rolland et Daoust, 1976. Meunier, J.-G.; Rolland, S.; Daoust, F. A system for Text and Content Analysis. *Computers and the Humanities,* Vol. 10, pp. 281-286, Pergamon Press, 1976.

OLAC. Site Web : http://www.language-archives.org/ .

Open Archive Initiative. Site Web : www.openarchives.org **.**

Ouellette, 1972. Ouellette, F. *Jeudemo, Système de traitement de texte*. Centre de calcul, Université de Montréal, version préliminaire, octobre 1972.

Paquin, 1992. Paquin, L.-C. La lecture experte. *Technologies, Idéologies, Pratiques: Sciences sociales et intelligence artificielle*, 10 (2-4), 1992: 209-222.

Paquin, Daoust et Dupuy, 1990. Paquin, L.-C.; Daoust, F.; Dupuy, L. ACTE: L'ingénierie textuelle et cognitive pour l'indexation hypertextuelle**.** In *Actes du 'Colloque sur les instruments de communication évolués, hypertextes, hypermédias'* (Paris, 16 mai 1990)**.** Paris: Le Journal de la Formation Continue et de l'EAO, 1990: 83-99, http://www.ling.uqam.ca/sato/publications/bibliographie/Hypertex.htm **.**

Paquin, Daoust et Dupuy, 1989. Paquin, L.-C.; Daoust, F.; Dupuy, L. ACTE: a workbench for knowledge engineering and textual data analysis in the social sciences. In *Proceedings of the Fourth International Conference on Symbolic and Logical Computing (ICEBOL4).* Madison: Dakota State University Press, 1989: 122-136.

Paquin, 1987. Paquin, L.-C. Déredec-EXPERT (Version 2.0). Université du Québec à Montréal, Centre d'Analyse de Textes par Ordinateur, 1987, 119 pages.

Pêcheux, 1994. Pêcheux M. Sur les contextes épistémologiques de l'analyse du discours. *Mots*, no. 9, oct 1984, Presses de la Fondation nationale des sciences politiques.

Plante et coll., 2003. Guidexpert ATO. http://fable.ato.uqam.ca/guidexpert/guidexpert-ato-wp.htm

Poirier, 1985. Poirier, D. *Pour des résumés adéquats de jurisprudence québécoise et canadienne: une étude du document jurisprudentiel, de sa structure, de ses citations, de son*

rôle et de sa spécificité. Montréal: Université de Montréal, École de bibliothéconomie et des sciences de l'information; 1985.

Prévost, Heiden, Dupuis, 2000. Prévost, S.; Heiden, S. ; Dupuis, F. « Catégorisation d'un corpus hétérogène de français médiéval », in *Actes du colloque JADT 2000* : 5es Journées Internationales d'Analyse Statistique des Données Textuelles' Lausanne, 2000, Ecole Polytechnique de Lausanne, vol 2, p. 485-492. http://halshs.archives-ouvertes.fr/docs/00/08/77/70/PDF/prevost-biblio8.pdf

Rolland et Daoust, 1976a. Rolland, S. ; Daoust, F. Documentation informatique : Les fichiers SATO in Meunier, J.-G., Système d'analyse des textes par ordinateur (SATO), rapport technique. Recherches et Théories no. 14. Département de philosophie, UQAM 1976.

Rolland et Daoust, 1976b. Rolland, S. ; Daoust, F. Documentation informatique : Description des programmes in Meunier, J.-G., Système d'analyse des textes par ordinateur (SATO), rapport technique. *Recherches et Théories* no. 14. Département de philosophie, UQAM 1976.

Rockwell et coll., 2002. Rockwell et coll. *TAPoR: Text-Analysis POrtal for Research.* http://huco.ualberta.ca/Tapor/ site visité le 7 janvier 2004.

Russell, 1967. Russell, D. B. *COCOA - A Word Count and Concordance Generator for Atlas*. Chilton: Atlas Computer Laboratory.

Salem et coll., 2004. Salem, A.; Lamalle, C.; Martinez, W.; Fleury, S.*Lexico 3*. http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/lexicoWWW/ site visité le 7 janvier 2004.

Salmon-Alt et coll. 2004. Salmon-Alt, S.; Romary, L.; Pierrel, J.-M. Un modèle générique d'organisation de corpus en ligne : application à la FreeBank, *Traitement Automatique des Langues*, Vol.45, n°3, pp. 145-169, 2004. ISSN 1248-9433

Silberztein, 1989. Silberztein, M. *Dictionnaire électronique et reconnaissance lexicale*, thèse de doctorat en informatique, LADL, Université Paris 7.

The TEI Consortium, 2001. *Text Encoding Initiative, The XML Version of the TEI Guidelines*. Edited by Sperberg-McQueen C.M. and Burnard L.

W3C, 2000. *Harvesting RDF Statements from XLinks*. Note http://www.w3.org/TR/xlink/rdf . Editors: Ron Daniel Jr. (Metacode Technologies Inc.).

W3C, 2001. *XML Linking Language (XLink) Version 1.0*. Recommandation http://www.w3.org/TR/xlink/ . Editors: Steve DeRose, Brown University Scholarly Technology Group, Eve Maler, Sun Microsystems, David Orchard, Jamcracker.

5 Le modèle d'implantation de SATO

5.1 Choix stratégiques

Nos points de vue théoriques sur l'analyse de discours et notre positionnement méthodologique par rapport à l'objet textuel dicteront un certain nombre d'objectifs à respecter pour l'implantation du logiciel SATO. La compréhension de ces choix stratégiques permettra de mieux comprendre les décisions informatiques qui ont guidé le développement du logiciel. En quelques mots, voici ces choix stratégiques.

1. Respect du texte intégral en tant qu'artefact. Comme l'indiquions au chapitre 2, ce respect du texte intégral s'inscrit dans une certaine tradition philologique, réactualisée à l'ère du document numérique, et impose un premier choix informatique. Le corpus, en tant qu'édition électronique, doit être une donnée non modifiable, ce qui correspond à des fichiers en lecture seulement. Ce choix méthodologique a par ailleurs un avantage économique en ce qu'il permet un accès partagé à ces fichiers par tous les lecteurs analystes utilisant un serveur donné.

2. Respect du travail de l'analyste. Le travail d'analyse textuelle par le chercheur individuel produit des documents d'annotation sur la ressource primaire. Ces documents secondaires, fichiers de propriétés annotant le corpus et son lexique, annotations structurelles et données textométriques, ont aussi un statut d'artefact portant la signature de l'analyste. Le partage de ces ressources peut se traduire, sur le plan informatique, de diverses manières. Une première façon de faire est d'adjoindre ces ressources secondaires à la ressource primaire comme s'il s'agissait d'une édition augmentée non-modifiable, mais *annotable* par d'autres documents secondaires. La deuxième façon de procéder consiste à appliquer une relation d'héritage statique permettant de prendre une copie des fichiers de propriétés du corpus partagé afin de permettre la modification directe de l'annotation héritée. Dans SATO, cela se traduit par l'héritage des propriétés

lors de l'accès initial à un corpus en mode partagé. Il s'agit en fait d'une extension de l'héritage statique, décrit à la section 2.1, et qui permet de créer une propriété à l'intérieur de SATO par héritage d'une propriété existante. Ici, l'héritage s'exécute de façon automatique lors de la consultation d'un corpus dans un espace partagé. Les fichiers d'annotation du corpus *emprunté* sont alors copiés dans l'espace de travail de l'utilisateur qui pourra les modifier à sa guise.

3. Respect de l'itérativité. Comme expliqué dans le chapitre 2 présentant les fonctionnalités de SATO, le modèle SATO repose sur la construction d'une relation d'héritage dynamique entre les formes lexicales, agissant à titre de classes, et leur instanciation en contexte à titre d'occurrences particulières. L'algorithme de découpage du fichier de caractères en formes lexicales est guidé par divers mécanismes.

   Le mécanisme qui a priorité est le marquage explicite des formes en contexte selon une certaine syntaxe de balisage. Il peut s'agit de la syntaxe SATO actuelle avec ses balises **(* et **)*. Il peut s'agir des balises XML-TEI *<w> </w>*, etc. Mais, au-delà du découpage du flux de caractères, la constitution de la classe lexicale est aussi sensible à des règles de normalisation, par exemples celles qui concernent la casse. Il faut aussi tenir compte de règles de normalisation Unicode permettant de créer des classes rassemblant des chaines qui diffèrent en termes de séquences d'octets.

   Enfin, la constitution des classes lexicales est sensible aux annotations lexicales qui permettent de créer des classes lexicales distinctes pour des chaines identiques, mais qui portent des traits distincts. Il peut s'agir de la langue, qui distingue des homographes selon leur appartenance à la langue de l'énoncé. Il peut aussi s'agir de traits sémantiques qui distinguent des formes lexicales homographes selon des systèmes de traits constitutifs d'une édition donnée du corpus.

   L'établissement d'une édition du corpus, *riche* du point de vue lexical, est, le plus souvent, un processus itératif qui intègre un travail d'analyse allant au-delà de la simple transcription. Le *respect de l'itérativité* signifie donc ici que le corpus en format SATO doit pouvoir être exporté sous forme de flux de caractères constituant une nouvelle édition enrichie qui pourra, à son tour, être soumise à SATO pour un approfondissement de l'analyse. Cette nouvelle édition, qui constitue un artefact distinct de la saisie

originale, pourra conserver les attributs du corpus de départ, ou, si l'analyse le justifie, en proposer une transformation raisonnée.

4. Principe d'économie. Comme le rapport au texte doit rester dans le mode *dialogue avec le corpus,* le temps de réponse à une commande doit être suffisamment court pour permettre l'interactivité. Cela signifie qu'une attention particulière devra être portée à l'efficacité du programme, sans négliger la richesse du modèle. Certes, l'évolution du matériel informatique fait en sorte que l'on dispose de quantités de plus en plus grandes de ressources :espace mémoire, espace disque et puissance de calcul. Cependant, l'évolution du matériel autorise et appelle également l'élargissement du volume des données. En d'autres mots, la masse documentaire nous rattrape et rend tout aussi pertinent le principe d'économie nécessaire au maintien de l'interactivité des traitements.

5.2 Le modèle de données de SATO

Le *plan lexique-occurrences de SATO*, s'il permet de décrire les fonctionnalités du logiciel, ne suffit pas, cependant, à décrire les structures organiques qui supportent l'implantation du modèle fonctionnel. En effet, il faut y ajouter des considérations d'optimisation informatique, mais aussi les exigences reliées à un modèle de traitement basé sur le principe d'une annotation incrémentielle respectant l'intégrité de la *ressource primaire*, vue ici sous forme d'un corpus électronique établi sous la responsabilité du chercheur agissant ici comme éditeur de la ressource numérique. Le terme *primaire* doit donc être compris ici au sens littéral de *matière première* d'une analyse qui le prend comme objet, sachant bien qu'il s'agit d'un construction relevant de décisions de l'éditeur, y compris dans le paramétrage de l'algorithme de segmentation en *mots* (token).

SATO est basé sur le principe de ce que l'on qualifie aujourd'hui, dans l'univers du balisage XML et TEI, d'*annotation débarquée externe* (*standoff annotation*). Cette annotation est notamment utilisée comme mécanisme de liage entre un élément et une structure de traits (cf. TEI P5: *Feature Structures*; ISO 24610-1, 2006; ISO/DIS 24611, 2008). Plusieurs expressions ont été proposées pour traduire en français le terme anglais : annotation *déportée, débarquée*, etc. Ces expressions réfèrent surtout à la forme que prennent ces annotations à l'intérieur de documents XML. Nous utiliserons plutôt ici le terme d'*annotation externe* pour

bien marquer que ce qui est visé ici est la séparation physique entre le texte brut et l'annotation analytique portant sur la ressource primaire. Ainsi, dans SATO, la notion de propriétés se matérialisera, le plus souvent, sous forme de fichiers de propriétés distincts mais dépendants de la ressource annotée.

Par ailleurs, il faut signaler que l'*annotation interne* ou *embarquée* est le format courant de SATO lorsqu'il s'agit d'exporter en mode texte un état enrichi du corpus susceptible d'agir à son tour comme *document primaire* pour une analyse ultérieure. C'est ce format qui correspond aux *propositions Sacacomie* (Daoust et Marcoux 2005). Mais il est aussi possible de maintenir l'*annotation externe* en exportant les propriétés dans des fichiers d'annotation utilisant les structures de traits ou autres. Pour reprendre la terminologie du TEI, on peut *internaliser* ou *externaliser* les annotations (TEI-P5 chapitre 11.9).

Le processus d'annotation étant itératif, on peut convenir que certains systèmes d'annotation sont de l'ordre de la transcription et de l'établissement du corpus électronique. Dans ce cas, la décision d'en faire des fichiers indépendants relève davantage de critères d'optimisation que de positions méthodologiques. Ce qui relève cependant de considérations méthodologiques est le fait de considérer ces propriétés comme étant *modifiables* ou non en phase d'analyse. Ainsi, les propriétés *page* et *commentaire* sont considérées comme non modifiables car intrinsèquement liées au corpus. La propriété *page* explicite la référence d'un texte dans le corpus (nom du document jusqu'à numéro du mot dans la ligne si nécessaire) tandis que la propriété commentaire identifie les segments du corpus qui seront exclus du plan lexique-occurrences.

Des propriétés qui découlent directement de la prise en charge du modèle fonctionnel de SATO, par exemple *Fréqtot* qui compte le nombre d'occurrences dans le corpus de chaque forme lexicale, sont non modifiables. Par ailleurs, la propriété *Édition* qui marque les attributs de présentation en contexte (*token*) de la forme lexicale est considérée comme modifiable, et prend de ce fait l'attribut d'annotation externe. Certes, plusieurs valeurs de cette propriété sont calculées sur la base du décodage des fichiers en format texte à l'origine de la construction de la matrice lexique-occurrences. Mais, ces traits peuvent aussi être précisés en cours d'analyse, par exemple pour marquer la différence entre majuscule de ponctuation et majuscule de nom propre.

La transformation du texte brut effectuée par SATO à partir des fichiers textuels en entrée produit une série de fichiers portant le même nom que le fichier corpus mais avec des suffixes différents Le fichier source en format SATO (suffixe *.sat* ex. *xxx.sat)* comprend une entête spécifiant les paramètres de segmentation en mots selon les diverses langues (alphabets) utilisées dans le corpus. Il définit aussi les systèmes de propriétés utilisés dans l'édition soumise à l'analyse de même que les métacaractères utilisés dans le codage. On peut aussi préciser des paramètres de segmentation automatique en lignes et en pages s'il y a lieu. Suivent ensuite les divers documents faisant partie du corpus avec un identificateur qui fera partie de la référence de pagination. Le contenu du document peut aussi être inclus dans le corpus à partir de fichiers textes indépendants. Au terme du décodage du texte source par SATO, on retrouvera les fichiers suivants.



Fichiers internes de SATO 4.3 (5.2a, définition)

Fichiers non modifiables :

xxx.tex qui contient le lexique et les occurrences ;
xxx.pag qui contient le catalogue des documents et des pages ;
xxx.pco qui contient la pseudo propriété *commentaire* ;

Fichiers modifiables :

xxx.pro qui contient les propriétés symboliques et entières ;
xxx.fsi qui contient les propriétés en format libre ;
xxx.ini qui contient les définitions de propriétés et de l'information de configuration ;
xxx.jou, un fichier en format texte qui contient le journal de toutes les opérations effectuées sur le corpus.

SATO offre une compatibilité ascendante qui permet la relecture des fichiers binaires créés depuis le début de la version 3 du logiciel en 1984. C'est donc dire que le système doit encore supporter des contraintes issues d'anciens systèmes d'exploitation et de matériel informatique offrant peu de ressources en termes de mémoire vive et d'espace-disque. La structure des fichiers est donc encore caractérisée par des considérations d'économie d'espace dans la représentation des données. En dehors de la distinction entre données modifiables et non-modifiables, la répartition des données internes en divers fichiers n'a pas de réel enjeu, sinon des particularités d'implantation qui favorisent la séparation de données de formats différents dans des fichiers différents. Ainsi, les propriétés en format libre (annotation libre) utilisent un format séquentiel indexé alors que les propriétés symboliques (modales, catégorielles,

ensemblistes) utilisent un format binaire en accès direct dont la position dans le fichier se calcule facilement puisqu'elles affectent une forme lexicale ou une occurrence unique numérotée en continue. Le fichier de définitions («.ini») correspond pour sa part au format standard des fichiers Windows de ce type, proche parent du format Unix.



Représentation interne du corpus dans SATO 4.3 (5.2b, définition)

Le fichier «.tex» est un fichier binaire qui contient dans l'ordre :
1- Un entête ;
2- Les caractères utf-8 des formes lexicales triées d'après le contenu binaire des bytes ;
3- Les fréquences des formes lexicales ;
4- Les numéros de la première occurrence de chaque forme (modulo 2**16) ;
5- Les numéros de lexèmes selon l'ordre de tri alphabétique Unicode ;
6- Suite des occurrences et des têtes de ligne.

Le rangement des formes lexicales par tri interne des caractères permet de réduire la taille du lexique sur disque en ne conservant que la variation entre deux formes consécutives. La partie gauche de la chaine de caractères de la forme, identique à celle de la forme précédente, n'est pas répétée. Le premier caractère d'une forme, si sa valeur interne est dans la plage numérique de 0 à 32, sera alors considéré comme une indication de la longueur de la sous-chaine gauche partagée entre les deux formes lexicales.

Chaque occurrence a une représentation binaire occupant 4 octets (32 bits). Les deux premiers octets seront interprétés comme un entier non signé représentant le numéro de la forme lexicale instanciée par cette occurrence. Les deux autres octets seront interprétés comme un entier non signé considéré comme une adresse relative (modulo 2**16) permettant de calculer la position de la prochaine occurrence de la même forme lexicale. Cette composante agit donc à la manière d'un index utilisé essentiellement à des fins d'optimisation permettant au logiciel de parcourir rapidement la chaine des occurrences d'un lexème donné.

Une tête de ligne est considérée comme occurrence de ligne et s'inscrit, à ce titre, juste avant le premier mot de la ligne. Dans sa version actuelle, le lexique de SATO est limité à 60000 entrées. La tête de ligne est reconnue par le fait que les deux premiers octets représentent un nombre supérieur à 60000. La plage numérique de 60001 à 65535 permet de conserver l'adresse relative de la prochaine tête de ligne. Les deux derniers octets de la tête de ligne sont

utilisés pour conserver le numéro de la ligne en référence à l'édition sur support papier, si elle existe.

La distinction entre fichiers modifiables et non-modifiables, correspondant à la distinction entre texte brut et annotations externes ajoutées par le lecteur-analyste, se double d'une stratégie de partage et d'héritage. Lorsque les fichiers internes de SATO sont générés, on se retrouve avec un état initial de description correspondant à une édition donnée du corpus. Cette ressource peut évoluer dans l'espace local du lecteur. Mais elle peut aussi être partagée. Dans ce cas, la partie non-modifiable du corpus restera unique alors que plusieurs lecteurs pourront y accéder concurremment. Ce n'est pas le cas des fichiers d'annotation. Le lecteur qui accède au corpus en mode partage en recevra une copie dans son espace de travail personnel. Il héritera de l'annotation du corpus dans l'état où il se trouve au moment du partage. La suite des opérations d'annotation et de catégorisation s'appliquera sur cette copie personnelle des fichiers de propriétés.

Il est à noter que le fichier journal ne fait pas l'objet d'un héritage initial lors de partage. Un nouveau journal sera créé dans l'espace local du lecteur afin de noter ses opérations personnelles sur le corpus.

5.3 Une architecture client-serveur

Avec la généralisation des interfaces d'utilisation en mode graphique, les logiciels héritant des interfaces en mode caractère étaient mal reçus par les utilisateurs, en particulier par les nouveaux utilisateurs qui n'avaient pas connu les interfaces en mode caractère. Cette réflexion sur les interfaces était un des aspects du projet *Visibilité* (voir 4.8) dont les objectifs visaient, plus largement, la diffusion des méthodes de l'analyse de texte par ordinateur, en particulier celles qui se sont développées dans le sillage de l'utilisation du logiciel SATO.

Cette réflexion devait tenir compte d'un ensemble de contraintes.

1. Un parc d'ordinateurs varié composé de McIntosh, et de PC de puissance variable faisant appel à plusieurs systèmes d'exploitation en évolution constante ;

2. Un modèle de traitement basé sur une explication des opérations avec journalisation et qui fait appel à un langage de commande souple permettant la création de scénarios ;

3. La précarité des ressources pour le développement et l'entretien du logiciel.

Au cours des années, le logiciel SATO avait migré d'une architecture serveur-terminaux de type *telnet* à une architecture micro alors largement dominante. Mais, dans le contexte du système DOS, l'interface était demeuré en mode caractères, laissait ouvert le problème du passage au mode graphique.

D'un autre côté, la popularité grandissante du Web amenait déjà, à travers l'usage des navigateurs, un modèle d'interface imposant ses propres habitudes au-delà des particularités des systèmes d'exploitation. L'utilisateur, quelle que soit la plateforme matérielle qu'il utilise, est donc initié à une culture Web qui devient incontournable. HTML a d'abord été conçu comme un format de diffusion de documents hypertextuels. Cependant, il est possible d'utiliser le protocole comme outil d'interface entre un programme et l'utilisateur dans le contexte d'une architecture client-serveur.

Ce modèle d'interface, basé sur le protocole *HTTP* (*Hyper Text Transfer Protocol*) se présente d'abord comme un mode de consultation documentaire avec des liens hypertextuels reliant les documents entre eux au moyen d'un mécanisme de pointage, les *URI* (*Uniform Resource Identifier*). Basé sur un modèle *client-serveur*, le protocole permet de concevoir l'outil d'affichage et de contrôle *du côté client* comme un logiciel autonome et indépendant des données et applications consultées. Il peut s'agir du navigateur Web traditionnel, ce qu'a privilégié SATO jusqu'à maintenant, ou d'un navigateur Web spécialisé. Même si, en 1996, on était encore dans une période de développement un peu anarchique du langage *HTML* (*Hypertext Markup Language*) exploitant le protocole *HTTP*, on pouvait identifier un ensemble stable de balises et de fonctionnalités permettant de faire appel aux navigateurs Web fournis par le marché. Dans ce contexte, on pouvait envisager de faire l'économie des interfaces propriétaires (Windows, Mac-OS ou X11) pour se concentrer sur l'utilisation de protocoles et de langages génériques (HTTP et HTML).

La motivation derrière ces choix technologiques va cependant bien au-delà des contraintes techniques liés aux interfaces propriétaires. Elle rejoint un préoccupation constante pour la formation et la diffusion des méthodes informatisées d'analyse de texte, ainsi que pour l'application de ces méthodes pour valoriser les corpus publics. Une implantation d'outils comme SATO dans le cadre du Web vise donc directement cet objectif de diffusion-formation. Par l'utilisation d'Internet, il est possible d'entrevoir un accès standardisé à des outils d'analyse de

texte, à des corpus, à des bases de données lexicales, à des fiches méthodologiques, etc. De plus, cette approche permet une exploitation à distance de gros corpus institutionnels : banques de lois et procédures, de jurisprudence, discours politiques, articles scientifiques, journaux, etc. Outre l'accès à ces données à travers les systèmes classiques de bases de données textuelles, ce qui est visé, c'est l'accès à des analyseurs plus complexes supportés par un logiciel générique tel SATO.

L'utilisation des protocoles Web pour contrôler à distance un logiciel interactif comme SATO, implique cependant qu'on *torde* un peu ces protocoles pour les utiliser au-delà de la consultation documentaire pour laquelle ils ont d'abord été conçus. Ainsi, dans le protocole *HTTP*, chaque requête est autonome et le serveur ignore le contexte historique de la requête. À l'opposé, l'interface des applications logicielles répond, en général, à une séquence d'opérations contrôlant un processus de calcul séquentiel avec des variables qui définissent l'état de l'automate. Ainsi, les pages affichées par le navigateur devaient correspondre, dans notre cas, à des résultats commandés à la volée plutôt qu'à la consultation de pages Web statiques.

L'Internet ne se résume pas à l'espace Web. Mais, le développement fulgurant de cet espace a forcé les informaticiens à s'accommoder des limites du protocole *HTTP* en développant une variété de méthodes pour prendre en compte le contexte des requêtes empruntant le protocole. C'est la voie que nous avons entrepris d'explorer, dès le milieu des années 1990, en décidant d'utiliser les navigateurs Web standards comme outils uniques pour le déploiement d'une version Web de SATO. Ainsi, on pouvait séparer la couche interface de la partie proprement algorithmique du logiciel. On voulait aussi s'appuyer sur des architectures ouvertes permettant de combiner des traitements autonomes, rendus accessibles grâce à une interface standard indépendante du logiciel et des *API propriétaires*. D'autres développeurs de logiciels d'analyse textuelle ont aussi expérimenté cette approche. Ainsi, TACT (Textual Analysis Computing Tools), le concordancier de John Bradley, se voit doté d'une interface Web sous le nom de *Tact-Web* (Bradley et Rockwell, 1995; Rockwell et coll. 1997) permettant d'interroger la version DOS du logiciel. Au cours des années 2000, d'autres développeurs universitaires se tourneront vers l'architecture Web pour donner accès à des outils d'analyse de texte par ordinateur. Citons en particulier Weblex (Heiden 2002) et TAPoR (Rockwell et coll. 2002).

Comme la plupart des projets de l'époque, nous avons choisi d'utiliser le protocole *Common gateway interface* (CGI) de la norme *HTTP* pour implanter le dialogue entre le navigateur Web et SATO. Rappelons qu'à cette époque, pourtant pas si lointaine, les langages de *scriptage* facilitant la gestion d'applications par le Web n'en étaient encore qu'à leur début. Par exemple, la première version du langage PHP n'a été rendue publique qu'en 1995 et se présentait comme un ensemble d'exécutables CGI écrits en C et devant remplacer une bibliothèque de scripts Perl .

Notre modèle d'implantation Web de SATO devait relever plusieurs défis.

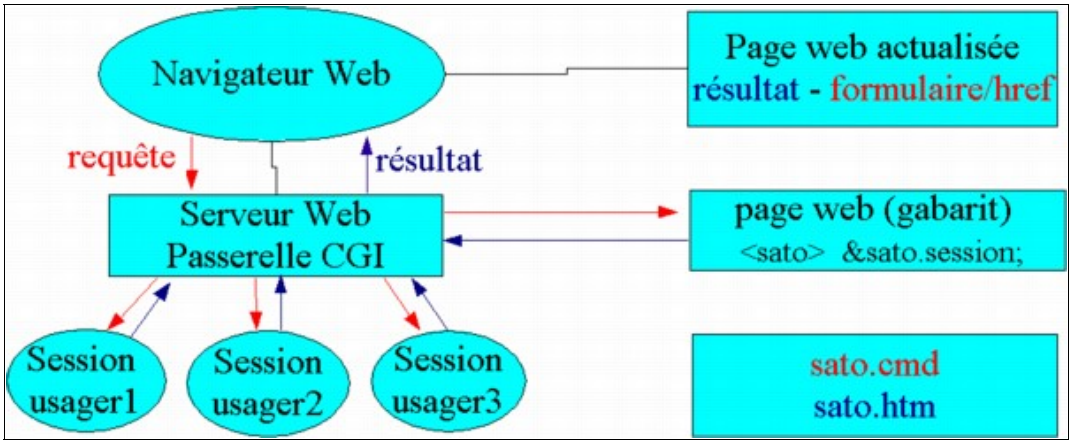
1. Proposer une ergonomie Web simple mais qui ne sacrifierait pas trop l'interactivité d'une application micro ;

2. Développer un protocole suffisamment général pour qu'on puisse l'utiliser aisément pour fédérer divers modules de traitement, notamment des analyseurs statistiques et linguistiques ;

3. Concevoir une stratégie évolutive permettant de disposer de prototypes opérationnels qu'on pourrait améliorer au cours des ans.

Le schéma général du dialogue *client-serveur* est assez classique (figure 1). Par un hyperlien ou un bouton de navigation inscrit dans une page Web, l'utilisateur fait une requête pour accéder à une page HTML. Cette page HTML est en fait un *gabarit* qui contient des balises supplémentaires spéciales qui seront interprétées par le programme *passerelle* qui reçoit la requête. En fait, ces balises seront interprétées comme des instructions de traitement qui seront passées à un programme. Le résultat de la commande sera ensuite réinjecté dans le gabarit en lieu et place de la balise. La passerelle agit donc comme intermédiaire entre la requête Web et l'application de traitement. Elle met en forme la requête, la transmet à l'application de traitement, et insère la réponse du logiciel dans le gabarit qui devient ainsi une page actualisée dans le contexte de la chaine de traitement.

Sessions client-serveur dans l'architecture Web de SATO 4.3 (5.3a, figure)



Après quelques expérimentations, on est donc arrivé à une architecture basée sur les éléments suivants.

1- L'utilisateur doit rester maitre de ses traitements et de ses données. On doit donc lui fournir un équivalent sur le serveur du bureau de l'usager sur le poste de travail : ouverture de session, gestion de fichiers, accès à des ressources documentaires (manuels, procéduriers, notes de cours, forums, etc.) et choix des traitements.

2- Le choix des traitements implique que l'utilisateur puisse démarrer des applications sur le serveur et dialoguer avec les processus de traitement à l'intérieur de son espace de travail : espace temporaire associée à sa session et espace permanent associé au compte de l'utilisateur.

3- Du côté serveur, on a choisi de développer une passerelle générique, de type *common gateway interface* (*cgi-bin*). Cette passerelle reçoit des requêtes soumises au moyen de formulaires HTML standards. Les paramètres transmis par le formulaire seront retransmis à l'application, sauf certains d'entre eux qui s'adressent directement à la passerelle : identification de la session, choix de la page HTML à retourner, variables intermédiaires, etc. La page HTML retournée par la passerelle, résultat d'une actualisation dynamique du gabarit HTML, contient à son tour une partie formulaire qui permettra la poursuite du dialogue.

4- Le gabarit HTML contient des balises sous forme d'instructions de traitement. Ce sont ces balises qui seront interprétées par la passerelle et dont les attributs seront transmis à l'application de calcul, par exemple une instance du logiciel SATO. Ces instructions de

traitement inscrites dans la page Web, permettent de démarrer une application et/ou de lui transmettre une requête sous la forme d'un fichier en format texte (par exemple *sato.cmd*). L'application produira une réponse affichable déposée dans un fichier (par exemple, *sato.htm*) en format texte contenant, ou pas, du balisage HTML ou XML. Le programme d'application détruit le fichier de commandes pour indiquer à la passerelle qu'il a répondu à la requête. Le contenu du fichier de réponse est inséré en lieu et place de l'instruction de traitement et la page HTML actualisée est retournée à l'utilisateur. Le gabarit connait le format de la réponse et peut ainsi l'encadrer dans les balises HTML appropriées : champ d'un formulaire, tableau, balise de préformatage (<pre>), etc. On pourrait aussi prévoir une étape de filtrage permettant une mise en forme du résultat par une librairie *XSLT* (W3C. XSL Transformations, 1999) ou autre. Les autres fichiers de résultats produits par les traitements seront normalement inscrits dans le répertoire permanent de l'utilisateur et pourront servir d'entrées pour d'autres traitements.

5- Le programme de traitement est susceptible de recevoir une commande partielle, par exemple dans le cas où l'usager attend de sélectionner une option dans un choix multiple dont les rubriques dépendent d'étapes antérieures. Dans ce cas, le résultat produit par le programme de traitement prendra la forme d'un formulaire HTML qui, retourné comme réponse à l'usager, lui permettra de compléter sa commande. C'est de cette façon que l'on gère des dialogues qui dépendent d'un contexte de traitement géré par l'application.

Ce protocole client-serveur, une fois rôdé, simplifie grandement la mise au point des programmes de traitement qui peuvent être écrits dans n'importe quel langage. Il est aussi possible de déployer ces programmes sur plusieurs ordinateurs et systèmes d'exploitation à partir du moment où les fichiers de commandes, de résultats et de contrôle peuvent être acheminés d'un ordinateur à l'autre. Ainsi, au Centre ATO, on utilise un serveur SAMBA sur un ordinateur SUN qui émule un réseau Microsoft donnant accès à des répertoires logés sur un ordinateur distant. Avec un protocole de transfert plus élaboré, par exemple des appels de procédure distantes de type *XML-RPC* (XML-RPC Specification, 1999, 2003), on pourrait envisager de faire collaborer non seulement des applications distinctes, mais aussi des centres de calculs distants. À l'inverse, on peut déployer cette architecture sur un seul PC combinant un navigateur Web, un serveur Web, la passerelle CGI et les logiciels d'application.

5.4 Modèle de l'interface

La conséquence directe de l'architecture client-serveur décrite à la section précédente est la totale indépendance entre SATO, en tant que logiciel de calcul basé sur un langage de commandes, et l'interface usager qui interroge SATO en utilisant le protocole HTTP du Web. C'est donc dire qu'il n'existe pas actuellement d'interface native Windows, sinon une console Windows primitive destinée au développeur. Cette console permet d'entrer une commande dans le fichier *sato.cmd*, et de récupérer les résultats inscrits dans *sato.htm*.

L'interface opérationnelle utilisant HTTP vise aussi à envoyer des commandes à SATO, ou à tout autre programme adhérant à ce protocole, et à récupérer le résultat produit. Ce résultat prend ici la forme de pages Web composites comprenant diverses fenêtres, boutons et hyperliens, tout cela dans le contexte d'une session de travail interactive et continue. La figure suivante présente une page HTML typique de l'interface actuelle de SATO.

Illustration de l'interface HTML de SATO (5.4a, figure**)**



La fenêtre de gauche est utilisée pour afficher le menu principal du programme. La fenêtre de droite contient les formulaires des commandes et les résultats, comme c'est le cas dans l'exemple. Chaque mot est un hyperlien qui affiche dans la fenêtre du bas le menu de catégorisation qui permet de révéler toute l'information sur l'objet et d'y appliquer certaines fonctions, dont la catégorisation et l'affichage des contextes courts (*kwic*). Cette fenêtre est aussi utilisée pour afficher le Manuel, l'aide contextuelle et le journal.

En 2010, la notion de session de travail Web fait partie de la réalité quotidienne de tout internaute qui fait des transactions par le Web sur son compte bancaire ou un quelconque réseau social. Mais, il y a quinze ans, la chose était moins évidente et les outils de programmation Web étaient beaucoup moins développés. Voilà pourquoi nous avons été amenés à développer notre propre langage de programmation d'interface Web. Ce langage de *scriptage* est interprété par un module informatique (*satox.exe*) sur le pseudo répertoire standard */cgi-bin* du serveur Web. La notice technique *5.3b* décrit le fonctionnement de cette passerelle, pierre angulaire de la programmation de nos interfaces.



Guide de programmation des interfaces HTML (5.3b, notice technique)

L'interface Web au logiciel SATO est assurée par une couche logicielle indépendante de SATO. Le dialogue entre l'utilisateur et le logiciel d'analyse de texte fait appel à un protocole HTTP selon des modalités décrites ici.

Principes de fonctionnement de la passerelle s*atox.exe.*

Le programme *satox.exe* agit comme une passerelle générale qui permet de contrôler une session SATO ou tout autre programme qui se conforme au même protocole. La passerelle permet de démarrer un programme ou de transmettre des commandes à un programme déjà en exécution. Le module *satox.exe* est un programme exécutable écrit en Pascal qui est démarré par le serveur Web suite à une requête de l'utilisateur au moyen d'un logiciel *client*. En général, le *logiciel-client* est un navigateur Web. La passerelle dialogue avec le serveur Web en utilisant le protocole standard CGI-BIN. La passerelle *satox.exe* termine en

retournant un fichier qui pourra incorporer les résultats produits par un programme démarré par la passerelle. Des balises spéciales inscrites dans ce fichier servent à contrôler le dialogue avec les programmes qui tournent dans l'espace de travail de la session. C'est aussi une de ces balises spéciales qui permettra de mettre fin à une session. Du point de vue de l'interface, une session de travail avec SATO prend donc la forme d'un enchainement d'appels à la passerelle *satox.exe* qui actualisent des pages Web servant de gabarit. Les fichiers HTML retournés par la passerelle contiennent des hyperliens et bordereaux HTML qui permettront de rappeler à nouveau la passerelle et de contrôler ainsi les prochaines étapes de la session de travail. Aussi, par l'utilisation des fenêtres multiples (cadres), l'utilisateur se verra offrir un ensemble d'hyperliens permettant de contrôler sa session de travail.

Balises et entités SATO interprétées par la passerelle.

La passerelle *satox.exe* agit de deux façons. D'abord, elle décode les paramètres de la requête CGI, via les méthodes GET et POST du protocole HTTP. Les données décodées serviront à bâtir le fichier de commandes à soumettre au programme démarré par la passerelle. Ensuite, la passerelle interprète le contenu du fichier, généralement une page HTML ou XML, à retourner en réponse à la requête, fichier qui contient des *balises* et des *entités* spéciales interprétables par la passerelle *satox.exe*.

Les balises interprétées par le programme *satox.exe* sont des *instructions de traitement* SGML/XML. Les instructions de traitement sont des balises qui prennent la forme suivante : *<?cible arg1 arg2 … ?>*. Le symbole *cible* identifie l'application qui devra interpréter le contenu de la balise avec ses divers paramètres *arg1 arg2....* Dans notre cas, la cible est *satox* alors que les paramètres transmis suivent la syntaxe régulière des attributs SGML/HTML ou XML/XHTML. Voici la liste des instructions de traitement qui pourront être interprétées par *satox.exe*.

- ***<?satox txt=... action=... prog=... exe=... ?>*** Cette instruction est la pierre angulaire de l'interface. C'est cette instruction de traitement qui se charge de transmettre une commande à un programme et à récupérer les résultats produits par le programme. Le texte de la commande à transmettre peut

provenir d'une interprétation des paramètres passés à *satox.exe* lors de la requête CGI. Elle peut aussi d'exprimer directement comme valeur de l'attribut ***txt*** si présent. Si on a ***action****="annuler"*, la commande n'est pas exécutée. Si on a ***action****="taire"*, la commande est exécutée mais le résultat de la commande est ignoré. La *valeur actualisée* de l'attribut *action* pourra donc servir de variable de contrôle de l'exécution de l'instruction de traitement.

La passerelle communique avec le programme en exécution en utilisant des fichiers sur le répertoire de travail de la session. Le nom des fichiers est donné par la valeur de l'attribut ***prog*** (*sato* par défaut) complété par différents suffixes. Ainsi, si on assume *prog="sato"*, le contenu de la requête sera écrit dans le fichier *sato.cmd*. Cette écriture terminée, la passerelle détruit l'ancien fichier *sato.htm* indiquant au programme en exécution de lire le contenu de *sato.cmd* et d'écrire ses résultats sur *sato.htm*. Une fois qu'il a terminé sa tâche, le programme en exécution détruira *sato.cmd* donnant à son tour le signal à *satox.exe* pour qu'il relise le fichier *sato.htm* et en insère le contenu en lieu et place de l'instruction de traitement. L'attribut ***exe*** permet de faire démarrer un programme, s'il n'est pas déjà en exécution et en attente du fichier de commande.

- ***<?satox type="bloc" action=... ?>*** ***<?satox type="/bloc" ?>*** Cette paire de balises permet d'encadrer un bloc de texte. Si on a ***action****="annuler"*, le texte à l'intérieur du bloc sera ignoré. Comme la valeur de l'attribut *action* peut être donnée au moyen d'une variable (*entité SATO*), cela permet d'activer ou de désactiver des parties du fichier de sortie au moyen d'un formulaire qui donnera la valeur de la variable au moment de l'appel à *satox.exe*.

- ***<?satox type="fichier" src=... ?>*** Cette instruction de traitement sert à insérer le contenu du fichier ***src*** à la place de la balise dans le fichier HTML ou XML à retourner.

- ***<?satox type="fin" action=... ?>*** Cette balise provoque la fin de la session

Web. Elle insère l'hyperlien (*href*) de retour à la page de départ, saut si ***action***=*taire*. Si ***action***=*annuler*, il n'y a pas d'insertion de l'hyperlien de retour ni de fermeture de session. Il s'agit en fait d'une instruction qui force le nettoyage des fichiers qui pourraient rester d'une session précédente qui n'aurait pas été fermée correctement.

Outre l'exécution des instructions de traitement, *satox.exe* actualise les *entités SATO*. Les entités en SGML/XML sont des *alias* qui désignent symboliquement un contenu qui devra prendre la place du symbole au moment de la lecture du fichier. Les entités sont des chaines de caractères comprises entre les caractères «&» et «;» Dans le cas des entités dédiées à SATO, elles sont de la forme« *&sato.xxx; »* où *xxx* désigne un des identificateurs dont la description suit. Les entités peuvent être vues comme des noms de variables que *satox.exe* remplacera par leur valeur telle que définie au moment de l'appel de la passerelle.

- *&sato.session;* est une entité utilisée pour désigner la session courante. Elle sera remplacée, lors de l'appel de *satox.exe*, par l'identificateur de la session, généralement le nom du répertoire de l'usager sur le serveur.

- *&sato.satorep;* Cette entité désigne le répertoire où est installé SATO (ex. *c:\sato*).

- *&sato.usagerrep;* Cette entité désigne le répertoire associé a l'usager sur le serveur. Par exemple, *<fichier src=&sato.usagerrep;journal.htm>* indiquera d'insérer le contenu du fichier *journal.htm* se trouvant sur le répertoire de l'usager référé par l'entité *&sato.usagerrep;*

- *&sato.v0; &sato.v1;* ... *&sato.v9;* Ces entités, dans le formulaire à retourner, représentent le contenu des paramètres *v0, v1* … *v9* transmis lors de l'appel de la passerelle. Typiquement, il s'agit de champs dans le formulaire Web d'appel. Ils seront récupérés dans la page Web retournée grâce aux *entités SATO*. Un fichier de configuration de la passerelle sera utilisé pour indiquer quels champs seront récupérables sous forme d'entités. Les champs *v0* … *v1* font partie de la configuration par défaut. On peut

aussi ajouter d'autres champs.

Paramètres d'appel du programme *satox.exe.*

Le programme cgi-bin/ *satox.exe* est généralement activé par un hyperlien dans une page Web ou par un formulaire Web. L'appel au programme est accompagné de paramètres codés avec l'hyperlien ou transmis par le formulaire. En particulier, le paramètre *formulaire* est utilisé pour indiquer quel fichier doit être retourné au *client* après que la passerelle aura interprété les instructions de traitement et les *entités SATO* contenues dans le fichier. La particularité du protocole CGI-BIN est que l'on appelle directement l'interpréteur (la passerelle) auquel on fournit le nom du fichier à retourner après traitement par l'interpréteur. Dans d'autres protocoles, le *client* appelle un fichier dont le suffixe est associé à un type d'interpréteur. Par exemple, une page Web portant l'extension *.php* activera l'interpréteur PHP sur la page demandée. Dans les deux cas, il s'agit de *pages dynamiques* qui sont produites ou actualisées par un programme qui est démarré par le serveur Web. Voici la liste des paramètres réservés pour le contrôle du programme *satox.exe*.

- *action*, si utilisé, ce paramètre doit avoir la valeur *creer* (dans le sens de créer, mais sans l'accent). Cela indique à *satox.exe* que le contenu des champs sera déposé dans un fichier de données sur le répertoire de l'usager. Ce fichier ne sera passé à aucun programme en exécution. Dans ce mode particulier du fonctionnement de la passerelle, les paramètres *v0* et *v1* seront utilisés pour construire le nom du fichier de données sous la forme *v1.v0*. Ils devront donc précéder les champs qui contiennent des données à inscrire dans le fichier.

- *exe* est le paramètre qui contient le nom, avec ses paramètres, d'un programme que la passerelle devra démarrer dans le répertoire associé à la session de l'usager; le nom du programme est un alias renvoyant à un exécutable dont la localisation devra être inscrite dans le fichier de configuration de la passerelle décrit plus loin. Ce paramètre a le même rôle que l'attribut *exe* dans l'instruction de traitement *satox* et il sera ignoré en présence de son équivalent dans l'instruction de traitement.

- *formulaire* est le paramètre qui contient le nom du fichier qui sera retourné par *satox.exe* après le traitement des balises et des entités SATO contenues dans le fichier. Si le paramètre *formulaire* est absent, *satox.exe* utilisera le fichier *satohtm.htm* (*satocat.htm* pour une requête de catégorisation). Si le suffixe du formulaire est *.htm*, il n'est pas nécessaire de mettre le suffixe dans le nom du fichier.

- *prog* est le paramètre qui définit le *canal de communication* entre le programme en exécution et la passerelle. Ce paramètre est l'équivalent de l'attribut *prog* de l'instruction *satox*. La communication entre la passerelle et le programme en exécution s'effectue actuellement au moyen de trois fichiers du nom défini par *prog* avec trois suffixes différents. À défaut de spécification explicite, le canal portera le nom *sato*. Dans ce cas, la passerelle transmet l'information au programme via le fichier *sato.cmd* et récupère les résultats produits via le fichier *sato.htm*. Aussi, le fichier de contrôle *sato.log* sera créé lors du démarrage du programme et devra être détruit par le programme appelé lors de sa fermeture. Ce fichier de contrôle est surtout utile pour les programmes qui, tel SATO continuent à tourner entre les requêtes. Cela permet de vérifier que le programme est en opération. Pour permettre à l'usager de contrôler plus d'un programme à la fois, on peut utiliser divers noms de fichier de communication pour le paramètre *prog*. Par ce dispositif, on peut disposer de plusieurs canaux de communication permettant de contrôler plusieurs programmes s'exécutant en parallèle.

- *style* est utilisé pour transmettre le nom d'une feuille de style à inscrire en référence dans le fichier retourné par la passerelle. Cela permet de définir dynamiquement le format de présentation du fichier.

- Une chaine de caractères, codée selon le protocole CGI-GET, peut être transmise dans l'hyperlien d'appel à *satox.exe*. La chaine qui suit le *?* dans l'hyperlien sera considérée comme valeur par défaut de l'attribut *txt* dans l'instruction de traitement *satox*.

Les paramètres suivants ne sont utilisés par *satox.exe* que lors de l'ouverture d'une session.

- *pseudo*. Ce paramètre est le nom de compte de l'utilisateur. Il correspond au nom du répertoire où se trouvent les données de l'utilisateur. Le programme va normaliser la valeur de *pseudo* en transformant les lettres en minuscules non-accentuées et en éliminant les caractères illégaux, en particulier les espaces. Si *pseudo* est absent ou vide, alors on ouvre une session anonyme. Une session anonyme est allouée sur le premier répertoire libre dans la séquence des noms identifiés par un nombre entier : 1, 2, ... n.
- *nom* est le nom complet de l'utilisateur (inutilisé pour les sessions anonymes). Ce paramètre n'est fournit que lors de la création du compte. Par la suite, *pseudo* suffira pour ouvrir une session au nom de l'usager.
- *pw* est le mot de passe de l'utilisateur (inutilisé pour les sessions anonymes).
- *email* est l'adresse électronique de l'utilisateur (inutilisé pour les sessions anonymes).

Si *nom* est donné mais *email* est vide, on valide un usager inscrit.

Si *nom* et *email* sont donnés, on définit un nouvel usager.

- *groupe* est le nom du compte correspondant au groupe de partage; le groupe de partage est un autre compte d'utilisateur qui autorise un accès en lecture aux corpus et autres ressources du compte partagé. L'accès doit être validé par un mot de passe (voir *pwg* plus bas).
- *langue* indique la langue à utiliser pour l'interface. Les noms des formulaires seront suffixés par la valeur de *langue*. Par exemple, *formulaire=lexique_afficher* avec *langue=es* commandera à la passerelle de chercher le fichier *lexique_afficher-es.htm* ». S'il est absent, *satox.exe* cherchera *lexique_afficher.htm*.
- *pwg* est le mot-de-passe permettant d'accéder au compte de partage nommé dans le paramètre *groupe*. Si on veut définir un mot de passe de partage

pour le compte donné par le paramètre *pseudo*, on le transmet dans le paramètre *pwg* en laissant vide le paramètre *groupe*.

Voici un exemple d'utilisation de ces paramètres dans un formulaire HTML destiné à créer un compte d'utilisateur et à ouvrir une session.

```
<FORM ACTION="/cgi-bin/satox.exe/" METHOD=POST>
<INPUT TYPE="text" NAME="pseudo">
<INPUT TYPE="password" NAME="pw">
<INPUT TYPE="text" NAME="nom">
<INPUT TYPE="text" NAME="email">
<INPUT TYPE="text" NAME="groupe">
<INPUT TYPE="password" NAME="pwg">
<INPUT TYPE="hidden" NAME="langue" VALUE="fr">
<INPUT TYPE="hidden" NAME="formulaire" VALUE="bureau_depart">
<INPUT TYPE="submit" VALUE="Appliquer">
</FORM>
```

Le formulaire d'inscription sert aussi de formulaire de modification des champs *nom* et *pw* si le couple *pseudo-pw* existe déjà.

Comme on a pu le voir, certains des paramètres transmis lors de l'appel *cgi/satox* peuvent jouer des rôles analogues aux attributs de l'instruction de traitement *satox* déjà décrite. Ils agiront alors comme valeurs *par défaut*, c'est-à-dire qu'ils ne seront considérés qu'en l'absence de redéfinition explicite sous forme d'attributs dans la première instruction de traitement *satox* inscrite dans le fichier interprété par la passerelle. On pourra donc trouver dans ce fichier une instruction de traitement réduite à sa forme la plus simple *<?satox ?>*. Dans ce cas, le contenu de l'attribut *txt* à transmettre au programme en exécution proviendra des paramètres transmis lors de l'appel *cgi/satox*. Si aucun attribut ou paramètre ne permet de donner un contenu à la commande à transmettre, c'est l'adresse IP de la requête qui sera inscrite dans le fichier *.cmd*.

Autres paramètres d'appel de *cgi-bin/satox.exe*.

En plus de paramètres réservés qui, tel *formulaire*, s'adressent directement à *satox.exe*, on peut avoir des champs qui seront interprétés comme des variables dont les valeurs seront transmises au fichier HTML retourné par la passerelle. Ainsi, comme on l'a vu, les champs *v0, v1 ...v9* seront, si présents, conservés par la passerelle afin d'être réinjectés dans le fichier de retour en lieu et place des *entités*

correspondantes *&sato.v0; &sato.v1; ... &sato.v9;* apparaissant dans le fichier transmis. Les champs *v0 … v9* servent donc de variables destinées à paramétrer le fichier retourné par la passerelle.

Outre ces paramètres à l'interprétation définie, on peut transmettre des paramètres quelconques à *cgi-bin/satox.exe*. Nous les appellerons *paramètres libres* pour les distinguer des paramètres directement dédiés à la passerelle. Les paramètres libres verront leur valeur inscrite sur le fichier *.cmd* qui sera transmis au programme en exécution. Ces valeurs seront simplement concaténées à la ligne de commande précédée d'un caractère de tabulation sauf en début de ligne. On notera que cette transmission de paramètres à *satox* joue un rôle similaire au paramètre *txt* qui, dans l'instruction de traitement *satox,* contient une ligne de commande à transmettre au programme en exécution.

Ce dispositif est un peu plus élaboré pour la création d'un fichier (cf. le paramètre *action=creer).* Chaque écriture d'un paramètre libre se termine alors par une fin de ligne. Toutefois, si un nom de champ se termine par *!*, sa valeur sera écrite précédée et suivie d'un espace sans fin de ligne. Si le nom d'un paramètre se termine par *%* , le nom du paramètre sera également inscrit dans le fichier *.cmd*. Le caractère *%* sera remplacé par un espace. Par exemple, la ligne suivante dans un formulaire HTML

<input type="checkbox" name="Alphabet%" value="fr" checked> fr (français)

provoquerait l'écriture de la chaine *Alphabet fr* dans le fichier de données si la case *fr* est coché par l'utilisateur.

Voici un exemple de requête transmise au moyen d'un hyperlien (*href*) codé selon le protocole HTML *GET*.

&sato.session;?*Texte+afficher&formulaire=texte_afficher&filtre=$&v1=Exemple

Dans cet exemple, **Texte+afficher* est une chaine de caractères qui sera transmise au programme en exécution via le fichier *sato.cmd*. Le + sera remplacé par un espace. La valeur du champ filtre, c'est-à-dire la chaine composée du caractère *$* sera concaténée à **Texte+afficher* précédée d'un espace. Donc, la chaine **Texte*

afficher $ sera écrite dans *sato.cmd* et le résultat sera inséré dans le formulaire *texte_afficher.htm* en lieu et place de l'instruction de traitement *<?sato?>* qui devrait se trouver dans le fichier *texte_afficher.htm*. Aussi, la valeur du paramètre *v1* (*Exemple*) pourra aussi être insérée dans le formulaire en lieu et place de l'entité *&sato.v1;*.

Si le nom du fichier retourné par *satox.exe* est généralement fourni comme valeur du champ *formulaire*, il est aussi possible de le transmettre à l'intérieur d'un champ non réservé à titre de paramètre de *satox.exe*. Ainsi, il sera possible de lier le nom du fichier à retourner à la réponse fournie dans un autre champ du bordereau de saisie HTML. Le nom du formulaire doit, dans ce cas, être codé dans la valeur du champ sous la forme d'une chaine spéciale qui a la syntaxe suivante: ((*nom-du-fichier-à-retourner*)). Un nom de formulaire transmis à l'intérieur de la valeur d'un autre champ aura priorité sur la valeur définie dans le champ réservé *formulaire*.

Un autre dispositif renforce la puissance de traitement de la passerelle. En effet, tout nom de paramètre peut être accompagné d'une instruction de filtrage du contenu transmis par le paramètre. Il suffit de mettre le nom du filtre entre parenthèses carrés *[ ]*. Ce nom désigne un programme inscrit dans le fichier de configuration (voir plus bas). Ce programme sera exécuté sur le contenu du paramètre en utilisant le canal *gestion.* Le résultat du programme prendra la place du contenu initial du paramètre.

Le premier appel à la passerelle *satox.exe* est destiné à ouvrir une session de travail. L'appel le plus simple peut se résumer à *http://localhost/cgi-bin/satox.exe/* qui aura pour effet, dans cet exemple, d'ouvrir une session anonyme en mode expert sur le serveur local. Une fois la session démarrée, les appels ultérieurs à la passerelle seront inclus dans les fichiers HTML retournés par *satox.exe*. Par exemple, au retour de l'appel précédent, on devrait avoir un hyperlien du type Entrer. Ces appels sont exprimés sous la forme d'hyperliens *href* ou de requêtes de type *form*. Dans tous les cas, l'adresse HTTP fournie sera une adresse relative commençant par l'identificateur de la session SATO, ce qui correspond au nom du répertoire de

l'usager. Ainsi, dans notre exemple, */1* indique à *http://localhost/cgi-bin/satox.exe/* qu'une session est ouverte sur le répertoire 1. Pour référer à ce nom de manière générale, on utilise l'entité générale *&sato.session;*. Par exemple :

Appel du formulaire gestion.htm

retournera le fichier *gestion.htm* après qu'il aura été interprété par la passerelle *satox.exe*. L'entité *&sato.session;*, une fois actualisée par la passerelle, désignera le répertoire de l'usager. Le fureteur Internet assumera que la partie gauche de l'URL sera identique à l'adresse de la page affichée, c'est-à-dire l'adresse de la passerelle. L'adresse IP de l'utilisateur sera utilisée pour vérifier que la session invoquée est bien utilisée par celui qui l'a initiée. Normalement, la passerelle utilise aussi le mécanisme des témoins (*cookie*) pour s'assurer que la session n'est pas dérobée par quelqu'un d'autre sur le réseau.

En terminant, signalons un dispositif qui ne concerne pas directement la passerelle *satox*, mais qui permet de contrôler du déroulement d'une session SATO. Chaque requête transmise à SATO via la passerelle débute par un caractère de contrôle, ou paramètre d'état, qui permet d'indiquer le statut de la chaine de caractères transmise à SATO. En voici la définition.

- **!** La requête s'adresse directement à la passerelle;
- ***** La requête débute une nouvelle commande et a pour effet de terminer la commande précédente; la commande est inscrite dans le journal;
- **/** La requête débute une nouvelle commande et a pour effet de terminer la commande précédente; la commande n'est pas inscrite dans le journal;
- **.** Il s'agit d'une requête de catégorisation d'un objet (lexème ou occurrence);
- **$** Il s'agit d'une touche qui requiert une réponse immédiate;
- **x** *x* désigne ici un chiffre entre 0 et 9 et qui est indique une réponse à un menu. Ce type de requête est généré automatiquement par SATO

- > Demande la poursuite d'un affichage interrompu parce qu'il dépassait le nombre limite de lignes;
- **a** Indique qu'il s'agit d'une requête d'aide immédiate, par exemple pour afficher les définitions des propriétés sur le corpus.

Si une requête transmise par la passerelle se termine par ** et si SATO doit générer un menu pour compléter la commande, ce menu sera généré sans déclaration de formulaire et sans bouton *soumettre*, présumant que ces informations se trouvent dans le formulaire.

Fichier de configuration de la passerelle.

Le fichier de configuration de la passerelle satox.exe se trouve dans le répertoire où est installée la passerelle, généralement *cgi-bin*. Le nom du fichier de configuration est *satox.ini*. En voici un exemple

```
[Adm]
tracer=non
cookie=oui
serveur=Serveur SATO personnel
[Env]
usagerrep=c:\sato\usagers\
satorep=c:\sato
sessionrep=c:\sato\sessions\
[Exe]
admin=c:\perl\bin\perl.exe c:\sato\bin\admin.pl
apostrophe=c:\perl\bin\perl.exe c:\sato\bin\apostrophe.pl
bino=c:\perl\bin\perl.exe c:\sato\bin\bino.pl
calibrer=c:\perl\bin\perl.exe c:\sato\bin\calibrer.pl
carcv=c:\perl\bin\perl.exe c:\sato\bin\carcv.pl
gestion=c:\perl\bin\perl.exe c:\sato\bin\gestion.pl
loc=c:\perl\bin\perl.exe c:\sato\bin\loc.pl
loc0=c:\perl\bin\perl.exe c:\sato\bin\loc0.pl
locbloc=c:\perl\bin\perl.exe c:\sato\bin\locbloc.pl
perltst=c:\perl\bin\perl.exe c:\sato\bin\perltst.pl
PropToTei=c:\perl\bin\perl.exe c:\sato\bin\PropToTei.pl
sato=c:\sato\sato.exe
source-html=c:\perl\bin\perl.exe c:\sato\bin\source-html.pl
TeiToProp=c:\perl\bin\perl.exe c:\sato\bin\TeiToProp.pl
tt_tab=c:\sato\bin\tag_tab.bat
ttf_tab=c:\sato\bin\tag-french_tab.bat
ttfa_tab=c:\sato\bin\tag-french_old_tab.bat
ttfa_txt=c:\sato\bin\tag-french_old_txt.bat
[Var]
v0=
```

v1=
v2=
v3=
v4=
v5=
v6=
v7=
v8=
v9=
mot_cle=mots à chercher
satoman=http://localhost

La section *[Adm]* contient des paramètres techniques : tracer (=oui|non pour le débogage; non autrement); cookie (=oui|non pour assurer que les sessions ne sont pas volées...); serveur (pour nommer le serveur qui exécute satox).

La section *[Env]* contient le paramètre *usagerrep* qui indique la racine à partir de laquelle seront créés les espaces de travail des usagers. Si ce paramètre est absent les répertoires des usagers seront créés à partir du sous-répertoire *usagers* du répertoire sato. Notre exemple ne fait donc que confirmer la chose en donnant le chemin complet du répertoire. Les paramètres s*atorep* et *sessionrep* donnent, respectivement la localisation du répertoire où réside SATO et le répertoire de travail associé aux sessions. Les valeurs données ici en exemple correspondent à une installation locale.

La section *[Exe]* contient la liste des noms de programmes qui peuvent être démarrés par la passerelle. La valeur du paramètre correspond à la localisation exacte du programme à démarrer.

Enfin, la section *[Var]* contient la liste de variables que l'on utilise dans les fichiers HTML de l'interface SATO. La valeur définie pour chaque variable sera utilisée si une nouvelle valeur n'est pas transmise avec l'appel de la passerelle. L'appel de la variable dans le formulaire retourné par la passerelle suit la syntaxe des entités SATO: par exemple *&sato.v1; &sato.mot_cle; &sato.satoman;*

Actuellement, les fichiers produits par SATO, et retournés par la passerelle, sont généralement en format texte préformaté. Ils peuvent aussi être en format HTML,

principalement pour les formulaires permettant de compléter dynamiquement la composition des commandes en fonction du contexte.

Les chaines linguistiques accompagnant ces résultats sont gérées par SATO en utilisant un fichier de ressources linguistiques pour chacune des langues de l'interface. Dans l'ensemble de l'environnement informatique, la gestion des langues se fait donc à deux niveaux. On a d'abord une sélection par la passerelle *satox* du gabarit Web selon la langue de l'interface choisie par l'usager en ouverture de session. Ce choix linguistique est à son tour transmis par la passerelle au programme en exécution. Cette stratégie fonctionne bien. Mais, elle a l'inconvénient de disperser les solutions de présentation linguistique et de mise en page.

Dans le contexte du développement futur de SATO qui privilégie les formats XML, on pourra s'interroger sur la possibilité de concentrer ces tâches de mise en page et de présentation linguistique des résultats par l'utilisation de feuilles de transformation *XSLT*.

5.5 Un code informatique en évolution

5.5.1 Description du programme

Comme on a pu le voir dans les chapitres précédents, l'historique de développement de SATO s'étend sur plusieurs décennies et plusieurs générations d'ordinateurs. Or, par respect pour les utilisateurs, nous avons tenu à assurer une compatibilité ascendante entre versions du logiciel de telle sorte que les fichiers binaires des corpus puissent être relus et exportés vers les nouveaux formats.

Dans son état actuel, le logiciel SATO comprend environ 78 000 lignes de code, dont la moitié correspond au code principal excluant les librairies auxiliaires. Comme on l'a vu au chapitre 4, à partir de sa version 3, SATO est un logiciel complètement nouveau par rapport aux versions Fortran antérieures. Il ne s'agit pas simplement d'une réécriture dans un nouveau langage, mais d'un nouveau modèle de calcul que nous avons décidé de programmer dans un langage qui, à l'époque, était également nouveau. Le langage Pascal était, en effet, le dernier mot en matière de programmation structurée et de clarté en termes d'expression. Le compilateur original développé par Niklaus Wirth était déjà de très bonne qualité. L'arrivée du

compilateur de la société Borland sur IBM-PC a permis de jouir d'un environnement qui a constitué une référence pendant des années. Il n'est pas étonnant que Borland ait pris très tôt un virage vers la programmation orientée objet avec la création de Delphi. Cependant, cette évolution du langage était essentiellement liée à une entreprise privée qui a connu plusieurs rachats menant à des stratégies commerciales difficiles à suivre pour le développeur-utilisateur. Dans ces circonstances, nous avons décidé de ne pas adhérer au niveau dialecte avant de disposer de compilateurs tiers nous permettant un repli dans le cas où l'évolution commerciale de Delphi nous mènerait à un cul de sac. Notre stratégie aura donc été de nous inspirer de certains principes de la programmation objet sans adopter le formalisme proposé par Delphi.

En résumé, la situation de l'état du code informatique de SATO est la suivante. Pour l'essentiel, le code est développé selon le paradigme procédural. Le nom des procédures et des structures de données tend à simuler la hiérarchie des objets que l'on retrouve dans le paradigme de la programmation objet. Tout au long du développement, nous avons tenté de respecter des principes de modularité facilitant l'entretien du code. En particulier, pour maintenir la possibilité de compiler le code sous diverses architectures matérielles et sous divers compilateurs, les dépendances associées à ces diversités ont été concentrées au sein d'une couche spécifique de procédures et de fonctions utilisant le formalisme de la compilation conditionnelle pour gérer les dépendances. Il reste que le principe d'encapsulage des données, que la programmation objet permet de garantir, n'a pas toujours été appliqué de façon systématique dans le cadre de la programmation procédurale actuelle. L'encapsulage des données consiste à en négocier l'accès, en lecture ou en écriture, au moyen de procédures et fonctions (méthodes dans la terminologie *objet*) de telle sorte que l'implantation effective des structures de données soit indépendante de leur utilisation et puisse être modifiée sans remettre en cause l'ensemble du code. Cette centralisation de l'accès a pour avantage de faciliter l'entretien du code et sa documentation. Même si, dans son formalisme même, le paradigme de la programmation objet permet de garantir l'application de ce principe d'encapsulage, on peut s'obliger à l'appliquer systématiquement dans le cadre même du paradigme procédural actuel.

Nous reviendrons en 5.4.3 sur nos hypothèses concernant l'évolution du code de SATO dans les années à venir.

Sans entrer dans les détails de la programmation de SATO, voici une brève description de la structure du programme permettant d'en saisir le fonctionnement général.

SATO, comme on l'a vu dans sa description fonctionnelle, propose un modèle basé sur un ensemble de concepts simples mais fortement intégrés. Au cœur de SATO, on retrouve donc une importante librairie de procédures et de fonctions, outre cette librairie déjà mentionnée de fonctions dépendantes des système. Les commandes de SATO, telles qu'on y accède à travers l'interface usager, gravitent autour de ce cœur. On pourrait synthétiser la présentation du cœur en le divisant en sous-systèmes.

1. Système de décodage des commandes.

   Contrairement aux logiciels qui se basent sur l'interface graphique du système d'exploitation pour percevoir des événements et déclencher des actions, SATO doit effectuer un décodage syntaxique de chaines de caractères. Ce décodage répond à la fois à des contraintes lexicales, dépendantes du contexte, et à des contraintes syntaxiques assez régulières mais certainement moins rigides que celles imposées par une syntaxe de type XML par exemple. Ainsi, pour faciliter l'entrée directe des commandes, surtout utilisées dans les premières années de SATO 3, les mots clés du langage de commandes peuvent être abrégés au nombre suffisant de caractères pour qu'ils soient univoques dans le contexte. Ils sont insensibles à la casse et à l'accentuation.

   Le système de décodage s'applique aussi aux fichiers de données en format caractère : scénarios de commandes, textes du corpus à générer, dictionnaires tabulaires, etc. Plusieurs systèmes de décodage peuvent être actifs. Aussi, plus récemment, un mode de lecture XML a été ajouté. Un ensemble important de structures, de fonctions et de procédures a été développé pour gérer ce système de décodage syntaxique de fichiers de commandes et de données.

2. Système de menus HTML.

   Greffé au système de décodage syntaxique, on trouve un système de gestion de menus qui sert à générer à la volée des menus Web. Ces menus interviennent dans le processus de décodage des commandes lorsqu'un item est manquant, par exemple parce qu'il doit être spécifié en contexte en fonction d'un état donné du corpus. Le

système de menus HTML est aussi utilisé pour gérer l'interaction du menu de catégorisation de l'interface Web.

3. Système de journalisation.

   Également très près logiquement du système de décodage des commandes se trouve le système de journalisation qui permet de tracer les commandes et les annotations en ligne.

4. Système d'écriture des résultats.

   Pour l'écriture des résultats produits, SATO utilise un système élaboré de couches emboitées. Il faut comprendre en effet que les résultats ne sont pas que de simples chaines de caractères. Il s'agit plutôt d'objets d'un certain type à personnalité variable, pouvant être associés à des hyperliens, s'afficher de diverses façons et sur plus d'un fichier à la fois. La couche d'écriture des objets peut donc faire appel à son tour à un sous-système d'écriture dans un canal pouvant se traduire par des écritures au format HTML, XML, ou chaines simples. De plus, comme textes, contextes et lexiques sont générés à la volée, on doit pouvoir les formater selon une variété de formats associés au corpus et aux paramètres de présentation choisis par l'utilisateur.

5. Système d'accès au format binaire du corpus et de ses propriétés.

   Comme indiqué en 5.2, le corpus et ses propriétés résident sur plusieurs fichiers en format interne. On a donc un ensemble de procédures destinées à accéder à ces données. À ce niveau tout particulièrement, le passage à la programmation objet serait très avantageux en raison de ses propriétés de polymorphisme. Comme l'écrit Delannoy (2004),

   « Le polymorphisme permet d'obtenir un comportement adapté à chaque type d'objet, sans avoir besoin de tester sa nature de quelque façon que ce soit. La richesse de cette technique amène parfois à dire que l'instruction *switch* est à la P.O.O. ce que l'instruction *goto* est à la programmation structurée. Autrement dit, le bon usage de la P.O.O. (et du polymorphisme) permet parfois d'éviter des instructions de test, de même que le bon usage de la programmation structurée permettrait d'éviter l'instruction *goto.* » *(p. 205).*

Comme les propriétés en SATO sont typées, la programmation doit multiplier le nombre d'instructions *case*, l'équivalent du switch de C et Java, pour tester le type et la portée des propriétés pour chacune des opérations de base sur les propriétés.

6. Système de gestion de l'environnement.

   L'environnement de travail de SATO sur un corpus donné contient un ensemble de spécifications qui peuvent être modifiées à loisir : formats de présentation des divers objets textuels, définitions de ressources (scénarios, touches de catégorisation, etc.). La librairie contient donc un ensemble de procédures pour manipuler ces objets et pour en préciser le comportement.

Un module important de SATO a trait à la génération du corpus dans le format SATO. Ce module ne se contente pas de segmenter le texte en mots. Il doit décoder un ensemble de balises qui auront une influence directe sur cette segmentation en occurrences annotées, instances de classes lexicales qualifiées. Ce module est aussi responsable du tri Unicode du lexique et de la production du *quasi-index* permettant d'optimiser certaines types de fouilles en contexte.

Enfin, on retrouve les modules associés à chacune des commandes de SATO.

5.5.2 Le passage à l'Unicode.

Parmi les problèmes liés à l'évolution du code de SATO depuis les débuts de la version Pascal, celui du passage à l'Unicode a constitué un défi d'envergure, comme pour la majorité des logiciels conçus à l'époque où *caractère* rimait avec *octet*. Dans la foulée de la transition de SATO vers la norme XML, ce passage à l'Unicode devenait incontournable.

Contrairement à son prédécesseur SGML, XML a renoncé à la définition des jeux de caractères pour s'en remettre totalement à la norme Unicode. Cette norme, gérée par le consortium Unicode, a pour objectif de répertorier tous les jeux de caractères utilisés dans presque tous les systèmes d'écriture. Il propose aussi divers modes de représentation informatique de ces caractères et discute des problèmes relatifs à leur classement, en particulier celui du tri alphabétique. Le passage des logiciels à la norme XML implique donc, au minimum, de supporter la représentation UTF8 de cet ensemble universel de caractères. Cette norme permet de représenter tous les caractères en utilisant des suites d'octets de

longueur variable. Dans ce système, les 127 caractères ASCII utilisés dans des milliers de fichiers textuels utilisant l'alphabet latin non accentué, sont représentés sur un seul octet. Ces fichiers sont donc d'office compatibles avec la norme UTF8.

Les systèmes informatiques ont été confrontés depuis leur début à une pluralité de systèmes d'encodage des caractères. Le nombre de caractères différents acceptés par ces normes n'a cessé d'augmenter. Dans son évolution, et compte tenu de l'architecture des ordinateurs sur lesquels il a été implanté, SATO a utilisé des systèmes d'encodage des caractères à six, sept et huit bits par caractère. L'encodage utilisant un octet, pour représenter des jeux pouvant aller jusqu'à 256 caractères, a dominé le monde de l'informatique et un très grand nombre d'applications reposent encore sur ce type de représentation. L'adaptation des logiciels à l'Unicode ne se réduit pas, dans plusieurs cas, à un problème localisé d'écriture et de lecture de caractères dans des fichiers externes. Souvent, les algorithmes de traitement eux-mêmes doivent être repensés alors que les techniques d'optimisation adaptées aux jeux de caractères à 8 bits peuvent devenir caduques.

SATO utilise un tri binaire du lexique pour optimiser la fouille et la catégorisation lexicale. Le tri binaire est un tri sur le code interne des caractères des formes lexicales. Depuis le développement de la version 3 de SATO, on utilisait un encodage des caractères basé sur une table de codes ASCII étendue proposée par IBM pour étendre l'ASCII aux langues latines les plus usuelles. Les 127 premiers codes correspondent à l'ASCII standard représentant les caractères non accentués de l'alphabet latin, les chiffres et ponctuations simples, précédés dans les 31 positions, des caractères de contrôle non imprimables. Les codes supérieurs du jeu de 256 caractères permettent de représenter les caractères accentués majuscules et minuscules de même que certains caractères non alphabétiques comme les guillemets français.

Le tri binaire facile la compression des listes triées en partageant la portion gauche commune des caractères de deux entrées successives. Ainsi, si on a la suite *aimer* et *aimera*, on pourra réduire le représentation d'*aimera* à *#5a*, *#5* étant ici interprétée comme la reprise des cinq caractères de l'entrée *aimer*. Le passage à l'Unicode ne pose pas ici de problèmes particuliers puisque le changement des codes de caractères s'applique à l'ensemble de la liste et permet donc toujours de grouper les entrées d'après leur partie commune de gauche à droite.

Les entrées lexicales étant numérotées d'après leur ordre de tri interne ascendant, il est possible de faire une fouille binaire pour trouver une forme lexicale donnée. Il est aussi

possible d'utiliser la fouille binaire du lexique pour restreindre la fouille séquentielle de filtres qui comportent des contraintes sur la portion gauche des caractères. La fouille sur le radical *aim$*, par exemple, trouve d'abord la plage du lexique comprise entre les formes fictives *aim#1* et *aim#255* où *#1* et *#255* représentent respectivement les caractères codés 1 et 255 dans l'encodage IBM850. Le tri binaire est aussi utilisé pour gérer les dictionnaires en format séquentiel. Le dictionnaire étant généré en utilisant le même tri interne, l'application d'un dictionnaire sur un lexique se réalise par un simple parcours séquentiel de deux listes triées : le lexique et le dictionnaire. Ce mode de traitement est très rapide. Le changement de l'ordre de tri interne dû à l'adoption de l'Unicode ne permet plus d'utiliser cet algorithme d'alignement avec un dictionnaire généré sous l'ancien code. Réciproquement, un dictionnaire nouvellement généré ne pourra plus être comparé avec un corpus basé sur l'encodage IBM850. Pour respecter le principe de compatibilité ascendante, il a donc fallu abandonner la comparaison par liste triée pour utiliser une table d'indices calculés (*table de hachage*) sur le lexique afin de pouvoir accéder aux entrées lexicales dans n'importe quel ordre. Voici donc un exemple, parmi d'autres, où le passage à Unicode a impliqué un changement d'algorithme.

Le format de présentation d'un corpus aux fins de traitement par SATO prévoit un certain nombre de déclarations faisant partie de l'entête du corpus et qui sont destinées à préciser le traitement des caractères appartenant à l'une ou l'autre des langues utilisées dans le corpus. Ainsi, la déclaration *Alphabet* permet de définir ce qui, d'un point de vue linguistique, correspond à un caractère pertinent pour un système d'écriture dans une langue donnée. Il est prévu qu'on puisse définir des caractères à frappe multiple, par exemple la suite de trois points successifs "…" pour représenter l'unité *points de suspension* qui ne fait pas partie du code *iso-8859-1*.

La déclaration *Alphabet* a aussi d'autres finalités. Elle permet de définir le statut des caractères aux fins de segmentation du texte en mots. Outre la définition des séparateurs, la déclaration permet d'identifier des caractères qui terminent ou initient la constitution d'une forme lexicale, par exemple, l'apostrophe d'élision en français. La définition de caractères à touches multiples permet aussi d'apporter une solution partielle à l'identification de points ou virgules non séparateurs : par exemple *.1 .2 .3 ,0 ,1 ,3* etc. Ainsi, *3.4.5* sera reconnu comme une forme lexicale autonome si on définit ces caractères comme étant non-séparateurs. Enfin, l'énumération des caractères dans la déclaration *Alphabet* permet de définir le premier niveau

pour la constitution de la clé de tri utilisée pour opérer le tri alphabétique du lexique des formes de chacune des langues utilisée dans le corpus.

Le tri lexical, en particulier dans un contexte multilingue est, en effet, loin d'être une question triviale… Comme l'indique le rapport technique #10 du consortium Unicode sur le tri alphabétique, ce tri dépend non seulement de la langue et de la culture, à la fois du document et du lecteur, mais aussi de l'application. Ainsi, la façon de trier des noms dans un bottin téléphonique peut être assez différente de celle utilisée dans un dictionnaire. SATO prévoit donc depuis très longtemps un certain contrôle sur la façon de trier les lexiques. Certes, il ne s'agit pas d'un contrôle optimal. Nous avons été confronté à la complexité du problème et de ses solutions à la fin des années 1980 alors qu'Alain Labonté au Québec élaborait des propositions qui conduiront à l'adoption d'une norme canadienne (CAN /CSA 2243.4.1). Cependant, dans le contexte utilitaire du tri alphabétique dans SATO, nous n'avions implanté qu'une version du tri limité à deux niveaux : le premier niveau est celui des caractères minuscules non accentués alors que le deuxième niveau suit le code interne IBM850.

Lors de la génération de la représentation lexique/occurrences du corpus, SATO doit systématiquement consulter la table de caractères de l'alphabet courant afin de segmenter le corpus en mots. L'utilisation d'une table indicée par un code de caractère à huit bits permet un accès direct aux caractères simples, et un accès séquentiel aux caractères à touches multiples débutant par un même caractère. L'utilisation d'un tableau indexé par un code de caractères à 16 bits augmenterait significativement la charge sur la mémoire vive de l'ordinateur. Il nous est donc apparu préférable de modifier les algorithmes en distinguant les informations requises pour la segmentation en mots (*tokenisation*) de celles requises par le tri alphabétique des formes lexicales construites par le découpage en mots.

Pour l’identification des mots, on a besoin de dresser la liste des caractères ou chaines de caractères agissant comme séparateur, en ajout aux séparateurs universels comme l’espace. On a aussi besoin de connaitre la nature du séparateur pour chaque alphabet considéré : séparateur constituant lui-même un *token*, comme une ponctuation, les séparateurs terminaux qui s’ajoutent et terminent le *token* précédent, et les séparateurs initiaux qui initient sans le clore un nouveau *token*. Implicitement, tout caractère qui n’aura pas été identifié comme séparateur sera considéré comme constituant du *token*. La règle implicite doit tout de même être complétée par une table explicite de chaines de caractères non-séparatrices pouvant

contenir des sous-chaines séparatrices. Comme l'algorithme de *tokenisation* donne priorité aux chaines les plus longues, l'énumération de chaines non séparatrices, comme la virgule immédiatement suivie d'un chiffre, permettra de neutraliser la définition explicite de la virgule comme séparateur.

En l'absence d'un branchement direct vers le statut du caractère, comme cela était possible auparavant par la construction d'un tableau indexé par le caractère, il faudra procéder à une fouille de la liste des définitions explicites des caractères simples et multiples énumérée par la déclaration *Alphabet* de SATO.

Les tables de tri à niveaux multiples, de même que l'existence d'une implantation en Perl de l'algorithme de tri Unicode nous ont incité à y faire appel afin de produire un tri alphabétique de grande précision. Il est à noter que la table de l'Unicode est un compromis pour l'ensemble des langues et doit être amendée pour satisfaire des habitudes de tri particulières.

L'utilisation de variables de caractères et de chaines de caractères de format 16 bits a exigé une révision minutieuse du code informatique de SATO. En effet, dans un langage typé comme PASCAL, le *transtypage* n'est généralement pas automatique ou, à tout le moins, requiert une évaluation détaillée des conséquences pour chacune des invocations de variables de type caractère ou chaine. De plus, certaines instructions deviennent carrément illégales. Par exemple, le langage permet des ensembles de caractères à 8 bits, mais interdit des ensembles de caractères à 16 bits. Il n'est donc plus possible de tester de façon directe l'appartenance d'un caractère à un ensemble impliquant Unicode. Dans ce cas, il faut modifier de façon significative l'implantation des algorithmes.

Tout ce travail de conversion a été réalisé avec succès. Certains traitement en ont cependant été ralentis, notamment en raison de particularités du compilateur. Certaines optimisations ont déjà été réalisées et d'autres pourraient être envisagées.

5.5.3 Des services Web au format XML-TEI

L'architecture client-serveur utilisée par SATO permet de fédérer des applications écrites dans des langages différents et provenant de diverses sources. Nous avons voulu pousser cette conception à son terme logique, à savoir la conception de *services Web*, c'est-à-dire des

applications autonomes pouvant résider sur des sites quelconques. Ces applications utilisent les protocoles du Web et des formats de données généralement en XML. C'est ainsi que nous avons défini un format de document XML-TEI, considéré comme un document d'annotation de corpus, pour l'analyse des cooccurrences. Cette expérimentation se situe au carrefour de nos orientations de recherche : architecture client-serveur (cf. 5.3), documents d'annotation, formalisation XML-TEI (cf. chapitre 6) et modèle documentaire (cf. 4.10b) permettant de gérer ces flux de documents dans le contexte d'un espace public d'échanges scientifiques. Il s'agit là d'un modèle de développement que nous entendons approfondir dans le futur. La notice qui suit rend compte de ce projet.



Un service Web pour l'analyse de la cooccurrence (5.5a, publication)

Dans une communication aux JADT 2010, nous présentions un exemple d'utilisation de service Web pour l'analyse de la cooccurrence. À la base de ce service, on retrouve une utilisation du format XML-TEI pour représenter tant les données que les résultats comme type particulier de document d'annotation externe (Martinez, Daoust, Duchastel 2010). Voici un extrait de la version longue de l'article que l'on pourra consulter en ligne : http://www.ling.uqam.ca/atonet/publications/martinez-daoust-duchastel2010.pdf .

Contexte informatique, documentaire et discursif

Dans son sens le plus général, on peut voir le service Web comme l'implémentation logicielle d'une ressource, identifiée par un *URI (Universal ressource Identifier)*, accessible en utilisant les protocoles Internet.

> Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions thanks to the use of XML. They can be combined in a loosely coupled way in order to achieve complex operations. Programs providing simple services can interact with each other in order to deliver sophisticated added-value

services. (http://www.w3.org/2002/ws/Activity)

Cette définition générale des services Web par le W3C met en évidence la notion d'interopérabilité d'applications logicielles indépendamment du langage utilisé pour les programmer, de l'architecture des calculateurs et des systèmes d'exploitation gérant ces calculateurs. Les requêtes, avec leurs données et leurs résultats, peuvent s'exprimer en utilisant diverses syntaxes concrètes, mais elles ont en commun de circuler à travers un réseau utilisant les standards du Web, en particulier l'URI (*Universal Ressource Identifier*) qui correspond à l'idée courante d'*adresse Internet*. Un des modèles d'implantation de services Web emprunte l'architecture REST (*Representational State Transfer*), d'après le terme inventé par Roy Fielding en 2000. Cette architecture fait largement appel au protocole HTTP couramment utilisé par les internautes dans leur navigation quotidienne. Elle a notamment pour particularité que chaque requête est *sans états,* dans le sens que la requête ne dépend pas d'un état antérieur de l'interaction et qu'elle repose uniquement sur les données transmises, lesquelles peuvent faire référence à des ressources existantes accessibles par URI. C'est ce modèle que nous avons employé dans le prototype fonctionnel que nous présentons ici.

La programmation d'applications sous forme de service Web n'implique pas qu'il faille toujours utiliser le Web pour accéder au service. Par exemple, au sein d'une grappe de calculateurs, un réseau interne à haut débit pourrait être utilisé pour fédérer des applications sans dépendre, au-delà du protocole de communication, d'environnements logiciels et matériels particuliers. On peut aussi faire tourner des services Web sur un même ordinateur qui *s'autoréférence* par son adresse IP locale. Donc, le service Web est d'abord une architecture de développement qui mise sur l'ouverture et qui peut se déployer à diverses échelles, y compris en mode autonome sur le poste de travail de l'utilisateur.

Les analyses statistiques de cooccurrence se prêtent bien à ce type d'architecture. D'abord, le modèle de calcul peut être défini de façon formelle et autonome en conformité avec l'architecture REST. On a l'habitude de présenter la sémantique de la cooccurrence statistique à partir de modèles probabilistes illustrés par des

exemples de tirages aléatoires de boules de différentes couleurs ou portant divers numéros. Dans le cadre de l'analyse textuelle, c'est souvent le mot qui sera pris comme l'équivalent de la *boule* et la *pige* du tirage prendra la forme d'empans textuels, par exemple la phrase, rassemblant un ensemble de mots. Pour rester dans des termes généraux, nous convenons de définir notre modèle de données comme étant constitué de deux ensembles : un ensemble d'objets, avec leur description, pouvant se retrouver dans un ensemble de contextes. Ce qui nous intéresse, c'est de savoir quels sont les objets qui, d'après un certain modèle probabiliste, occurrent ensemble dans les contextes avec une fréquence difficilement explicable par le hasard. Cette fréquence de *co-apparition* peut être beaucoup plus ou beaucoup moins élevée que prévue par le modèle probabiliste. Sur ce double ensemble de données, on peut donc procéder à plusieurs tests en faisant varier le modèle probabiliste, les paramètres du modèle, de même que le mot pôle par rapport auquel seront identifiés les objets dont la cooccurrence positive ou négative sera jugée significative en fonction du modèle statistique choisi.

Sur la base de ce modèle abstrait, on doit déterminer une syntaxe concrète pour représenter les deux ensembles de données, les paramètres de la requête et les résultats obtenus. En accord avec la proposition d'ATONET de format XML-TEI pour l'échange de corpus annotés (Daoust et Marcoux 2006), nous utiliserons les recommandations de la *Text Encoding Initiative* (TEI) pour représenter en XML les données de départ et l'enrichissement amené par les résultats de l'analyse de cooccurrence. Même si les données manipulées par le service Web sont principalement de nature numérique, le choix de considérer ces données comme faisant partie d'un texte est congruent avec notre proposition de modèle documentaire de dépôt de données adapté à la constitution de corpus de recherche (Daoust et coll. 2008). Ce modèle propose de publier l'annotation analytique sous la forme de documents numériques portant sur d'autres documents considérés comme primaires par rapport au document d'annotation. Dans son aspect documentaire, chacun des documents est une ressource possédant son URI. Les documents peuvent être décrits par des fiches de métadonnées pouvant être récoltées par des moteurs de métadonnées gérant le descriptif du document, à partir d'un noyau

Dublin Core. On peut aussi décrire et publier les relations entre documents sous la forme de relations RDF (*Resource Description Framework* ,W3C 2000) pouvant également faire l'objet de requêtes.

Cette mise en relation à l'échelle du document se superpose à une mise en relation beaucoup plus fine entre le document d'annotation et des éléments dans le document faisant l'objet de l'annotation. Pour établir ces relations, nous faisons appel aux structures de pointage recommandées par la TEI. L'utilisation d'éléments syntaxiques communs entre les documents *primaires,* sujets de l'annotation, et les documents *secondaires,* annotant les documents primaires, permet de rendre compte du mouvement réel de l'intertextualité dans lequel un texte commentant un autre texte pourra lui-même devenir objet de commentaires, d'analyses et d'annotations. Sous cet aspect, le document qui sera soumis à l'analyse de la cooccurrence demeure un texte sur un texte, même si son contenu emprunte davantage une forme numérique que prosaïque.

Le document de cooccurrence a donc, en quelque sorte, un double statut. Du point de vue du service Web calculant la cooccurrence, il est autonome et autosuffisant. Pour l'interprétation statistique de la cooccurrence, le document *secondaire* est nécessaire et suffisant. Il peut faire l'objet d'analyses successives, avec des algorithmes différents. Il peut être comparé à d'autres documents de cooccurrence établissant des relations entre *objets* sur la base de *contextes* différents pouvant même référer à d'autres textes. Ainsi, pourra-ton constater les différences entre réseaux de cooccurrents construits à partir de corpus différents.

D'un autre côté, la validation interprétative des résultats de la cooccurrence, du point de vue de leur portée discursive, exigera probablement un retour aux sources textuelles dans lesquelles les objets comptés prennent leur sens à l'intérieur de contextes inscrits dans une textualité concrète déployant de multiples réseaux de relations. Voilà pourquoi il est essentiel de prévoir, dans le document de cooccurrence, tous les mécanismes permettant de référer aux éléments du document primaire qui ont servi à construire le document de cooccurrence.

Syntaxe XML-TEI du document de cooccurrence

Pour concrétiser notre proposition, voici des extraits d'un document TEI soumis au service Web de cooccurrence. À sa suite, nous présentons les diverses composantes du document.

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="xsl/coocs-result.xsl"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt><title>Corpus constitutionnel, période 1941-1950 : dénombrement sur contextes et cooccurrences
</title></titleStmt>
<publicationStmt><p>Publié par le Centre ATO de l'UQAM</p></publicationStmt>
<sourceDesc> <p>La source primaire utilisée pour produire ce document de cooccurrence est le « Corpus
constitutionnel canadien (1941-1987) » produit par Gilles Bourque et Jules Duchastel et publié par la Chaire de
recherche du Canada en Mondialisation, Citoyenneté et Démocratie. http://www.chaire-mcd.uqam.ca/ato-mcd/
</p></sourceDesc>
</fileDesc>
<encodingDesc>
<fsdDecl> <fsdLink type="cooc" target="http://www.atonet.net/def/cooc_fsd.xml"/> </fsdDecl>
</encodingDesc>
</teiHeader>
<text>
<body>

<div type="Statistiques">
 <ab type="Sommaire">
   <measure type="Corpus_OccNbr" quantity="368672"/>
   <measure type="Contexte_OccNbr" quantity="123914"/>
   <measure type="Contexte_Nbr" quantity="4314"/>
   <measure type="Objet_Nbr" quantity="736"/>
  <measure quantity="2" type="Req_Nbr"/>
 </ab>
 <ab type="Cooccurrence" xml:id="req1">  <!-- req1-unité identifie la requête et les spécificités associées
-->
  <rs type="MéthodeStatistique" key="2"
ana="http://www.atonet.net/def/cooc.xml#binomiale">Binomiale</rs>
   <rs type="Pôle" key="obj425">unité</rs>
   <rs type="RéponseCumulative" key="2">Réponse cumulative</rs>
   <measure type="Seuil" quantity="0.05"/>
   <measure type="CoocNbr" quantity="20"/> <date>2009-11-11</date> <time>15:19</time>
 </ab>
 <ab type="Cooccurrence"  xml:id="req2"> <!-- req2-unité identifie la requête et les spécificités associées
-->
  <rs type="MéthodeStatistique" key="1"
ana="http://www.atonet.net/def/cooc.xml#spécificités">Spécificités</rs>
   <rs type="Pôle" key="obj425">unité</rs><!-- obj425 réfère à la structure de traits du pôle -->
   <rs type="RéponseCumulative" key="0">Réponse non cumulative</rs>
   <measure type="CoFréqMinimale" quantity="3"/>
   <measure type="SpécifMinimale" unit="exposant" quantity="2"/>
  <measure type="CoocNbr" quantity="20"/> <date>2009-11-15</date> <time>16:10</time>
 </ab>
</div>

<div type="fs">
<fs xml:id="obj1" type="Objet" n="Acte_de_l'_amérique_britannique_du_nord">
<f name="Numéro"><numeric value="1"/></f>
```

```
<f name="Effectif"><numeric value="71"/></f>
<f name="Contexte_Nbr"><numeric value="67"/></f>
<f name="Description"><string>ANALYSEUR COMPTAGE APPLIQUER FREQUENCES
$*socio~nil*freqtot&gt;19</string></f>
<f name="Id"><string>px428</string></f>
</fs>
<!-- ... -->
<fs xml:id="obj98" type="Objet" n="canadienne">
  <f name="Numéro"><numeric value="98"/></f>
  <f name="Effectif"><numeric value="92"/></f>
  <f name="Contexte_Nbr"><numeric value="90"/></f>
  <f name="Description"><string>ANALYSEUR COMPTAGE APPLIQUER FREQUENCES
$*socio~nil*freqtot&gt;19</string></f>
  <f name="Id"><string>px3034</string></f>
  <f name="Cooc" ana="#req1" n="bino">
    <numeric n="attendu" value="1"/> <numeric n="observé" value="10"/> <numeric n="prob"
value="0.00000001"/>
  </f>
  <f name="Cooc" ana="#req2"><numeric value="14"/></f>
</fs>
 <!-- ... -->
<fs xml:id="obj692" type="Objet" n="unité">
<f name="Numéro"><numeric value="692"/></f>
<f name="Effectif"><numeric value="43"/></f>
<f name="Contexte_Nbr"><numeric value="38"/></f>
<f name="Description"><string>ANALYSEUR COMPTAGE APPLIQUER FREQUENCES
$*socio~nil*freqtot&gt;19</string></f>
<f name="Id"><string>px16570</string></f>
</fs>
<!-- ... -->
</div>

<div type="Contexte" xml:base="discourscc.xml">
<span type="Dénombrement" from="#w2" to="#w32" xml:id="w2-w32" n="29">
<cb n="137"/>1<cb n="219"/>1<cb n="472"/>1<cb n="555"/>1<cb n="578"/>1
</span>
 <!-- ... -->
<span type="Dénombrement" from="#w135918" to="#w135945" xml:id="w135918-w135945" n="26">
<cb n="292"/>1<cb n="502"/>1<cb n="515"/>1<cb n="539"/>1
</span>
</div>
</body>
</text>
</TEI>
```

Cet exemple de document de cooccurrence a été généré par le logiciel SATO (Daoust, 2009) appliqué au *corpus constitutionnel canadien 1941-1987* (Bourque et Duchastel, 1996). Les contextes considérés dans l'exemple sont les phrases extraites de la partie du corpus correspondant à la période 1941-1950. Voici une description des diverses composantes du document de cooccurrence.

1. **xml-stylesheet**. Cette ligne provoque l'exécution d'une feuille de style XSLT qui produira une représentation HTML des résultats contenus dans

le document. Ainsi, si on ouvre cette ressource dans un navigateur Web à partir de son URI, on aura un affichage convivial. L'ensemble du fichier XML est cependant disponible et pourra être conservé dans son entièreté.

2. **teiheader**. Cette section correspond à l'entête standard de tout document TEI. On y retrouve une description du contenu du document avec ses références bibliographiques. Le contenu de l'élément *encodingDesc* fait référence au fichier de déclaration de structures de traits utilisé pour l'analyse de cooccurrence.

3. **div type="Statistiques"**. Après l'entête TEI, on retrouve trois divisions dans le corpus du texte. La division de type *Statistiques* contient deux blocs de données sous les balises *ab* (*arbitrary bloc*).

   Le bloc **ab type="Sommaire"** donne des informations quantitatives générales comme valeurs des attributs *quantity* des éléments *measure*. L'attribut *type* indique la nature de la mesure : taille du corpus source en nombre d'occurrences (*Corpus_OccNbr*) ; nombre d'occurrences dans les contextes considérés (*Contexte_OccNbr*) ; nombre de contextes considérés (*Contexte_Nbr*) et nombre d'objets qui sont dénombrés dans ces contextes (Objets_*Nbr*). On y trouve aussi le nombre de requêtes exécutées sur le fichier (Req_Nbr).

   Le bloc **ab type="Cooccurrence"** contient les données relatives à chacune des requêtes de cooccurrence. On aura autant de blocs de ce type qu'on aura soumis de requêtes au service Web de cooccurrence. Chaque bloc possède un identifiant unique (*xml:id="req1"*). Les éléments *rs* (*referring string*) et *measure* décrivent les paramètres de la requête.

4. **div type="fs"**. Cette division fait appel au formalisme des structures de traits pour décrire les propriétés (éléments **f** pour *feature*) de chacun des objets à dénombrer dans les contextes. Chaque objet est numéroté (*f name="Numéro"*) et l'ensemble de la structure *fs* reçoit un identifiant unique (attribut *xml:id*). Un autre trait (*f name="id"*) renvoie, si pertinent, à un identifiant dans le corpus primaire sur lequel a été construit le

document de cooccurrence.

Pour faciliter la lecture autonome du document, l'attribut *n* de l'élément *fs* contient un descriptif de l'objet, par exemple les formes lexicales *Acte_de_l'_amérique_britannique_du_nord*, *canadienne* et *unité*. Les traits *f name="Effectif"* et *f name="Contexte_Nbr"* indiquent le nombre total d'occurrences de l'objet dans les *Contexte_Nbr* contextes où il apparait. Finalement le trait *f name="Description"* indique comment cet objet a été obtenu à partir du corpus source. Dans ce cas-ci, on a utilisé la commande SATO *ANALYSEUR COMPTAGE APPLIQUER FREQUENCES $*socio~nil*freqtot>19*. On apprend ainsi que les objets sont toutes les formes lexicales ayant reçu une catégorie sociosémantique et dont la fréquence dans le corpus est supérieure à 19.

L'objet *canadienne* reçoit des traits supplémentaires qui ont été ajoutés par le service de cooccurrence. Par exemple, *f name="Cooc" ana="#req2"* avec un contenu numérique de 20. L'attribut *ana* indique que ce résultat a été produit par l'analyse de cooccurrence pointée par la valeur de l'attribut. Le contenu numérique du trait est l'exposant donnant la probabilité que cette cooccurrence soit due au hasard. Dans le cas de la binomiale (req1), le trait contient plusieurs valeurs donnant les détails du calcul comme le nombre de cooccurrents attendus selon le modèle probabiliste.

5. **div type="Contexte"**. Cette section du document est composée d'un ensemble d'éléments *span* qui définissent des empans textuels dans le document source *discourscc.xml*. Les attributs *from* et *to* du *span* pointent sur des identificateurs *xml:id* dans *discourscc.xml*. Ces identificateurs sont associés à des balises *w* qui découpent le corpus source en mots. L'attribut *n* donne la longueur de l'empan textuel dans le corpus de référence. Cette longueur se calcule en nombre d'occurrences. Le contenu du *span* est composé d'une suite de paires formées d'une balise vide <*cb/*> suivie d'un nombre. La balise *cb* marque une frontière de colonne dans une ligne de texte. On utilise l'attribut *n* pour indiquer le numéro de la colonne, ce qui permet d'omettre les frontières de colonnes pour lesquelles il n'y a pas de

contenu. Cette idée de colonne est une traduction directe de la représentation des données sous forme de tableau. Dans ce format, chaque ligne représente un contexte et chaque colonne le nombre d'occurrences de l'objet compté dans l'ordre séquentiel des objets identifiés dans la ligne d'entête. Ces tableaux sont des matrices creuses composées d'un très grand nombre de zéros. Dans le format XML, on assume qu'une colonne qui n'est pas décrite contient zéro comme valeur implicite. On peut donc omettre de la représentation XML la grande majorité des colonnes. On aurait aussi pu choisir de représenter le décompte des objets dans les contextes comme des séries de mesures balisées par des éléments *measure*. On aurait alors utilisé un attribut *ana* pour pointer sur la structure de traits de l'objet dénombré et un attribut *quantity* pour donner le résultat du comptage.

5.5.4 Vers un SATO en logiciel libre?

Même s'il nécessite des travaux de mise à jour pour le faire évoluer vers le paradigme de la *programmation orientée objet*, nous estimons que le code informatique de SATO est tout à fait en mesure de poursuivre son évolution, d'autant qu'il implémente des modèles originaux de traitement et de données. Ces modèles ont prouvé leur pertinence et sont pleinement compatibles avec l'annotation structurelle dont on discutera dans les chapitres suivants. Certes, toute implantation peut être remise en cause et reprise par un nouveau projet de développement qui miserait sur l'existence de librairies déjà éprouvées, par exemple dans l'espace du logiciel libre. Mais, encore faut-il savoir si les librairies nouvelles correspondent vraiment aux besoins et aux modèles de données et de traitement qu'elles sont censées remplacer. De plus, si ces choix impliquent la déqualification de l'expertise ancienne au simple profit de la *nouveauté*, il faudra être conscient qu'il s'agit là d'un choix social qui va bien au-delà d'un choix technique motivé seulement par des impératifs pratiques. En ce qui nous concerne, nous soutenons que le code informatique actuel pourra, à tout le moins, constituer une base valide pour la pérennisation du modèle SATO au-delà de l'action de son développeur. Cette évolution du code doit être considérée dans le contexte d'une intégration

du modèle documentaire et du principe élargi de l'annotation sous la forme d'annotation structurelle. Les problèmes principaux en rapport avec l'évolution du code informatique se posent à deux niveaux.

1. des problèmes techniques concernant les outils de programmation, le remplacement de certaines librairies désuètes et l'entretien d'un code documenté et bien conçu ;
2. des problèmes sociaux relatifs à l'organisation humaine du développement logiciel, de sa maintenance et de son support en termes d'utilisation.

SATO étant en opération sur nos serveurs, il est utilisé de façon régulière par un certain nombre d'usagers, notamment pour leurs travaux de thèse. Toute évolution du logiciel doit donc se faire sans interruption majeure de services et en assurant la compatibilité ascendante des formats de fichier. Ne disposant pas de ressources suffisantes pour tester les mises à jour du logiciel, ce sont souvent les usagers qui, dans leur application quotidienne, font ressortir les bogues inévitables. Cela plaide en faveur d'une évolution incrémentielle permettant de revenir rapidement à une version antérieure, ou, à tout le moins, de localiser rapidement la source des problèmes. Cette nécessité de maintien de service nous dicte donc une stratégie *étapiste* tendant à faire évoluer le code informatique de façon prudente.

Donc, sur le plan technique, divers aspects de l'évolution du code peuvent être envisagés de façon successive ou concurrentielle.

- Évolution dans le cadre du paradigme procédural visant à parfaire l'encapsulage des données et le cloisonnement entre les divers sous-systèmes logiques ;
- Documentation plus serrée des fonctions logicielles et remise en place du système de gestion des sources et des versions ;
- Utilisation accrue de l'architecture des services Web pour dégager certains analyseurs du code principal ;
- Analyse comparative des contraintes spécifique de la P.O.O. de Delphi par rapport à celles de d'autres langages, en particulier Java ; dans la mesure où ces contraintes sont compatibles, évolution du code Delphi vers sa couche objet ;
- Examen des possibilités de coexistence entre langages de P.O.O. afin d'utiliser les meilleures librairies dans chacun des langages ;

- Abandon éventuel de l'ensemble du code Pascal au profit d'un autre langage.

Sur le plan de l'organisation du développement, on a aussi différents modèles possibles articulés à différents contextes économiques et organisationnels. Le modèle actuel s'est établi dans le cadre de l'engagement personnel du développeur en synergie avec des chercheurs-utilisateurs. Si on juge que l'intérêt du logiciel dépasse cette situation conjoncturelle, il faudra trouver des cadres organisationnels qui permettent au projet de continuer d'exister et d'évoluer. Un des modèles pour prolonger l'existence d'un projet informatique, lorsque ses conditions initiales d'existence ne lui permet plus d'évoluer, est la formule du logiciel libre. Diverses formules juridiques peuvent encadrer cette formule. Mais leur caractéristique générale est d'empêcher une appropriation d'un code informatique par des groupes qui n'ont pas nécessairement intérêt à en soutenir l'utilisation la plus large.

La formule du logiciel libre ne signifie pas l'extinction des droits de propriété intellectuelle. Elle concerne plutôt une libération des droits de commercialisation et un appel à un développement collaboratif de la part de tous ceux qui trouvent intérêt à entretenir et à développer un logiciel utile. L'intérêt économique des développeurs ne tient donc pas à la vente du logiciel, mais à la vente de services de support et de formation, de même qu'à l'utilisation du logiciel dans le cadre de projets qui en nécessitent l'utilisation.

Dans la mesure où le code informatique de SATO pourrait évoluer vers un cadre qui en facilite la compréhension par de nouveaux développeurs, cette formule pourrait s'avérer intéressante pour faciliter la poursuite à long terme du développement. Il resterait à voir quelle forme organisationnelle serait la plus appropriée pour la gestion des projets de développement. Il pourrait s'agir d'un projet dont la responsabilité reposerait essentiellement sur un chef de projet, généralement le concepteur principal. Il pourrait s'agir d'une responsabilité relevant d'une institution existante, par exemple une ou plusieurs universités. Il pourrait s'agir d'une gestion par une entité juridique autonome, de type organisme sans but lucratif. En général, ce genre d'organisme doit se donner des règles de fonctionnement claires pour l'adhésion des membres et sa gestion, en particulier concernant les orientations de développement. La formule coopérative est aussi possible, mais probablement moins fréquente dans ce domaine.

D'autres voies sont aussi possibles. Le logiciel pourrait continuer à être développé par un réseau de collaborateurs universitaires sous la protection juridique des universités impliquées.

L'apport de partenaires privés est aussi possible dans la mesure où les intérêts impliqués seraient compatibles avec les orientations de recherche et de diffusion existantes.

Bibliographie du chapitre 5

Bradley et Rockwell, 1995. Bradley, J.; Rockwell, G. TACT and the WWW. *ACH/ALLC '95 Conference Abstracts*, University of California, Santa Barbara, July 11-15, 1995, p. 11-13.

Daoust, F. et Marcoux, Y. (2006). Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés. In *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp- 327-340, Presses universitaires de Franche-Comté, 2006. ISBN 2.84867130.0
http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/029.pdf

Delannoy, 2004. Delannoy, C. *Programmer en JAVA*, Eyrolles , Paris 2004. ISBN 2-212-11501-6.

Heiden, 2002. Heiden, S. *Weblex, Manuel Utilisateur, version 4.1 intermédiaire,* http://lexico.ens-lsh.fr/doc/weblex.pdf https://weblex.ens-lsh.fr/ sites visités le 7 janvier 2004.

ISO 24610-1, 2006. Language resource management -- Feature structures -- Part 1: Feature structure representation.

ISO/DIS 24611, 2008. Gestion des ressources linguistiques -- Cadre d'annotation morphosyntaxique.

Labonté, 1989. Labonté, A. The Canadian Alphanumeric Sort Order Standard for Z243.4, Preliminary Standard pz243.4.1, document de travail du Canadian Standards Association, File P111-23.

Labonté, 1988. Labonté, A. Fonctions de systèmes, Soutien des langues nationales. *Le curseur*, vol. 4, no. 3, juin 1988.

Martinez et coll. 2010. Martinez, W. ; Daoust, F. ; Duchastel, J. Un service Web pour l'analyse de la cooccurrence. *JADT 2010. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2010/allegati/JADT-2010-1079-1090_081-Martinez.pdf*

PHP, 1995. La première version de PHP, appelée PHP/FI, a été développée par Rasmus Lerdorf . http://www.php.net/

Rockwell et coll. 1997. Rockwell, G.; Passmore, G. ; Bradley, J. TACTweb: The Intersection of Text-Analysis and Hypertext, *Educational Computing Research*, vol. 17, no. 3, 1997, p. 217-230.

Söze-Duval Keyser, 2008. *Pour une textométrie opérationnelle*, http://www.cavi.univ-paris3.fr/lexicometrica/jadt/ressources-textometriques/textes/RTI6provisoire.pdf

Rockwell et coll., 2002. Rockwell et coll. *TAPoR: Text-Analysis POrtal for Research.* http://huco.ualberta.ca/Tapor/ site visité le 7 janvier 2004.

TACT-Web. http://tactweb.humanities.mcmaster.ca/

TEI P5. *Guidelines for Electronic Text Encoding and Interchange*, chapitre 5 *Feature Structures.* http://www.tei-c.org/Guidelines/P5/

6 L'annotation structurelle

6.1 problématique

Dans la tradition de l'analyse de texte par ordinateur, l'annotation et la catégorisation font partie des opérations permettant d'enrichir les données textuelles au fur et à mesure de leur analyse, éclairée par des outils statistiques et divers modes de lecture comparative. En général cependant, les unités ainsi annotées sont des occurrences individuelles, des unités de contexte ou des formes lexicales affublées de propriétés, attributs ou structures de traits. Mais, la structuration de ces unités et leur mise en relation sont plus rarement abordées. C'est cette dimension, que nous appelons l'annotation structurelle, que nous présenterons ici sous la forme de propositions de format de document externe d'annotation. Nous poursuivrons par une exemplification de ces propositions sur des problèmes de linguistique textuelle.



L'annotation structurelle (6.1a, publication, extrait)

Dans une communication aux JADT 2010, nous présentions ainsi la notion d'annotation structurelle (Daoust, Marcoux, Viprey 2010).

Nous désignons, par *annotation structurelle,* l'ajout à des ressources textuelles existantes d'annotations analytiques visant la mise en relation de segments textuels explicitant le fonctionnement de la langue, du discours et de la mise en texte. Ces mises en relation sont des pratiques de base de l'analyse textuelle dans sa tradition scolaire. Sur un plan plus formel, l'analyse syntaxique est la forme la plus connue de l'annotation structurelle avec ses forêts d'arbres qui annotent les divers composants de la proposition et de la phrase. Au-delà de la phrase, la linguistique textuelle, dans la foulée de Bakhtine (Bakhtine 1984), perçoit le texte comme un *réseau de déterminations*.

La linguistique textuelle a pour rôle, au sein de l'analyse de discours, de

> théoriser et de décrire les agencements d'énoncés élémentaires au sein de l'unité de haute complexité que constitue un texte. Elle a pour tâche de détailler les « relations d'interdépendance » qui font d'un texte un « réseau de déterminations » (Weinrich 1973 : 174). La linguistique textuelle porte autant sur la description et la définition des différentes unités que sur les opérations dont, à tous les niveaux de complexité, les énoncés portent la trace. (Adam 2005:33)

Malgré que l'analyse textuelle en général et la linguistique textuelle en particulier fassent grand état des multiples structures qui traversent le texte, la tradition de l'analyse statistique des données textuelles y a fait peu de place. Certes, plusieurs chercheurs ont situé leur travaux aux confins de l'analyse syntaxique, telle que pratiquée en traitement automatique de la langue, et de l'analyse de discours à tradition lexicométrique (voir, entre autres Habert 1998). Mais, ces travaux sont généralement limités à la prise en compte des syntagmes nominaux dans l'analyse contrastive des énoncés. Les connexions du texte et du discours, en tant qu'*unités structuralement ouvertes* (Charolles 1993:311, cité par Adam 2005:36), sont rarement prises en compte.

Même si ces connexions peuvent partager le même formalisme d'annotation que les relations syntaxiques, leur nature est très différente. Adam souligne que, dès qu'on dépasse le seuil de la phrase, ce ne sont plus les *solidarités syntaxiques* qui prévalent mais plutôt « des marques et des instructions relationnelles de portée plus ou moins lointaine » (Adam 2005:36). S'appuyant sur Charolles, Adam introduit l'idée de *marques instructionnelles* qui signalent au destinataire que « telle unité doit être comprise comme entretenant telle relation avec telle ou telle autre » (Charolles 1993:311, cité par Adam 2005:36).

Dans la tradition de l'analyse statistique des données textuelles, on marque habituellement les parties du corpus. Il s'agit généralement de balisage de la structure formelle du corpus en termes de documents, de tours de parole, de locuteurs, de paragraphes. etc. Ainsi, par exemple, l'analyse factorielle des correspondances pourra, sur la base de l'analyse des fréquences lexicales de

chacune des parties marquées, produire une synthèse des données contrastant simultanément les profils lexicaux et les parties du corpus. Mais, ces divisions simples entre parties demeurent un pâle reflet des relations structurales entre segments textuels.

Dans la tradition de l'analyse de texte par ordinateur (ATO), certains logiciels, par exemple SATO (Daoust, 2009) permettent d'annoter, en cours d'analyse, les unités lexicales, les occurrences et les segments afin de rendre compte d'une variété de paradigmes catégoriels. Il reste qu'il s'agit d'une annotation *à plat* qui ne peut marquer la relation que par héritage sur les unités terminales. Ainsi, par exemple, pour marquer la relation dialogique entre locuteurs, on pourra avoir une propriété indiquant qui est l'énonciateur et une autre indiquant à qui il s'adresse. La conjonction des deux permettra de configurer dynamiquement les parties du texte et du lexique à soumettre aux analyseurs statistiques. L'annotation structurelle vise à aller au-delà de cette annotation simple, à structure implicite, en marquant sous forme de multiples graphes les connexions induites par les *marques instructionnelles* dont parle Adam. En conjonction avec le filtrage des annotations simples, le parcours des graphes permettra de contraster beaucoup plus aisément les segments textuels en fonction de leurs positions dans l'une ou l'autre des annotations structurelles.

Dans la tradition de l'ATO, la catégorisation, dans sa dimension lexicale (forme en tant que classe) et textuelle (occurrence de la forme en contexte), permet de soumettre à l'analyse statistique des fréquences de catégories marquant des résultats d'analyse et d'interprétation susceptibles, par exemple, de rendre compte d'éléments de la structure syntaxique ou sémantique de l'énoncé. L'annotation structurelle permet aussi de compter des *configurations*, c'est-à-dire des *motifs structurels* à l'intérieur de certains empans déterminés par des structures plus amples, par exemple, telle structure argumentative dans tel type d'épisode narratif.

L'intérêt de l'annotation structurelle ne se limite pas, bien entendu, à la qualification des unités soumises au calcul statistique. Comme les concordances, par exemple, elle est un outil de navigation permettant des parcours hypertextuels appuyant

l'interprétation sur l'explicitation des connexions qui tissent le discours et le texte. Cette navigation doit aller dans les deux sens : de la localité, l'occurrence, vers les structures et les éléments qu'elles connectent, d'une part et, d'autre part, de la structure, par exemple le plan du texte, vers ses parties constituantes. Ces parcours sont l'extension de notre pratique actuelle qui nous plonge du contexte au lexique, du lexique au contexte, une extension aussi des parcours des réseaux de cooccurrents et des réseaux lexicaux.

Ce premier type de considérations, justifiant notre proposition d'annotation structurelle, est complété par des considérations de type davantage documentaires. La mise en connexion n'est pas seulement intratextuelle : elle est aussi intertextuelle. Les textes font référence les uns aux autres, directement ou par le partage de mêmes paradigmes. Plus encore, l'analyse textuelle, en tant qu'elle-même pratique discursive, produit des textes sur des textes, des annotations sur des textes, y compris des textes d'annotation et d'analyse. Notre entreprise de modélisation doit donc aussi comporter une dimension documentaire permettant de mettre en relation les textes qui circulent dans l'espace public et autour desquels s'articule le discours social. Voilà pourquoi, du point de vue de son inscription concrète dans l'espace public, nous proposons que l'annotation analytique, commentaires, catégories ou graphes, prenne la forme de documents d'annotation XML respectant une syntaxe conforme aux recommandations de la Text Encoding Initiative (TEI). Ces documents pourront ainsi s'intégrer plus aisément au modèle de dépôt de données adapté à la constitution de corpus de recherche (Daoust et coll. 2008). Ces systèmes de dépôt de données, surtout connus pour la diffusion des publications scientifiques, peuvent être étendus aux résultats et procédures d'analyse au-delà de leur synthèse dans les articles scientifiques.

6.2 Linguistique textuelle et TEI

Dans cette section, nous voulons introduire des hypothèses de représentation TEI pour rendre compte des théories d'analyse textuelle présentées dans le livre de Jean-Michel Adam sur

l'analyse textuelle des discours (Adam 2005). On examinera d'abord des hypothèses de balisage TEI rendant compte de la perspective fonctionnelle de la phrase. Ensuite, on verra qu'on peut utiliser ces formalismes TEI pour rendre compte des diverses dimensions d'analyse que l'on retrouve illustrées dans l'analyse d'un court texte de Borges développée par Adam en conclusion de son ouvrage. Ces dimensions vont d'une analyse fine des énoncés jusqu'aux commentaires libres mettant en relation plusieurs ouvrages de l'auteur.

Les propositions développées dans les pages qui suivent s'inscrivent dans un modèle d'annotation externe conforme au cadre documentaire présenté notamment dans une communication aux JADT 2008 : *Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche* (Daoust et coll. 2008).



Document d'annotation (6.2a, définition)

Dans notre communication aux JADT 2010, nous résumions ainsi la notion de document d'annotation et l'utilisation des propositions de Sacacomie pour l'annotation structurelle (Daoust, Marcoux, Viprey 2010).

Un document d'annotation est une ressource électronique possédant un identifiant unique, au sens du W3C, et qui utilise des mécanismes de pointage permettant de faire référence à des parties d'un ou de plusieurs autres documents numériques aussi localisables par les mécanismes standards du Web (URI et URL). On utilise le terme d'annotation dans son sens le plus large comprenant aussi le simple fait de commenter et de citer une ressource. On peut qualifier les documents d'annotation de *secondaires* par rapport aux documents annotés que l'on pourrait qualifier de *primaires*. Bien sûr, un document, considéré à une étape donnée comme *secondaire*, deviendra *primaire* par rapport à un autre document *secondaire* qui l'annoterait.

Le langage de balisage XML est maintenant l'approche privilégiée pour constituer des documents structurés ou semi-structurés en offrant une syntaxe unique et extensible selon des principes bien définis. La *Text Encoding Initiative* (TEI) est ce consortium qui se consacre depuis 1987 à formuler des propositions pour l'encodage des textes en format numérique pour la communauté des sciences humaines. Depuis

leur version P4, les propositions de la TEI sont exprimées dans une syntaxe XML.

L'adoption des recommandations de la TEI par un grand nombre d'organismes dans le monde nous a incités, tout naturellement, à nous référer à ces recommandations pour proposer des formats XML-TEI pour l'échange de corpus et de ressources textuelles au sein des communautés qui gravitent autour des JADT. C'est ainsi que le réseau ATONET (2005) a proposé un sous-ensemble de balises TEI pour traduire, à des fins d'échange, les formats propriétaires utilisés par les logiciels d'analyse textuelle couramment employés au sein de la communauté de la recherche. C'est, ce que nous avons appelé les *propositions de Sacacomie* (Daoust et Marcoux 2006), du nom du lieu où s'est tenu le séminaire présentant ces propositions.

Les *propositions de Sacacomie* comprennent un encodage dit *embarqué* (*embedded* en anglais) des annotations simples. Cela signifie que les annotations peuvent s'inscrire dans le document primaire selon la pratique de la majorité des logiciels considérés par le groupe de travail d'ATONET : ALCESTE (Reinert, 2002), Diatag-Astartex (Viprey, 2005), DTM (Lebart, 2005), Lexico (Salem et coll., 2003) et SATO (Daoust. 2009). En fait, nous formulions à l'époque deux propositions : une *proposition de base* servant de commun dénominateur aux logiciels existants et une *proposition avancée* comprenant un découpage en mots marqué par la paire de balises *<w> </w>*. L'élément *w* est accompagné d'un attribut *xml:id* identifiant chacun des mots de manière unique. Cette proposition comprenait aussi le principe du document d'annotation externe utilisant les structures de traits (avec leur élément *fs « feature structure»*) pour annoter les formes lexicales et leurs occurrences. Notre proposition de format pour l'annotation structurelle s'appuie sur cette *proposition avancée de Sacacomie.* Elle reprend l'utilisation de l'élément *span* suggéré par la TEI pour référer, dans le document secondaire d'annotation, à un empan textuel dans le document primaire annoté.

Cet élément *span* est présenté dans le chapitre intitulé *Simple Analytic Mechanisms* du *TEI P5: Guidelines* (TEI Consortium 2007). Il y est décrit comme un des mécanismes simples de référence à des empans textuels utilisés à des fins analytiques. Il permet d'associer une annotation interprétative à un passage de texte

référé par des pointeurs. Les <*span*> peuvent être coiffés d'un élément <*spanGrp*>, comme illustré dans l'exemple suivant.

```
<spanGrp resp="#Adam2005" type="ThèmeRhème" xml:base="http://monsite.org/doc-source.xml">
<span from="#w1" to "#w4" xml:id="Th1" ana="#thème"> Thème initial en début de phrase ( «Et un jour » ) </span>
</spanGrp>
```

La balise <*spanGrp resp="#Adam2005" type="ThèmeRhème"*> permet de factoriser des attributs communs à un ensemble de <*span*> : *resp* renvoie à la description, généralement dans l'entête TEI, de la personne responsable de cette annotation, alors que *type* indique de quel type d'annotation qu'il s'agit. L'attribut *xml:base* contient l'URL du document analysé. Dans l'exemple, il s'agit du nom du document *doc-source.xml* sur *monsite.org*. On assume ici que ce document contient le texte à analyser découpé en mots identifiés par l'attribut *xml:id* des éléments <*w*>.

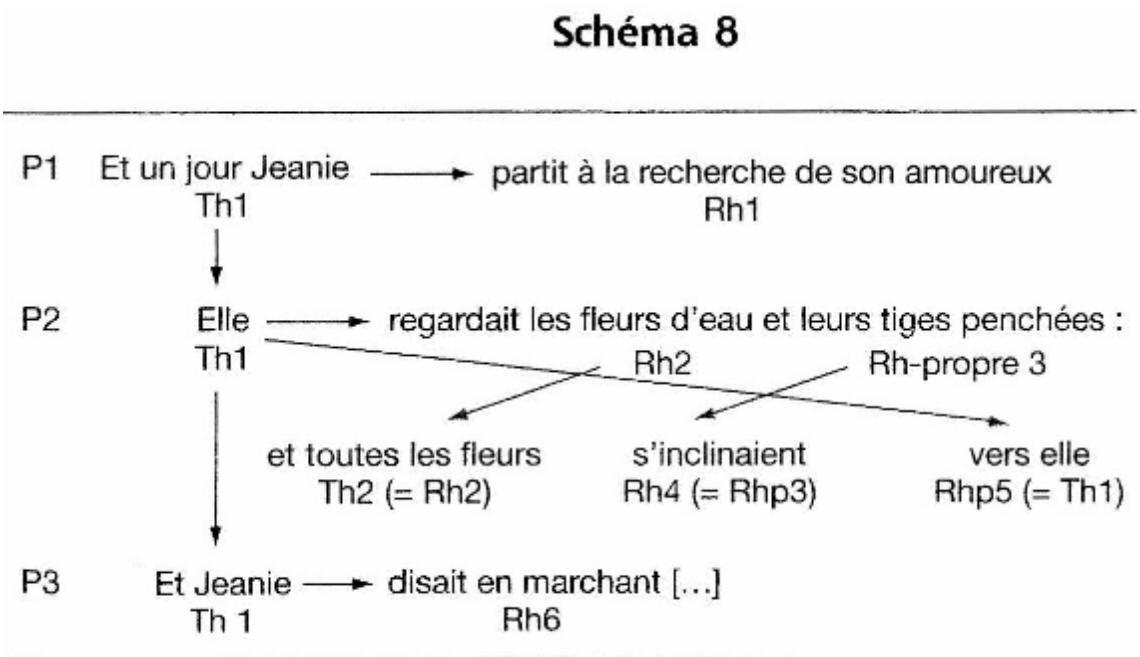
Le contenu de la balise <*span*> est utilisé pour délimiter un passage et expliquer la nature de l'annotation. Les attributs *from* et *to* contiennent un pointeur sur le début et la fin du passage sur lequel porte l'annotation. L'attribut *to* est facultatif si la fin coïncide avec le début du passage. Dans l'exemple, *w1* et *w4* renvoient aux valeurs de l'attribut *xml:id* des éléments <*w*> dans le document primaire *doc1.xml*. L'attribut *ana* pointe sur une interprétation de l'élément. Il est courant d'inscrire cette interprétation dans un élément <*interp*>. Les recommandations de la TEI indiquent que cet élément <*interp*> vise à résumer l'interprétation d'une annotation analytique. L'élément <*interp*> peut faire partie d'un <*interpGrp*> qui permet aussi de factoriser des attributs communs à un ensemble de balises <*interp*>. Ici, on fait appel à la combinaison des éléments <*span*> et <*interp*> pour distinguer le schéma général de l'analyse, avec la définition des concepts, de l'instanciation du concept sur un passage donné. La TEI signale qu'on pourrait aussi utiliser des structures de traits, plutôt que des éléments <*interp*>. Les structures de traits sont particulièrement appropriées lorsque l'analyse renvoie à des systèmes catégoriels.

6.2.1. Perspective fonctionnelle de la phrase : la relation thème-rhème.

6.2.1.1 Présentation de l'exemple

Dans les paragraphes qui suivent, on discutera de diverses propositions TEI aptes à traduire l'exemple du schéma 8 (Adam 2005:49) illustrant l'application de la *perspective fonctionnelle de la phrase* sur une courte phrase. Voici la phrase et le schéma.

Et un jour Jeanie partit à la recherche de son amoureux. Elle regardait les fleurs d' eau et leurs tiges penchées : et toutes les fleurs s' inclinaient vers elle. Et Jeanie disait en marchant …



Cet extrait pourrait faire partie d'un balisage TEI imbriqué dans le document texte lui-même. Cependant, conformément à nos choix méthodologiques, nous ferons appel, pour la représentation des structures fonctionnelles du schéma 8, à une annotation externe s'appuyant sur un balisage du document source qui permet d'identifier chacun des mots (*occurrence, token* en anglais) par un élément *w* (cf. proposition *Sacacomie avancée* d'*ATONET* : Daoust, Marcoux 2006). L'élément est accompagné de l'attribut *xml:id*, un attribut standard de l'espace de nom *xml*. La valeur de cet attribut est un identifiant quelconque, mais qui se doit d'être unique à l'intérieur du document. Le fichier *doc1.xml*, donné en exemple, est un document TEI complet avec un entête simplifié, l'emphase étant mise ici sur le corps du texte (*body*) avec les balises *w* et leur contenu.



Marquage des occurrences par la balise *w* (document *doc1.xml*) (6.2.1a, exemple)

<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
 <**teiHeader**>
  <fileDesc>
   <titleStmt>
    <title>Texte utilisé pour exemplifier une analyse fonctionnelle de type thème-rhème (Phébus, 1884, 2002 :429) : version électronique</title>
    <respStmt> <resp>mis en forme par</resp> <name>François Daoust</name> </respStmt>
  </titleStmt>
  <publicationStmt>
   <distributor>Université du Québec à Montréal, Centre ATO</distributor>
   <pubPlace>Québec, Canada</pubPlace>
   <date>2008-02-05</date>
  </publicationStmt>
  <notesStmt>
   <note>Des annotations analytiques sur le texte figurent dans des fichiers séparés.</note>
  </notesStmt>
  <sourceDesc>
  <bibl> Adam, Jean-Michel. La linguistique textuelle, Introduction à l'analyse textuelle des discours. Page 49. Armand Colin, Paris 2005, ISBN 2-200-26752-5.</bibl>
  </sourceDesc>
  </fileDesc>
  <profileDesc> <langUsage> <language ident="fr">Français</language> </langUsage> </profileDesc>
  <encodingDesc>
   <refsDecl>
     Les références de pagination utilisent les balises pb (début de page), lb(début de ligne) et w (mot dans la ligne).
   </refsDecl>
   <editorialDecl>
    <segmentation>
     La balise w est utilisée pour segmenter le texte en mots.
    </segmentation>
   </editorialDecl>
  </encodingDesc>
 </teiHeader>

 <text>
  <**body**>
<pb n="49"/>
 <lb/><w xml:id="w1">**Et**</w> <w xml:id="w2">**un**</w> <w xml:id="w3">**jour**</w> <w xml:id="w4">**Jeanie**</w> <w xml:id="w5">**partit**</w> <w xml:id="w6">**à**</w> <w xml:id="w7">**la**</w> <w xml:id="w8">**recherche**</w> <w xml:id="w9">**de**</w><w xml:id="w10">**son**</w><w xml:id="w11">**amoureux**</w><w xml:id="w12">**.**</w>
<w xml:id="w13">**Elle**</w> <w xml:id="w14">**regardait**</w> <w xml:id="w15">**les**</w> <w xml:id="w16">**fleurs**</w> <lb/><w xml:id="w17">**d'**</w><w xml:id="w18">**eau**</w> <w xml:id="w19">**et**</w> <w xml:id="w20">**leurs**</w> <w xml:id="w21">**tiges**</w> <w xml:id="w22">**penchées**</w> <w xml:id="w23">**:**</w> <w xml:id="w24">**et**</w> <w xml:id="w25">**toutes**</w> <w xml:id="w26">**les**</w> <w xml:id="w27">**fleurs**</w> <w xml:id="w28">**s'**</w><w xml:id="w29">**inclinaient**</w> <w xml:id="w30">**vers**</w> <w xml:id="w31">**elle**</w><w xml:id="w32">**.**</w>
<w xml:id="w33">**Et**</w> <w xml:id="w34">**Jeanie**</w>
<lb/><w xml:id="w35">**disait**</w> <w xml:id="w36">**en**</w> <w xml:id="w37">**marchant**</w>

```
<!-- etc. -->
</p>
  </body>
 </text>
</TEI>
```

6.2.1.2 Identification des segments textuels.

On désigne ici par *segment textuel* une suite continue de mots. Plusieurs éléments XML définis par la TEI peuvent être utilisés pour identifier de tels segments. Cependant, nous ne considérerons ici que les éléments pouvant être utilisés dans un document d'annotation externe au document annoté (*Stand-off Markup*), distinguant ainsi formellement *document source* (*doc1.xml* dans l'exemple) et *document d'annotation* (*ana1.xml* dans l'exemple) portant sur le document analysé. Pour simplifier la présentation, nous assumons que les fichiers *doc1.xml* et *ana1.xml* sont localisés au même endroit, ce qui nous évitera l'emploi d'URL complets dans les attributs de type pointeur.

Dans ses recommandations, la TEI aborde la référence à des empans textuels dans deux chapitres distincts. Le chapitre 16 (*Linking, Segmentation, and Alignment*) présente une série de dispositifs spécialisés de pointage. Pour sa part, le chapitre 17 (*Simple Analytic Mechanisms*) présente des mécanismes simples de référence à des empans textuels utilisés à des fins analytiques. Il s'agit de l'élément <*span*>, qui associe directement une annotation interprétative à un passage de texte, et de l'élément <*spanGrp*> qui permet de rassembler un ensemble de <*span*> et de factoriser des attributs communs. Par exemple,

```
<spanGrp resp="#Adam2005" type="ThèmeRhème" xml:base="http://monsite.com/doc1.xml">
<span from="#w1" to "#w4" xml:id="Th1" ana="#thème"> Thème initial en début de phrase ( «Et un jour » )
</span>
</spanGrp>
```

La balise <*spanGrp resp="#Adam2005" type="ThèmeRhème"*> permet de factoriser des attributs communs à un ensemble de <*span*> *: resp* renvoie à la description, généralement dans l'entête TEI, de la personne responsable de l'annotation alors que *type* indique de quel type d'annotation qu'il s'agit. L'attribut *xml:base* contient l'URI du document analysé. Dans l'exemple, il s'agit du nom du document *doc1.xml* sur *monsite.com*. On assume ici que ce document contient le texte à analyser découpé en mots identifiés par l'attribut *xml:id* des éléments <*w*>.

Le contenu de la balise <*span*>, décrit en 6.2a, est utilisé pour expliquer la nature de l'annotation d'un empan textuel.

Les recommandations de la TEI ne semblent pas donner d'indications sur le traitement des attributs *from* et *to* lorsqu'ils pointent sur un élément qui contient lui-même des pointeurs, un autre <*span*> par exemple. Mais, on comprendra que l'évaluation récursive des pointeurs permettra finalement d'atteindre la réalisation textuelle du segment annoté.

La balise <*span*> fournit un cadre pratique pour une annotation simple. On doit en moduler l'usage pour une annotation structurelle comme l'analyse fonctionnelle de la phrase qui peut conférer divers rôles à un même segment en fonction de son rapport avec d'autres segments. On pourrait donc utiliser plusieurs éléments <*span*> désignant un même segment mais dans des contextes interprétatifs différents.

Par ailleurs, dans le chapitre des *Recommandations* sur *le liage, la segmentation et l'alignement* des recommandations de la TEI, on retrouve un élément <*ref*> auquel on pourrait faire jouer un rôle similaire au <*span*>. En voici un exemple.

<ref type="ThèmeRhème" target="range(doc1.xml#w1,doc1.xml#w4)" xml:id="Th1" ana="#thème">Thème initial en début de phrase («Et un jour »)</ref>

L'attribut *target* remplace la paire *from to* en faisant appel à un *schéma d'adressage.* Ici, le schéma TEI *range* indique que l'empan textuel s'étend de l'élément identifié par *w1* dans *doc1.xml* à l'élément *w4* de *doc1.xml.* Les identificateurs utilisés dans la fonction *range* de l'expression *Xpath* renvoient aux valeurs de l'attribut *xml:id* des éléments <*w*> dans le document *doc1.xml.* Le contenu des éléments <*w*> est la chaine de caractères du mot.

Le contenu de l'élément <*ref*> fournit une explication sur le texte référé. L'attribut *xml:id* est utilisé pour inscrire l'identifiant unique utilisé par Adam pour référer à ce segment thématisé.

L'attribut *ana* fait partie des attributs communs à tous les éléments TEI. On l'utilise donc, comme pour l'élément <*span*>, afin de marquer une interprétation analytique. Le contenu de l'attribut pointe sur un ou plusieurs éléments <*interp*> *ou* <*fs*>.

Du point de vue syntaxique, un élément <*span*>, avec ses attributs *from* et *to*, ne pointe que sur un empan unique alors que <*ref*> permet de pointer sur plusieurs objets. En effet, l'attribut *target* de l'élément <*ref*> offre une jointure implicite puisque sa valeur consiste en un ou plusieurs *URI* et dispositifs de pointage (*range* par exemple) séparés par des espaces. Ce

mécanisme est puissant, mais plus exigeant du point de vue de l'implantation informatique qui doit opérer une analyse de la valeur de l'attribut *target.*

Le chapitre 16 (*Linking, Segmentation, and Alignment*) des recommandations de la TEI propose aussi des balises explicites permettant de combiner des segments de diverses façons. Ils font tous appel à des attributs à valeurs multiples. Un premier dispositif consiste à utiliser un *pointeur intermédiaire*, pour reprendre les termes de la TEI (Recommandations section 16.1.4). Il s'agit simplement d'un élément <*ptr*> dont l'attribut *target* est identique à celui de l'élément <*ref*>. On peut aussi utiliser un élément <*join*> marquant, de façon explicite, la jointure de deux empans. Cet élément fait partie de la même classe que l'élément <*alt*> qui permet de décrire une disjonction de segments. Ici encore, le dispositif de pointage passe par un attribut *targets* (avec un *s* cette fois) qui contient une suite de pointeurs séparés par des espaces. Donc, dans des cas plus complexes de références à des segments, il est possible de faire appel à des dispositifs plus élaborés que les <*span*>.

Cela dit, pour illustrer l'annotation analytique de la linguistique textuelle, il suffira le plus souvent de faire appel aux dispositifs les plus simples. Dans les paragraphes qui suivent, nous allons donc privilégier l'élément <*span*> en ne recourant aux dispositifs plus complexes de pointage qu'en cas de nécessité.

6.2.1.3 Annotation structurelle des relations thèmes-rhèmes et de la progression thématique

Dans cette section, nous explorons diverses manières de noter en XML-TEI les structures *thèmes-rhèmes* introduites en début de chapitre.

Une première façon de faire consisterait à décrire une succession d'arbres dont les feuilles pointeraient sur des segments textuels du document analysé.

Les segments impliqués dans le schéma 8 de Jean-Michel Adam sont les suivants :

```
<spanGrp type="Segmentation" ana="#Énoncé" xml:base="doc1.xml">
<span from="#w"1   to="#w4"   xml:id="s1-4">Et un jour Jeanie</span>
<span from="#w5"   to="#w12" xml:id="s5-12"> partit à la recherche de son amoureux. </span>
<span from="#w13" to="#w13" xml:id="s13-13">Elle</span>
<span from="#w14" to="#w18" xml:id="s14-18">regardait les fleurs d'eau</span>
<span from="#w19" to="#w23" xml:id="s19-23">et leurs tiges penchées:</span>
<span from="#w24" to="#w27" xml:id="s24-27">et toutes les fleurs</span>
<span from="#w28" to="#w29" xml:id="s28-29">s'inclinaient</span>
<span from="#w30" to="#w32" xml:id="s30-32">vers elle.</span>
<span from="#w33" to="#w34" xml:id="s33-34">Et Jeanie</span>
<span from="#w35" to="#w37" xml:id="s35-37">disait en marchant</span>
</spanGrp>
```

L'élément <*SpanGrp*> permet d'identifier les attributs communs à un ensemble d'empans textuels construits sur le document externe *doc1.xml*. L'attribut *type* avec la valeur *Segmentation* nous indique qu'il s'agit simplement d'une énumération de segments de texte. Ce découpage est déjà une analyse basée ici sur l'idée d'*énoncé*. L'attribut *ana* pointera, comme on le verra dans la suite du texte, sur un élément <*interp xml:id="Énoncé"*> définissant ce que l'on entend par énoncé.

On pourrait choisir d'ajouter aux balises du TEI un nouvel élément décrivant spécifiquement la structure thème-rhème. Par exemple :

```
<ThemeRheme>
<span type="thème" xml:id="Th1" from="#s1-4"/>
<span type="rhème" xml:id="Rh1" from="#s5-12"/>
</ThemeRheme>
```

Dans cet exemple, la valeur de l'attribut *to* étant identique à celle de l'attribut *from*, l'attribut *to* pouvait être omis. L'élément pointé étant lui-même un <*span*>, l'empan référé résulte de l'évaluation des pointeurs de ce deuxième <*span*>. L'attribut *type* indique la nature de l'empan textuel. Le choix de *Th1* à titre d'identifiant unique de l'empan textuel n'est cependant pas approprié. Le schéma d'Adam montre qu'en fait, il n'est pas unique. *Th1* est une étiquette pouvant s'appliquer à plusieurs empans textuels ayant un même référent dans le monde. On utilisera donc plutôt les constructions suivantes dans lesquelles ces étiquettes catégorielles seront considérées à titre de contenu explicatif des éléments <*span*>. Aussi, nous allons substituer un attribut *ana* à l'attribut *type* pour bien indiquer que les étiquettes *Thème* et *Rhème* correspondent à des choix analytiques sur un empan, plutôt qu'à une détermination d'un type particulier d'empan. Ces étiquettes seront utilisées comme des pointeurs sur des éléments explicatifs ou définitoires.

```
<ThemeRheme xml:base="doc1.xml">
<span ana="#Thème" xml:id="T1-4" from="#s1-4>Th1</span>
<span ana="#Rhème" xml:id="R5-12" from="#s5-12">Rh1</span>
</ThemeRheme>

<ThemeRheme xml:base="doc1.xml">
<span ana="#Thème" xml:id="T13-13" from="#s13-13">Th1</span>
<span ana="#Rhème" xml:id="R14-18" from="#s14-18">Rh2</span>
<span ana="#Rhème" xml:id="R19-23" from="#s19-23">Rhp3</span>
</ThemeRheme>

<ThemeRheme xml:base="doc1.xml">
<span ana="#Thème" xml:id="T13-13" from="#s13-13">Th1</span>
<span ana="#Rhème" xml:id="R14-18" from="#s14-18">Rh2</span>
```

Rhp3
</ThemeRheme>

<ThemeRheme xml:base="doc1.xml">
Th1
Rh6
</ThemeRheme>

Comme on le voit, *Th1* revient dans trois structures différentes avec des segments qui ne pointent pas sur les mêmes mots. Ces segments s'inscrivent dans une *continuité référentielle* (Adam 2005:86) non explicitée. Le segment T13-13 (*Elle*) est une anaphore pronominale du segment T1-4 (*Et un jour Jeanie*) alors que le segment T33-34 (*Et Jeanie*) est une reprise explicite du nom propre *Jeanie*. La coréférence est aussi présente avec *Th2* (*et toutes ces fleurs*) qui est une reprise de *Rh2* (*fleurs d'eau*). La continuité référentielle se manifeste aussi à travers le liage sémantique entre *penchées* et *s'inclinaient* des segments *Rhp3* et *Rh4* et par la reprise du pronom *elle* entre *Rhp5* et *Th1*. C'est à travers ce réseau référentiel que se construit la *progression thématique.*

L'ajout d'éléments nouveaux (*ThemeRheme* dans l'exemple) pour décrire de nouvelles dimensions d'analyse est discutable. En effet, ce que l'on gagne en lisibilité pour la lecture humaine, on le perd au niveau du traitement informatique qui profiterait du caractère générique du formalisme d'annotation pour effectuer directement des calculs statistiques, des analyses de graphes ou de traitement automatique de la langue à partir de ces éléments génériques.

Or, la TEI fournit des éléments généraux, agissant comme *conteneurs,* dont la portée sémantique peut être précisée par des attributs. C'est le cas en particulier des élément <*div*> (*division*) et <*ab*> (*anonymous block*) qui peuvent être utilisés de façon récursive de façon à marquer un emboitement structurel. D'autres éléments peuvent être utilisés pour grouper des empans (<*spanGrp*> : ensemble de <*span*>) ou des interprétations (<*interpGrp*> : ensemble de <*interp*>), mais ils ne sont pas récursifs. Certes, la création d'éléments nouveaux comme <ThemeRheme> pourraient donner lieu à des contraintes syntaxiques spécifiques définies dans un schéma, mais il peut être hasardeux d'imposer ces contraintes syntaxiques dans un cadre d'analyse en progression. L'utilisation de valeurs spécifiques d'attributs dans des éléments généraux renvoie plutôt à des prescriptions locales d'analyse dans des éléments <*interp*>, par exemple. Des mécanismes de contraintes syntaxiques pourront être ajoutés au schéma TEI pour valider des constructions en fonction de valeurs d'attributs.

On pourrait aussi utiliser les éléments de pointage <*linkGrp*> et <*link*> pour marquer les relations *ThemeRheme* par des liens explicites établissant des relations entre le thème et ses rhèmes. Voici un exemple.

```
<linkGrp type="ThemeRheme" targFunc="thème rhème-1 rhème-2 rhème-x">
<link targets="#s1-4 #s5-12"/>
<link targets="#s13-13 #s14-18 #s19-23"/>
<link targets="#s24-27 #s28-29 #s30-32"/>
<link targets="#s33-34 #s35-37"/>
</linkGrp>
```

Dans l'exemple précédent, la valeur de l'attribut *type* de <*linkGrp*> indiquera que l'on a des liens de type *ThemeRheme* alors que l'attribut *targFunc* précisera la portée sémantique de chacun des pointeurs selon leur position dans la séquence. Suivent ensuite les liens eux-mêmes avec l'attribut *targets* qui contient la liste des pointeurs correspondant aux attributs *xml:id* des éléments <*span*> qui réfèrent aux énoncés en position de thème et de rhème.

Toujours dans l'idée d'utiliser les structures les plus simples possibles, on pourrait également remplacer l'élément non-TEI *ThemeRheme* par des éléments <*div*> ou <*ab*>. Dans l'exemple qui suit, après l'entête TEI, un premier <*div*> marque une division d'analyse de type *thème-rhème* contenant des blocs décrivant chacune des relations.



Marquage des occurrences par la balise w (document *ThemeRheme.xml)* (6.2.1b, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
<TEI>
   <teiHeader>
      <fileDesc>
         <titleStmt>
            <title>Analyse due à J.M. Adam d'un exemple de relations
               thèmes-rhèmes et de progressions thématiques</title>
         </titleStmt>
         <publicationStmt>
            <p>Publié par...</p>
         </publicationStmt>
         <sourceDesc>
            <bibl> ... </bibl>
         </sourceDesc>
      </fileDesc>

      <encodingDesc>
         <p>Utilisation des éléments div ab et span et interp pour exprimer
            en TEI l'analyse d'Adam</p>
      </encodingDesc>
   </teiHeader>
```

```
<text>
  <body>
    <div type="Analyse" subtype="ThèmeRhème" xml:id="ana1">
      <interpGrp type="Unités_discursives">
        <interp xml:id="Énoncé">On considèrera comme énoncé...</interp>
      </interpGrp>

      <interpGrp type="Thématisation">
        <interp xml:id="Thème">Le thème est l'énoncé qui se pose comme
          connu</interp>

        <interp xml:id="Rhème">Le rhème est un énoncé qui ajoute de
          l'information sur un énoncé thème</interp>

        <interp xml:id="ThèmeConstant">Le thème constant correspond à
          une progression thématique dans lequel un même thème est
          repris dans une suite de relations thèmes-rhèmes</interp>

        <interp xml:id="ThématisationLinéaire">La thématisation
          linéairee correspond à une progression thématique dans
          laquelle un rhème est repris à titre de thème dans la
          succession des énoncés.</interp>
      </interpGrp>

      <spanGrp xml:id="Seg1" type="Segmentation" ana="#Énoncé" xml:base="doc1.xml">
        <span from="#w1" to="#w4" xml:id="s1-4">Et un jour Jeanie</span>
        <span  from="#w5" to="#w12" xml:id="s5-12"> partit à la recherche de son amoureux.
</span>
        <span  from="#w13" to="#w13" xml:id="s13-13">Elle</span>
        <span  from="#w14" to="#w18" xml:id="s14-18">regardait les fleurs d'eau</span>
        <span  from="#w19" to="#w23" xml:id="s19-23">et leurs tiges penchées:</span>
        <span  from="#w24" to="#w27" xml:id="s24-27">et toutes les fleurs</span>
        <span  from="#w28" to="#w29" xml:id="s28-29">s'inclinaient</span>
        <span  from="#w30" to="#w32" xml:id="s30-32">vers elle.</span>
        <span  from="#w33" to="#w34" xml:id="s33-34">Et Jeanie</span>
        <span  from="#w35" to="#w37" xml:id="s35-37">disait en marchant</span>
      </spanGrp>

      <spanGrp xml:id="TR1" type="ThèmeRhème">
        <span ana="#Thème" xml:id="T1-4" from="#s1-4"
          >Th1 (thème initial en début de phrase)</span> <!-- Et un jour Jeanie -->
        <span ana="#Rhème" xml:id="R5-12" from="#s5-12">Rh1</span> <!-- partit à la
recherche de son amoureux. -->
      </spanGrp>

      <spanGrp xml:id="TR2" type="ThèmeRhème">
        <span ana="#Thème" xml:id="T13-13" from="#s13-13"
          >Th1</span> <!-- Elle -->
        <span ana="#Rhème" xml:id="R14-18" from="s14-18"
          >Rh2</span> <!-- regardait les fleurs d'eau -->
        <span ana="#Rhème" xml:id="R19-23" from="#s19-23"
          >Rhp3</span> <!-- et leurs tiges penchées: -->
      </spanGrp>

      <spanGrp xml:id="TR3" type="ThèmeRhème">
        <span ana="#Thème" xml:id="T24-27" from="#s24-27"
          >Th2 (=Rh2)</span> <!-- et toutes les fleurs -->
```

```
                    <span ana="#Rhème" xml:id="R28-29" from="#s28-29"
                        >Rh4 (=Rhp3)</span> <!-- s'inclinaient -->
                    <span ana="#Rhème" xml:id="R30-32" from="#s30-32"
                        >Rhp5 (=Th1)</span> <!-- vers elle. -->
                </spanGrp>

                <spanGrp xml:id="TR4" type="ThèmeRhème">
                    <span ana="#Thème" xml:id="T33-34" from="#s33-34"
                        >Th1</span> <!-- Et Jeanie -->
                    <span ana="#Rhème" xml:id="R35-37" from="#s35-37"
                        >Rh6</span> <!-- disait en marchant -->
                </spanGrp>

                <spanGrp xml:id="PT1" type="ProgressionThématique"
                    ana="#ThèmeConstant">
                    <span from="#T1-4">Th1 : thème
                        initial en début de phrase </span> <!-- Et un jour Jeanie -->
                    <span from="#T13-13">Th1 : anaphore pronominale</span> <!-- Elle -->
                    <span from="#T33-34">Th1 : reprise </span> <!-- Et Jeanie -->
                </spanGrp>

                <spanGrp xml:id="PT2" type="ProgressionThématique"
                    ana="#ThématisationLinéaire">
                    <span from="#R14-18">Rh2</span> <!-- regardait les fleurs d'eau -->
                    <span from="#T24-27">Th2</span> <!-- et toutes les fleurs -->
                </spanGrp>

                <spanGrp xml:id="PT3" type="ProgressionThématique"
                    ana="#ThématisationLinéaire">
                    <span from="#R30-32">Rhp5</span> <!-- vers elle -->
                    <span from="#T13-13">Th1 (chiasme qui rhématise
                        le pronom anaphorique de PT1)</span> <!--elle -->
                </spanGrp>
            </div>
        </body>
    </text>
</TEI>
```

L'élément <*div*> introduit une section de *type Analyse*. L'attribut *subtype* précise la nature de l'analyse tandis que l'attribut *xml:id* fournit un identifiant à cette analyse.

Ce bloc d'analyse contient d'abord un élément <*interpGrp*> qui contient les diverses définitions invoquées en cours de l'analyse pour opérer l'annotation structurelle. Il est à noter que ces définitions pourraient aussi être dans un document externe puisque le pointage vers ces éléments d'interprétation est un URI pouvant donc localiser un élément dans un document externe.

Ensuite, on a un premier <*spanGrp*> de type *Segmentation* qui contient la liste des segments impliqués dans l'analyse à titre d'énoncés. Les <*span*> réfèrent aux mots du document externe analysé (attribut *xml:base*). Logiquement, cette liste n'est pas obligatoire puisqu'on peut

pointer directement sur des empans textuels dans les annotations qui suivent. La liste pourrait donc être calculée à partir des annotations elles-mêmes. Ici, nous l'explicitons dans un élément <*spanGrp*> de type *Segmentation* dans le but de rendre la représentation transparente dans la suite de l'exposé. Le contenu des éléments <*span*> contient des commentaires explicatifs supplémentaires facilitant la lecture humaine. Le texte intégral, ajouté en commentaire, peut être retrouvé par les attributs *from* et *to* de chacun des <*span*>.

Viennent ensuite les <*spanGrp*> de type *ThèmeRhème*. Chaque groupe contient une suite d'éléments <*span*> avec un attribut *ana* et ses valeurs #*Thème* et #*Rhèm*e, selon la catégorie analytique appliquée à l'empan. Il s'agit de pointeurs vers un texte libre dans des éléments <*interp*>. On aurait aussi pu pointer sur des structures de traits (*fs*) . On utilise le contenu des <*span*> pour introduire un texte explicatif. Dans ces exemples, la valeur *ThèmeRhème* de l'attribut *type* de <*spanGrp*> prend donc la place de l'ancien élément ThemeRheme de notre première hypothèse de représentation. La valeur de l'attribut *xml:id* du <*spanGrp*> donne un identifiant unique à chacune des relations.

Les relations thèmes-rhèmes se complètent par des relations de progression thématique reliant les thèmes entre eux. La structure de progression linéaire, par exemple, indique qu'un élément *rhématisé* est repris à titre de thème dans une autre relation. Ces relations sont introduites par des <*spanGrp*> de type *progression_thématique*. L'attribut *ana* précise le type de progression impliquée. Les <*span*> identifient les énoncés impliqués dans la structure. Ce sont des énoncés qui ont déjà été analysés dans des structures *ThèmeRhème* identifiant le rôle des énoncés à titre de thème ou de rhème. Voilà pourquoi on pointe sur une structure interne au document d'analyse et non pas directement au texte analysé. Celui-ci peut être retrouvé par une évaluation récursive des pointeurs. Le contenu des <*span*> permet d'apporter des commentaires explicatifs supplémentaires, s'il y a lieu.

Si on avait voulu utiliser des éléments de pointage <*ref*> plutôt que des <*span*>, on aurait eu une notation plus condensée comme dans les exemples suivants.

<ref type="progression_thématique" target="#T1-4 #T13-13 #T33-34" ana="#thème_constant"> Et un jour Jeanie ... Elle ... Et Jeanie</ref>

<ref type="progression_thématique" target="#R14-18 #T24-27" ana="thématisation_linéaire">regardait les fleurs d'eau ...et toutes les fleurs </ref>

<ref type="progression_thématique" target="#R30-32 #T13-13" ana="thématisation_linéaire"> vers elle ...elle </ref>

Le contenu de l'élément <*ref*> est utilisé pour commenter la progression. Ici, on reprend simplement des parties de texte. L'attribut *ana* est utilisé pour qualifier la configuration de cette progression thématique en renvoyant à un élément <*interp*> ou à une structure de traits. L'attribut *target* contient une liste de pointeurs sur des <*span*> déjà définis. Moins compacte, l'utilisation des <*span*> a cependant l'avantage de permettre de commenter la contribution de chaque énoncé à la structure de progression thématique.

6.2.1.4 Annotation structurelle. : utilisation des structures de graphe

On pourrait aussi exprimer l'annotation en utilisant directement un formalisme de graphe. Dans l'exemple qui suit, nous utiliserons le formalisme de graphe proposé par la TEI. Concevoir, dès le départ, l'annotation structurelle comme la construction d'un graphe a l'avantage de suggérer une ergonomie et des interfaces informatiques développées pour l'affichage et l'édition de graphes. Cela dit, une transformation, simple algorithmiquement, des formalismes précédents peut conduire à la construction du graphe. Ainsi, par rapport au formalisme d'annotation déjà présenté, les segments textuels correspondent aux nœuds du graphe tandis que les relations entre segments pointés correspondent aux arcs.

Voici une définition TEI du graphe calculé par une feuille de style XSLT appliquée au fichier *ana1.xml*.



ThemeRheme-graph.xml (6.2.1c, exemple)

```
<?xml version="1.0" encoding="utf-8"?><?oxygen RNGSchema="http://www.tei-
c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="xml"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Analyse due à J.M. Adam d'un exemple de relations
              thèmes-rhèmes et de progressions thématiques</title>
      </titleStmt>
      <publicationStmt>
        <p>Publié par...</p>
      </publicationStmt>
      <sourceDesc>
        <bibl> ... </bibl>
      </sourceDesc>
    </fileDesc>
    <encodingDesc>
      <p>Utilisation des éléments div ab et span et interp pour exprimer
           en TEI l'analyse d'Adam</p>
```

```
    </encodingDesc>
  </teiHeader>

  <text>
    <body>
      <interpGrp type="Unités_discursives">
        <interp xml:id="Énoncé">On considèrera comme énoncé...</interp>
      </interpGrp>
      <interpGrp type="Thématisation">
        <interp xml:id="Thème">Le thème est l'énoncé qui se pose comme
            connu</interp>
        <interp xml:id="Rhème">Le rhème est un énoncé qui ajoute de
            l'information sur un énoncé thème</interp>
        <interp xml:id="ThèmeConstant">Le thème constant correspond à
            une progression thématique dans lequel un même thème est
            repris dans une suite de relations thèmes-rhèmes</interp>
        <interp xml:id="ThématisationLinéaire">La thématisation
            linéairee correspond à une progression thématique dans
            laquelle un rhème est repris à titre de thème dans la
            succession des énoncés.</interp>
      </interpGrp>

      <spanGrp xml:id="Seg1" type="Segmentation" ana="#Énoncé" xml:base="doc1.xml">
        <span from="#w1" to="#w4" xml:id="s1-4">Et un jour Jeanie</span>
        <span from="#w5" to="#w12" xml:id="s5-12"> partit à la recherche de son amoureux.
</span>
        <span from="#w13" to="#w13" xml:id="s13-13">Elle</span>
        <span from="#w14" to="#w18" xml:id="s14-18">regardait les fleurs d'eau</span>
        <span from="#w19" to="#w23" xml:id="s19-23">et leurs tiges penchées:</span>
        <span from="#w24" to="#w27" xml:id="s24-27">et toutes les fleurs</span>
        <span from="#w28" to="#w29" xml:id="s28-29">s'inclinaient</span>
        <span from="#w30" to="#w32" xml:id="s30-32">vers elle.</span>
        <span from="#w33" to="#w34" xml:id="s33-34">Et Jeanie</span>
        <span from="#w35" to="#w37" xml:id="s35-37">disait en marchant</span>
      </spanGrp>

      <graph type="directed" xml:id="ana1">
        <label>ThèmeRhème</label>

        <node xml:id="ana1_T1-4" value="#s1-4">
          <label ana="#Thème">Th1 (thème initial en début de phrase)</label>
        </node>
        <node xml:id="ana1_R5-12" value="#s5-12">
          <label ana="#Rhème">Rh1</label>
        </node>
        <node xml:id="ana1_T13-13" value="#s13-13">
          <label ana="#Thème">Th1</label>
        </node>
        <node xml:id="ana1_R14-18" value="s14-18">
          <label ana="#Rhème">Rh2</label>
        </node>
        <node xml:id="ana1_R19-23" value="#s19-23">
          <label ana="#Rhème">Rhp3</label>
        </node>
        <node xml:id="ana1_T24-27" value="#s24-27">
          <label ana="#Thème">Th2 (=Rh2)</label>
        </node>
```

```
            <node xml:id="ana1_R28-29" value="#s28-29">
              <label ana="#Rhème">Rh4 (=Rhp3)</label>
            </node>
            <node xml:id="ana1_R30-32" value="#s30-32">
              <label ana="#Rhème">Rhp5 (=Th1)</label>
            </node>
            <node xml:id="ana1_T33-34" value="#s33-34">
              <label ana="#Thème">Th1</label>
            </node>
            <node xml:id="ana1_R35-37" value="#s35-37">
              <label ana="#Rhème">Rh6</label>
            </node>

            <arc from="#ana1_T1-4" to="#ana1_R5-12">
              <label>ThèmeRhème</label>
            </arc>
            <arc from="#ana1_T13-13" to="#ana1_R14-18">
              <label>ThèmeRhème</label>
            </arc>
            <arc from="#ana1_T13-13" to="#ana1_R19-23">
              <label>ThèmeRhème</label>
            </arc>
            <arc from="#ana1_T24-27" to="#ana1_R28-29">
              <label>ThèmeRhème</label>
            </arc>
            <arc from="#ana1_T24-27" to="#ana1_R30-32">
              <label>ThèmeRhème</label>
            </arc>
            <arc from="#ana1_T33-34" to="#ana1_R35-37">
              <label>ThèmeRhème</label>
            </arc>
            <arc from="#ana1_T1-4" to="#ana1_T13-13">
              <label n="Th1 : thème initial en début de phrase // Th1 : anaphore
pronominale">ThèmeConstant</label>
            </arc>
            <arc from="#ana1_T13-13" to="#ana1_T33-34">
              <label n="Th1 : anaphore pronominale // Th1 : reprise">ThèmeConstant</label>
            </arc>
            <arc from="#ana1_R14-18" to="#ana1_T24-27">
              <label n="Rh2 // Th2">ThématisationLinéaire</label>
            </arc>
            <arc from="#ana1_R30-32" to="#ana1_T13-13">
              <label n="Rhp5 // Th1 (chiasme qui rhématise le pronom anaphorique de
PT1)">ThématisationLinéaire</label>
            </arc>

          </graph>
        </body>
      </text>
    </TEI>
```

Ce nouveau document reprend l'entête du document *ana1.xml*, la liste des interprétations de même que la liste des segments impliqués par l'analyse avec la référence au document analysé. Mais les relations sont remplacées par un élément <*graph*> du même nom. Le

graphe associe un nœud à chacun des éléments <*span*> utilisés dans les relations *ThèmeRhème*. L'attribut *value* du nœud pointe sur le <*span*> qui réfère au corpus analysé. Un élément <*label*> fournit une étiquette au nœud qui reprend le contenu des <*span*>. L'attribut *ana* du <*label*> rappelle la fonction de l'empan représenté par le nœud. Dans ce graphe, les arcs entre les nœuds correspondent à deux types de relation entre segments. Il y a la relation *thème-rhème* et la relation de progression thématique qui se caractérise ici en *thème_constant* et *thématisation_linéaire*.

On notera que la représentation en graphe correspond à une transformation directe de l'annotation structurelle utilisant les éléments <*span*> et <*spanGrp*>. Elle a été produite la une feuille de style XSLT *ThemeRheme-graph.xsl* appliquée sur l'annotation structurelle.



Feuille de style transformant en graphe l'analyse thème-rhème (fichier *ThemeRheme-graph.xsl*). (6.2.1d, exemple)

```
<?xml version="1.0" encoding="UTF-8"?><xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
   xmlns:TEI="http://www.tei-c.org/ns/1.0">
   <xsl:strip-space elements="*"/>
   <xsl:output method="xml" indent="yes"/>
   <xsl:variable name="newline">
      <xsl:text>
</xsl:text>
   </xsl:variable>

   <xsl:template match="/">
      <xsl:call-template name="teiHeader"/>
   </xsl:template>

   <xsl:template name="teiHeader">
      <xsl:apply-templates mode="teiHeader"/>
   </xsl:template>

   <xsl:template match="title" mode="teiHeader">
      <xsl:copy>
         <xsl:value-of select="."/>
         <xsl:value-of select="$newline"/>
         <xsl:text> Présentation sous forme de graphe.</xsl:text>
      </xsl:copy>
   </xsl:template>

   <xsl:template match="@*|node()" mode="teiHeader">
      <xsl:copy>
         <xsl:apply-templates select="@*" mode="teiHeader"/>
         <xsl:choose>
            <xsl:when test="TEI:body">
               <xsl:apply-templates mode="body" select="."/>
            </xsl:when>
```

```
            <xsl:otherwise>
               <xsl:apply-templates mode="teiHeader"/>
            </xsl:otherwise>
         </xsl:choose>
      </xsl:copy>
   </xsl:template>

   <xsl:template match="TEI:body" mode="body">
      <xsl:copy>
         <xsl:for-each select="@*">
            <xsl:copy/>
         </xsl:for-each>
         <xsl:apply-templates mode="body"/>
      </xsl:copy>
   </xsl:template>

   <xsl:template match="TEI:div[@type='Analyse' and @subtype='ThèmeRhème']" mode="body">
      <!-- interp -->
      <xsl:for-each select="TEI:interpGrp">
         <xsl:apply-templates mode="teiHeader" select="."/>
      </xsl:for-each>
      <xsl:value-of select="$newline"/>

      <!-- Segmentation -->
      <xsl:for-each select="TEI:spanGrp[@type='Segmentation']">
         <xsl:apply-templates mode="teiHeader" select="."/>
      </xsl:for-each>
      <xsl:value-of select="$newline"/>

      <!-- graph -->
      <xsl:element name="graph">
         <xsl:attribute name="type">
            <xsl:text>directed</xsl:text>
         </xsl:attribute>
         <xsl:attribute name="xml:id">
            <xsl:value-of select="@xml:id"/>
         </xsl:attribute>
         <!-- label -->
         <xsl:element name="label">
            <xsl:value-of select="@subtype"/>
         </xsl:element>
         <xsl:value-of select="$newline"/>
         <!-- node -->
         <xsl:call-template name="noeud-ThèmeRhème">
            <xsl:with-param name="graph_nom">
               <xsl:value-of select="@xml:id"/>
            </xsl:with-param>
         </xsl:call-template>
         <xsl:value-of select="$newline"/>
         <!-- arcs ThèmeRhème -->
         <xsl:call-template name="ThèmeRhème">
            <xsl:with-param name="graph_nom">
               <xsl:value-of select="@xml:id"/>
            </xsl:with-param>
         </xsl:call-template>
         <!-- arcs ProgressionThématique -->
         <xsl:call-template name="ProgressionThématique">
```

```
            <xsl:with-param name="graph_nom">
               <xsl:value-of select="@xml:id"/>
            </xsl:with-param>
         </xsl:call-template>
      </xsl:element>
   </xsl:template>

   <xsl:template name="noeud-ThèmeRhème">
      <xsl:param name="graph_nom"/>
      <xsl:for-each select="TEI:spanGrp[@type='ThèmeRhème']/TEI:span">
         <xsl:element name="node">
            <xsl:attribute name="xml:id">
               <xsl:value-of select="$graph_nom"/>
               <xsl:text>_</xsl:text>
               <xsl:value-of select="@xml:id"/>
            </xsl:attribute>
            <xsl:attribute name="value">
               <xsl:value-of select="@from"/>
            </xsl:attribute>
            <xsl:if test="@n">
               <xsl:attribute name="n">
                  <xsl:value-of select="@n"/>
               </xsl:attribute>
            </xsl:if>
            <xsl:element name="label">
               <xsl:if test="attribute::ana">
                  <xsl:attribute name="ana">
                     <xsl:value-of select="@ana"/>
                  </xsl:attribute>
               </xsl:if>
               <xsl:value-of select="normalize-space(.)"/>
               <xsl:value-of select="@n"/>
            </xsl:element>
         </xsl:element>
         <xsl:value-of select="$newline"/>
      </xsl:for-each>
   </xsl:template>

   <xsl:template name="ThèmeRhème">
      <xsl:param name="graph_nom"/>
      <xsl:for-each select="TEI:spanGrp[@type='ThèmeRhème']">
         <!-- from -->
         <xsl:variable name="thème">
            <xsl:value-of select="TEI:span[@ana='#Thème']/@xml:id"/>
         </xsl:variable>
         <!-- n-theme -->
         <xsl:variable name="thème-contenu">
            <xsl:value-of select="TEI:span[@ana='#Thème']"/>
         </xsl:variable>
         <!-- arc -->
         <xsl:for-each select="TEI:span">
            <xsl:if test="@ana='#Rhème'">
               <xsl:element name="arc">
                  <xsl:attribute name="from">
                     <xsl:text>#</xsl:text>
                     <xsl:value-of select="$graph_nom"/>
                     <xsl:text>_</xsl:text>
```

```
                        <xsl:value-of select="$thème"/>
                    </xsl:attribute>
                    <xsl:attribute name="to">
                        <xsl:text>#</xsl:text>
                        <xsl:value-of select="$graph_nom"/>
                        <xsl:text>_</xsl:text>
                        <xsl:value-of select="@xml:id"/>
                    </xsl:attribute>

                    <xsl:element name="label">
                        <xsl:text>ThèmeRhème</xsl:text>
                    </xsl:element>
                </xsl:element>
                <xsl:value-of select="$newline"/>
            </xsl:if>
        </xsl:for-each>
    </xsl:for-each>
</xsl:template>

<xsl:template name="ProgressionThématique">
    <xsl:param name="graph_nom"/>
    <xsl:for-each select="TEI:spanGrp[@type='ProgressionThématique']">
        <!-- ana -->
        <xsl:variable name="ana">
            <xsl:value-of select="substring-after(@ana,'#')"/>
        </xsl:variable>
        <!-- arc -->
        <xsl:for-each select="TEI:span">
            <xsl:if test="following-sibling::TEI:span">
                <xsl:element name="arc">
                    <xsl:attribute name="from">
                        <xsl:text>#</xsl:text>
                        <xsl:value-of select="$graph_nom"/>
                        <xsl:text>_</xsl:text>
                        <xsl:value-of select="substring-after(@from,'#')"/>
                    </xsl:attribute>
                    <xsl:attribute name="to">
                        <xsl:text>#</xsl:text>
                        <xsl:value-of select="$graph_nom"/>
                        <xsl:text>_</xsl:text>
                        <xsl:value-of
                            select="substring-after(following-sibling::TEI:span/@from,'#')"/>
                    </xsl:attribute>

                    <xsl:element name="label">
                        <xsl:attribute name="n">
                            <xsl:value-of select="normalize-space(.)"/>
                            <xsl:text> // </xsl:text>
                            <xsl:value-of select="normalize-space(following-sibling::TEI:span)"
                            />
                        </xsl:attribute>
                        <xsl:value-of select="$ana"/>
                    </xsl:element>
                </xsl:element>
                <xsl:value-of select="$newline"/>
            </xsl:if>
        </xsl:for-each>
```

```
            </xsl:for-each>
        </xsl:template>

        <xsl:template match="node()" mode="body">
            <xsl:apply-templates mode="body"/>
        </xsl:template>

    </xsl:stylesheet>
```

D'autres types d'arcs pourraient être ajoutés au graphe, par exemple pour représenter les anaphores qui nous permettent de conclure à une relation de thème constant. Ces précisions font présentement partie des contenus textuels des éléments <*span*> qui sont à l'origine des nœuds du graphe. L'ajout, sous forme d'arcs dans le graphe, des relations d'anaphore et de reprise supposerait cependant une formalisation plus poussée que l'explication libre utilisée dans le contenu textuel des <*span*>. L'ajout de ces arcs, représentant diverses relations de liage, pourrait prendre la forme suivante.

```
<arc from="#G1-T13-13" to="G1-T1-4"> <label  ana="#liage">anaphore-pronominale</label> </arc>
<arc from="#G1-T33-34" to="G1-T13-13"> <label n="liage">reprise</label> </arc>
<arc from="#G-R19-23" to="G1-R8-29"> <label n="liage">sémantique</label> </arc>
<arc from="#G1-R30-32" to="G1-T13-13"> <label n="liage">anaphore-pronominale</label> </arc>
```

L'intérêt de la représentation graphique est de permettre une juxtaposition visuelle de structures diverses. L'utilisation des couleurs pourrait permettre de souligner l'appartenance des arcs à des structures de nature différente.

Comme on le verra plus loin dans la section 7.5, la manipulation de la représentation graphique peut aussi servir d'interface pour créer ou modifier des annotations. Ces outils graphiques peuvent également être utilisés pour créer des motifs qui seront traduits automatiquement en requêtes dans un langage de fouille des annotations.

6.2.2. Analyse textuelle d'un récit de Jorge Luis Borges.

Afin de valider davantage la puissance de représentation des formalismes TEI déjà utilisés pour rendre compte de la perspective fonctionnelle de la phrase, nous tenterons, dans cette section, de représenter conjointement plusieurs perspectives d'analyse. L'idée ici est d'exploiter les formalismes déjà discutés pour représenter les diverses dimensions d'analyse que l'on retrouve illustrées dans l'analyse d'un court texte de Borges développée en conclusion de l'ouvrage d'Adam sur la linguistique textuelle (Adam2005). Cette nouvelle exploration

nous donnera l'occasion de présenter un document d'annotation avec son entête. Voici d'abord le texte tel que présenté dans le livre d'Adam.



Un récit de Borges : *LE CAPTIF* en format texte (6.2.2a, exemple)

LE CAPTIF

[él] À Junín ou à Tapalqué, on raconte l'histoire suivante. [é2a] Un enfant disparut après un raid d'Indiens ; [é2b] on dit que les Indiens l'avaient enlevé. [é3a] Ses parents le cherchèrent inutilement ; [é3b] des années plus tard, un soldat qui venait de l'intérieur leur parla d'un Indien aux yeux couleur de ciel qui pouvait bien être leur fils. [é4a] Ils le rencontrèrent enfin ([é4b] la chronique a perdu les circonstances [é4c] et je ne veux pas inventer ce que je ne sais pas) [é4d] et ils crurent le reconnaître. [é5a] L'homme, marqué par le désert et la vie sauvage, ne comprenait déjà plus les mots de sa langue natale, [é5b] mais, indifférent et docile, il se laissa conduire à la maison. [é6a] Il s'arrêta sur le seuil, [é6b] peut-être parce que les autres s'y arrêtèrent. [é7a] Il regarda la porte, [é7b] comme s'il ne la comprenait pas. [é8a] Soudain, il baissa la tête, [é8b] poussa un cri, [é8e] traversa en courant le corridor et les deux vastes cours [é8d] et pénétra dans la cuisine. [é9a] Sans hésiter, il plongea le bras dans la hotte enfumée [é9b] et sortit le petit couteau à manche de corne qu'il avait caché là, [é9c] lorsqu'il était enfant. [él0a] Ses yeux brillèrent de joie [él0b] et ses parents pleurèrent, [él0c] parce qu'ils avaient retrouvé leur fils.
[élla] Ce souvenir fut peut-être suivi par d'autres, [éllb] mais l'Indien ne pouvait vivre entre quatre murs [é11c] et un jour il partit à la recherche de son désert. [é12a] Moi je voudrais savoir [é12b] ce qu'il ressentit en cet instant de vertige [é12c] où le passé et le présent se confondirent ; [é12d] moi je voudrais savoir [é12e] si le fils perdu renaquit et mourut en cette extase, [é12f] ou s'il parvint à reconnaître, [é12g] ne fût-ce qu'à la manière d'un nouveau-né ou d'un chien, [é12f_fin] les parents et la maison.
(Jorge Luis Borges, El Hacedor (J 960). Traduction de J.-M. Adam, Adam2005:203-204)

Nous avons numérisé le texte à partir de l'édition papier et nous l'avons mis en format SATO. On y trouvera d'abord un entête SATO se terminant par une déclaration de titre. Le découpage en énoncés est marqué par la propriété textuelle *Én* dont les valeurs reprennent l'annotation d'Adam. Dans la transcription qui suit, les affectations de propriété ont été mises en gras pour en faciliter le repérage. La référence bibliographique en fin de texte est marquée comme un commentaire et ne sera pas lexicalisée par SATO.



Un récit de Borges : *LE CAPTIF* en format SATO (fichier *borges_adam.sat*). (6.2.2.b, exemple)

Alphabet fr
Caractère citation \
Caractère propriété *
propriété Én symbolique pour texte
Titre LE CAPTIF : Borges, El Hacedor. Traduction de J.-M. Adam

***page=/203**

Én="él"** À Junín ou à Tapalqué, on raconte l'histoire suivante. ***Én="é2a"** Un enfant disparut après un raid d'Indiens ; ***Én="é2b"** on dit que les Indiens l'avaient enlevé. ***Én="é3a"** Ses parents le cherchèrent inutilement ; ***Én="é3b"** des années plus tard, un soldat qui venait de l'intérieur leur parla d'un Indien aux yeux couleur de ciel qui pouvait bien être leur fils. ***Én="é4a"** Ils le rencontrèrent enfin (***Én="é4b"** la chronique a perdu les circonstances ***Én="é4c"** et je ne veux pas inventer ce que je ne sais pas) *Én="é4d" et ils crurent le reconnaître. ***Én="é5a"** L'homme, marqué par le désert et la vie sauvage, ne comprenait déjà plus les mots de sa langue natale, ***Én="é5b"** mais, indifférent et docile, il se laissa conduire à la maison. ***Én="é6a"** Il s'arrêta sur le ***page=/204** seuil, ***Én="é6b"** ***(**peut-être)** parce que les autres s'y arrêtèrent. ***Én="é7a"** Il regarda la porte, ***Én="é7b"** comme s'il ne la comprenait pas. ***Én="é8a"** Soudain, il baissa la tête, *Én="é8b" poussa un cri, ***Én="é8c"** traversa en courant le corridor et les deux vastes cours *Én="é8d" et pénétra dans la cuisine. ***Én="é9a"** Sans hésiter, il plongea le bras dans la hotte enfumée ***Én="é9b"** et sortit le petit couteau à manche de corne qu'il avait caché là, ***Én="é9c"** lorsqu'il était enfant. ***Én="él0a"** Ses yeux brillèrent de joie ***Én="él0b"** et ses parents pleurèrent, ***Én="él0c"** parce qu'ils avaient retrouvé leur fils.

Én="élla"** Ce souvenir fut ***(**peut-être)** suivi par d'autres, ***Én="éllb"** mais l'Indien ne pouvait vivre entre quatre murs ***Én="é11c"** et un jour il partit à la recherche de son désert. ***Én="é12a"** Moi je voudrais savoir ***Én="é12b"** ce qu'il ressentit en cet instant de vertige ***Én="é12c"** où le passé et le présent se confondirent ; ***Én="é12d"** moi je voudrais savoir ***Én="é12e"** si le fils perdu renaquit et mourut en cette extase, ***Én="é12f"** ou s'il parvint à reconnaître, ***Én="é12g"** ne fût-ce qu'à la manière d'un ***(**nouveau-né***)** ou d'un chien, ***Én="é12f fin"** les parents et la maison.

*{(Jorge Luis Borges, El Hacedor (J 960). Traduction de J.-M. Adam, Adam2005:203-204)}

À partir de cette version soumise à SATO, on obtient un découpage en formes lexicales d'après la définition standard de l'alphabet français (*fr*). Les balises **(* et **)* sont utilisées pour encadrer les mots composés que l'on veut lexicaliser tels quels (*nouveau-né, peut-être*). On utilise ensuite SATO pour produire une exportation en XML-TEI suivant le protocole ATONET. Dans ce fichier (*borges_adam.xml),* chacun des mots est encadré par une balise <*w*> possédant un identificateur unique dans l'attribut *xml:id.* Le découpage en énoncés est marqué par des balises vides <*milestone*>. On y retrouve aussi les autres balises de la proposition ATONET permettant de marquer les pages, les paragraphes et les lignes. Voici le contenu du fichier du fichier XML produit par SATO.



Un récit de Borges : *LE CAPTIF* en format TEI (fichier *borges_adam.xml*). (6.2.2c, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt>
```

<title>LE CAPTIF : Borges, El Hacedor. Traduction de J.-M. Adam</title>
</titleStmt>
<publicationStmt> Publié par... </publicationStmt>
<sourceDesc> <bibl>...</bibl></sourceDesc>
</fileDesc>
<encodingDesc>
<refsDecl>
Les balises «milestone n="valeur-de propriété" unit="nom-de-propriété"» concernent les mots qui suivent la balise jusqu'à l'apparition d'un nouveau milestone de même «unit». Les références de pagination utilisent les balises pb (début de page), lb(début de ligne) .

milestone Én symbol "él" "é2a" "é2b" "é3a" "é3b" "é4a" "é4b" "é4c" "é4d" "é5a" "é5b" "é6a" "é6b" "é7a" "é7b" "é8a" "é8b" "é8e" "é8d" "é9a" "é9b" "é9c" "él0a" "él0b" "él0c" "élla" "éllb" "é11c" "é12a" "é12b" "é12c" "é12d" "é12e" "é12f" "é12g" ""é12f_fin""
Le découpage en mots a été effectué par le logiciel SATO en utilisant les règles décrites dans les déclarations d'alphabet suivantes.

<?sato cmd="Alphabet fr ,0 .0 ,1 .1 ,2 .2 ,3 .3 ,4 .4 ,5 .5 ,6 .6 ,7 .7 ,8 .8 ,9 .9 '_ aujourd' presqu'île presqu'ile 's *séparateur - , : ; . ? ¿ ! ... … < > ( ) [ ] { } « » % $ £ ¢ ¥ # " @ & + = / \ | * ÷ ± ® ¦ *terminal ' ´ ` ’
"?>
</refsDecl>
</encodingDesc>
</teiHeader>
<text>
<body>

<pb n="borges_adam/203"/>
<lb n="1"/><milestone unit="Én" n="**él**"/><w xml:id="w2">À</w> <w xml:id="w3">Junín</w> <w xml:id="w4">ou</w> <w xml:id="w5">à</w> <w xml:id="w6">Tapalqué</w><w xml:id="w7">,</w> <w xml:id="w8">on</w> <w xml:id="w9">raconte</w> <w xml:id="w10">l'</w><w xml:id="w11">histoire</w> <w xml:id="w12">suivante</w><w xml:id="w13">.</w> <milestone unit="Én" n="**é2a**"/><w xml:id="w14">Un</w> <w xml:id="w15">enfant</w> <w xml:id="w17">disparut</w>

<lb n="2"/><w xml:id="w18">après</w> <w xml:id="w19">un</w> <w xml:id="w20">raid</w> <w xml:id="w21">d'</w><w xml:id="w22">Indiens</w> <w xml:id="w23">;</w> <milestone unit="Én" n="**é2b**"/><w xml:id="w24">on</w> <w xml:id="w25">dit</w> <w xml:id="w26">que</w> <w xml:id="w27">les</w> <w xml:id="w28">Indiens</w> <w xml:id="w30">l'</w><w xml:id="w31">avaient</w> <w xml:id="w32">enlevé</w><w xml:id="w33">.</w>

<lb n="3"/><milestone unit="Én" n="**é3a**"/><w xml:id="w34">Ses</w> <w xml:id="w35">parents</w> <w xml:id="w36">le</w> <w xml:id="w37">cherchèrent</w> <w xml:id="w38">inutilement</w> <w xml:id="w39">;</w> <milestone unit="Én" n="**é3b**"/><w xml:id="w40">des</w> <w xml:id="w42">années</w> <w xml:id="w43">plus</w> <w xml:id="w44">tard</w><w xml:id="w45">,</w> <w xml:id="w46">un</w> <w xml:id="w47">soldat</w>

<lb n="4"/><w xml:id="w48">qui</w> <w xml:id="w49">venait</w> <w xml:id="w50">de</w> <w xml:id="w51">l'</w><w xml:id="w52">intérieur</w> <w xml:id="w53">leur</w> <w xml:id="w55">parla</w> <w xml:id="w56">d'</w><w xml:id="w57">un</w> <w xml:id="w58">Indien</w> <w xml:id="w59">aux</w> <w xml:id="w60">yeux</w> <w xml:id="w61">couleur</w> <w xml:id="w62">de</w> <w xml:id="w63">ciel</w> <w

xml:id="w64">qui</w>

<lb n="5"/><w xml:id="w65">pouvait</w> <w xml:id="w66">bien</w> <w
xml:id="w68">être</w> <w xml:id="w69">leur</w> <w xml:id="w70">fils</w><w
xml:id="w71">.</w> <milestone unit="Én" n="**é4a**"/><w xml:id="w72">Ils</w> <w
xml:id="w73">le</w> <w xml:id="w74">rencontrèrent</w> <w xml:id="w75">enfin</w> <w
xml:id="w76">(</w> <milestone unit="Én" n="**é4b**"/><w xml:id="w77">la</w> <w
xml:id="w78">chronique</w> <w xml:id="w79">a</w>

<lb n="6"/><w xml:id="w80">perdu</w> <w xml:id="w82">les</w> <w
xml:id="w83">circonstances</w> <milestone unit="Én" n="**é4c**"/><w xml:id="w84">et</w> <w
xml:id="w85">je</w> <w xml:id="w86">ne</w> <w xml:id="w87">veux</w> <w
xml:id="w88">pas</w> <w xml:id="w89">inventer</w> <w xml:id="w90">ce</w> <w
xml:id="w91">que</w> <w xml:id="w92">je</w> <w xml:id="w93">ne</w> <w
xml:id="w94">sais</w> <w xml:id="w96">pas</w><w xml:id="w97">)</w>

<lb n="7"/><milestone unit="Én" n="**é4d**"/><w xml:id="w98">et</w> <w xml:id="w99">ils</w>
<w xml:id="w100">crurent</w> <w xml:id="w101">le</w> <w
xml:id="w102">reconnaître</w><w xml:id="w103">.</w> <milestone unit="Én" n="**é5a**"/><w
xml:id="w104">L'</w><w xml:id="w105">homme</w><w xml:id="w106">,</w> <w
xml:id="w107">marqué</w> <w xml:id="w108">par</w> <w xml:id="w109">le</w> <w
xml:id="w110">désert</w> <w xml:id="w112">et</w> <w xml:id="w113">la</w> <w
xml:id="w114">vie</w>

<lb n="8"/><w xml:id="w115">sauvage</w><w xml:id="w116">,</w> <w
xml:id="w117">ne</w> <w xml:id="w118">comprenait</w> <w xml:id="w119">déjà</w> <w
xml:id="w120">plus</w> <w xml:id="w121">les</w> <w xml:id="w122">mots</w> <w
xml:id="w123">de</w> <w xml:id="w124">sa</w> <w xml:id="w125">langue</w> <w
xml:id="w127">natale</w><w xml:id="w128">,</w> <milestone unit="Én" n="**é5b**"/><w
xml:id="w129">mais</w><w xml:id="w130">,</w>

<lb n="9"/><w xml:id="w131">indifférent</w> <w xml:id="w132">et</w> <w
xml:id="w133">docile</w><w xml:id="w134">,</w> <w xml:id="w135">il</w> <w
xml:id="w136">se</w> <w xml:id="w137">laissa</w> <w xml:id="w138">conduire</w> <lb
n="11"/><w xml:id="w140">à</w> <w xml:id="w141">la</w> <w
xml:id="w142">maison</w><w xml:id="w143">.</w> <milestone unit="Én" n="**é6a**"/><w
xml:id="w144">Il</w> <w xml:id="w145">s'</w><w xml:id="w146">arrêta</w> <w
xml:id="w147">sur</w> <w xml:id="w148">le</w>

<pb n="borges_adam/204"/>
<lb n="10"/><w xml:id="w149">seuil</w><w xml:id="w150">,</w> <milestone unit="Én"
n="**é6b**"/><w xml:id="w151">peut-être</w> <w xml:id="w152">parce</w> <w
xml:id="w153">que</w> <w xml:id="w154">les</w> <w xml:id="w155">autres</w> <w
xml:id="w157">s'</w><w xml:id="w158">y</w> <w xml:id="w159">arrêtèrent</w><w
xml:id="w160">.</w> <milestone unit="Én" n="**é7a**"/><w xml:id="w161">Il</w> <w
xml:id="w162">regarda</w> <w xml:id="w163">la</w>

<lb n="11"/><w xml:id="w164">porte</w><w xml:id="w165">,</w> <milestone unit="Én"
n="**é7b**"/><w xml:id="w166">comme</w> <w xml:id="w167">s'</w><w xml:id="w168">il</w>
<w xml:id="w169">ne</w> <w xml:id="w170">la</w> <w xml:id="w171">comprenait</w> <w
xml:id="w173">pas</w><w xml:id="w174">.</w> <milestone unit="Én" n="**é8a**"/><w
xml:id="w175">Soudain</w><w xml:id="w176">,</w> <w xml:id="w177">il</w> <w
xml:id="w178">baissa</w> <w xml:id="w179">la</w> <w xml:id="w180">tête</w><w
xml:id="w181">,</w>

<lb n="12"/><milestone unit="Én" n="**é8b**"/><w xml:id="w182">poussa</w> <w
xml:id="w183">un</w> <w xml:id="w184">cri</w><w xml:id="w185">,</w> <milestone

unit="Én" n="**é8c**"/><w xml:id="w186">traversa</w> <w xml:id="w187">en</w> <w xml:id="w188">courant</w> <w xml:id="w190">le</w> <w xml:id="w191">corridor</w> <w xml:id="w192">et</w> <w xml:id="w193">les</w> <w xml:id="w194">deux</w> <w xml:id="w195">vastes</w>

<lb n="13"/><w xml:id="w196">cours</w> <milestone unit="Én" n="**é8d**"/><w xml:id="w197">et</w> <w xml:id="w198">pénétra</w> <w xml:id="w199">dans</w> <w xml:id="w200">la</w> <w xml:id="w201">cuisine</w><w xml:id="w202">.</w> <milestone unit="Én" n="**é9a**"/><w xml:id="w204">Sans</w> <w xml:id="w205">hésiter</w><w xml:id="w206">,</w> <w xml:id="w207">il</w> <w xml:id="w208">plongea</w> <w xml:id="w209">le</w> <w xml:id="w210">bras</w> <w xml:id="w211">dans</w>

<lb n="14"/><w xml:id="w212">la</w> <w xml:id="w213">hotte</w> <w xml:id="w214">enfumée</w> <milestone unit="Én" n="**é9b**"/><w xml:id="w215">et</w> <w xml:id="w216">sortit</w> <lb n="16"/><w xml:id="w218">le</w> <w xml:id="w219">petit</w> <w xml:id="w220">couteau</w> <w xml:id="w221">à</w> <w xml:id="w222">manche</w> <w xml:id="w223">de</w> <w xml:id="w224">corne</w> <w xml:id="w225">qu'</w><w xml:id="w226">il</w> <w xml:id="w227">avait</w>

<lb n="15"/><w xml:id="w228">caché</w> <w xml:id="w229">là</w><w xml:id="w230">,</w> <milestone unit="Én" n="**é9c**"/><w xml:id="w231">lorsqu'</w><w xml:id="w232">il</w> <w xml:id="w234">était</w> <w xml:id="w235">enfant</w><w xml:id="w236">.</w> <milestone unit="Én" n="**él0a**"/><w xml:id="w237">Ses</w> <w xml:id="w238">yeux</w> <w xml:id="w239">brillèrent</w> <w xml:id="w240">de</w> <w xml:id="w241">joie</w> <milestone unit="Én" n="**él0b**"/><w xml:id="w242">et</w>

<lb n="16"/><w xml:id="w243">ses</w> <w xml:id="w244">parents</w> <w xml:id="w245">pleurèrent</w><w xml:id="w246">,</w> <lb n="18"/><milestone unit="Én" n="**él0c**"/><w xml:id="w248">parce</w> <w xml:id="w249">qu'</w><w xml:id="w250">ils</w> <w xml:id="w251">avaient</w> <w xml:id="w252">retrouvé</w> <w xml:id="w253">leur</w> <w xml:id="w254">fils</w><w xml:id="w255">.</w>

<lb n="17"/><milestone unit="Én" n="**élla**"/><w xml:id="w257">Ce</w> <w xml:id="w258">souvenir</w> <w xml:id="w259">fut</w> <w xml:id="w260">peut-être</w> <w xml:id="w261">suivi</w> <w xml:id="w262">par</w> <w xml:id="w263">d'</w><w xml:id="w264">autres</w><w xml:id="w265">,</w> <milestone unit="Én" n="**éllb**"/><w xml:id="w266">mais</w> <w xml:id="w267">l'</w><w xml:id="w268">Indien</w> <w xml:id="w269">ne</w> <w xml:id="w270">pouvait</w>

<lb n="18"/><w xml:id="w272">vivre</w> <w xml:id="w273">entre</w> <w xml:id="w274">quatre</w> <w xml:id="w275">murs</w> <milestone unit="Én" n="**é11c**"/><w xml:id="w276">et</w> <w xml:id="w277">un</w> <w xml:id="w278">jour</w> <w xml:id="w279">il</w> <w xml:id="w280">partit</w> <w xml:id="w281">à</w> <w xml:id="w282">la</w> <w xml:id="w283">recherche</w> <w xml:id="w285">de</w> <w xml:id="w286">son</w>

<lb n="19"/><w xml:id="w287">désert</w><w xml:id="w288">.</w> <milestone unit="Én" n="**é12a**"/><w xml:id="w289">Moi</w> <w xml:id="w290">je</w> <w xml:id="w291">voudrais</w> <w xml:id="w292">savoir</w> <milestone unit="Én" n="**é12b**"/><w xml:id="w293">ce</w> <w xml:id="w294">qu'</w><w xml:id="w295">il</w> <w xml:id="w296">ressentit</w> <w xml:id="w297">en</w> <w xml:id="w298">cet</w> <w xml:id="w300">instant</w> <w xml:id="w301">de</w>

<lb n="20"/><w xml:id="w302">vertige</w> <milestone unit="Én" n="**é12c**"/><w xml:id="w303">où</w> <w xml:id="w304">le</w> <w xml:id="w305">passé</w> <w xml:id="w306">et</w> <w xml:id="w307">le</w> <w xml:id="w308">présent</w> <w xml:id="w309">se</w> <w xml:id="w310">confondirent</w> <w xml:id="w312">;</w>

```
<milestone unit="Én" n="é12d"/><w xml:id="w313">moi</w> <w xml:id="w314">je</w> <w
xml:id="w315">voudrais</w>

<lb n="21"/><w xml:id="w316">savoir</w> <milestone unit="Én" n="é12e"/><w
xml:id="w317">si</w> <w xml:id="w318">le</w> <w xml:id="w319">fils</w> <w
xml:id="w320">perdu</w> <w xml:id="w321">renaquit</w> <w xml:id="w322">et</w> <w
xml:id="w323">mourut</w> <w xml:id="w324">en</w> <w xml:id="w326">cette</w> <w
xml:id="w327">extase</w><w xml:id="w328">,</w> <milestone unit="Én" n="é12f"/><w
xml:id="w329">ou</w> <w xml:id="w330">s'</w><w xml:id="w331">il</w>

<lb n="22"/><w xml:id="w332">parvint</w> <w xml:id="w333">à</w> <w
xml:id="w334">reconnaître</w><w xml:id="w335">,</w> <milestone unit="Én" n="é12g"/><w
xml:id="w336">ne</w> <w xml:id="w337">fût</w><w xml:id="w338">-</w><w
xml:id="w339">ce</w> <w xml:id="w340">qu'</w><w xml:id="w341">à</w> <w
xml:id="w343">la</w> <w xml:id="w344">manière</w> <w xml:id="w345">d'</w><w
xml:id="w346">un</w> <w xml:id="w347">nouveau-né</w> <w xml:id="w348">ou</w> <w
xml:id="w349">d'</w><w xml:id="w350">un</w>

<lb n="23"/><w xml:id="w351">chien</w><w xml:id="w352">,</w> <milestone unit="Én"
n="é12f_fin"/><w xml:id="w353">les</w> <w xml:id="w354">parents</w> <w
xml:id="w355">et</w> <w xml:id="w356">la</w> <w xml:id="w357">maison</w><w
xml:id="w358">.</w>
</p>
<!-- *{(Jorge Luis Borges, El Hacedor (J 960). Traduction de J.-M. Adam, Adam2005:203-204)}
 -->
</body>
</text>
</TEI>
```

Il existe d'autres schémas que ceux de la TEI pour représenter ce découpage en mots. Ainsi, la Bibliothèque nationale de France utilise le protocole ALTO dans son projet d'édition électronique Gallica2 :

> ALTO (analyzed layout and text object) stores layout information and OCR recognized text of pages of any kind of printed documents like books, journals and newspapers. ALTO is a standardized XML format to store layout and content information. It is designed to be used as an extension schema to METS (Metadata Encoding and Transmission Standard), where METS provides metadata and structural information while ALTO contains content and physical information. (http://bibnum.bnf.fr/numerisation/charte_technique_ocr_presse.pdf ; http://www.ccs-gmbh.com/alto/general.html ; http://www.loc.gov/ndnp/alto_1-1-041.xsd ; http://wiki.loria.fr/wiki/Les_standards_ALTO/TEI/METS).

Ce format est d'abord conçu pour lier les mots, sujets à une indexation plein texte, à la position du mot dans la page numérisée en format image. Ce format se présente comme une

extension de METS et est utilisé pour le projet NDNP (National Digital Newspaper Project) : http://www.loc.gov/ndnp/ . Voici un exemple de texte en format ALTO appliqué au dictionnaire Trévoux.

Format ALTO. (6.2.2d, exemple)

```
<?xml version="1.0" encoding="UTF-8" ?>
<alto>
<Description/>
<Styles>
<TextStyle ID="Arial_11." FONTFAMILY="Arial" FONTSIZE="11."/>
<TextStyle ID="Times New Roman_10._466_29" FONTFAMILY="Times New Roman"
FONTSIZE="10." FONTSPACING="29" FONTSCALING="
466"/>
</Styles>
<Layout>
<Page WIDTH="2480" HEIGHT="2480" ID="P0_DUT01_0082" PHYSICAL IMG NR="1">
<PrintSpace>
<TextBlock ID="B0" HPOS="64" WIDTH="124" VPOS="54" HEIGHT="74">
<TextLine BASELINE="124" ID="B0_L0" HPOS="72" WIDTH="101" VPOS="58" HEIGHT="66"
STYLEREFS="Times New
Roman_22._616_-17">
<String CONTENT="'%°5" STYLE="bold italic" HPOS="72" VPOS="58" WIDTH="101"/>
</TextLine>
</TextBlock>
<TextBlock ID="B3" HPOS="52" WIDTH="1174" VPOS="136" HEIGHT="4038">
<TextLine BASELINE="189" ID="B3_L0" HPOS="98" WIDTH="1115" VPOS="144"
HEIGHT="58" STYLEREFS="Times New
Roman_11._-4">
<String CONTENT=" blir" HPOS="98" VPOS="186" WIDTH="80"/>
<SP HPOS="178" VPOS="161" WIDTH="10"/>
<String CONTENT="un" HPOS="188" VPOS="161" WIDTH="43"/>
<SP HPOS="231" VPOS="148" WIDTH="10"/>
<String CONTENT="d" HPOS="241" VPOS="148" WIDTH="17"/>
<SP HPOS="258" VPOS="148" WIDTH="10"/>
<String CONTENT="Cqnftantinpple" HPOS="268" VPOS="148" WIDTH="273"/>
<String CONTENT="'txanf" HPOS="1111" VPOS="156" WIDTH="89" SUBS TYPE="HypPart1">
<HYP CONTENT="-" HPOS="1200" VPOS="174" WIDTH="13"/>
</String>
</TextLine>
<TextLine BASELINE="237" ID="B3_L1" HPOS="89" WIDTH="1123" VPOS="192"
HEIGHT="61" STYLEREFS="Times New Roman_11._1">
</TextLine>
</TextBlock>
</PrintSpace>
</Page>
</Layout>
</alto>
```

(Source : Falk, Ingrid. *Représentation et Stockage des données de la numérisation du dictionnaire Trévoux.* Fig. 2 – extrait du fichier ALTO.15 mars 2006. http://www.loria.fr/~falk/rapport_read_mar06.pdf)

Le format TEI a une portée plus large que le format ALTO. Si nécessaire, il est possible d'utiliser le format ALTO dans un document d'annotation associé à la codification TEI au même titre que l'image de la page numérisée. Un élément <*linkGrp*> pourrait, par exemple, contenir un ensemble de pointeurs dans le document TEI reliant les occurrences (éléments <*w*>) avec les éléments <*String*> du document ALTO.

Ce qui nous intéressera dans les paragraphes qui suivent, c'est la construction d'un fichier externe d'annotation (*StructureCompositionnelle.xml*) destiné à contenir le marquage analytique utilisé par Adam dans son livre. Voici le contenu du fichier d'analyse que nous commenterons dans les paragraphes qui suivront.



Document d'annotation structurelle
(*StructureCompositionnelle.xml*). (6.2.2e, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
<?oxygen RNGSchema="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng"
type="xml"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
   <teiHeader>
      <fileDesc>
         <titleStmt>
            <title>Analyse due à J.M. Adam du texte LE CAPTIF : Borges, El Hacedor. Traduction
               de J.-M. Adam</title>
         </titleStmt>
         <publicationStmt>
            <p>Publié par...</p>
         </publicationStmt>
         <sourceDesc>
            <p>...</p>
         </sourceDesc>
      </fileDesc>
   </teiHeader>
   <text>
      <body>
         <!-- Définition des catégories interprétatives -->
         <div type="Analyse" subtype="StructureCompositionnelle" xml:id="ana1">
            <interpGrp type="Unités_discursives">
               <interp xml:id="Énoncé">On considèrera comme énoncé...</interp>
               <interp xml:id="Phrase">On entendra par phrase typographique...</interp>
            </interpGrp>

            <interpGrp type="StructureCompositionnelle">
               <interp xml:id="plan_de_texte">Le plan du texte fait partie de la structure
                  compositionnelle qui organise la cohésion d'une suite linéaire de séquences
                  (Adam2005:chapitre 6).</interp>

               <interp xml:id="séquence">Les séquences sont des unités textuelles complexes,
```

composées d'un nombre limité de paquets de propositions-énoncés. Elles constituent des réseaux relationnels hiérarchiques formant des entités relativement autonomes présentant des agencements dits narratifs, argumentatif, explicatif, dialogal, etc. (Adam2005:chapitre 5). </interp>

<interp xml:id="SCséquence_narrative">La séquence narrative est... .</interp>

<interp xml:id="SCsituation_initiale">La situation initiale est un des composants de la séquence narrative...</interp>

<interp xml:id="SCnoeud">Le noeud est un des composants de la séquence narrative...</interp>

<interp xml:id="SCaction">L'action (ou la réaction) est un des composants de la séquence narrative...</interp>

<interp xml:id="SCdénouement">Le dénouement est un des composants de la séquence narrative...</interp>

<interp xml:id="SCsolution_finale">La solution finale est un des composants de la séquence narrative...</interp>

<interp xml:id="SCpériode">Selon Aristote, la période est une forme d'élocution qui renferme en elle-même un commencement et une fin, ainsi qu'une étendue qui se laisse embrasser d'un coup d'oeil» Rhétorique III, cité par Adam 2006, p. 141. </interp>

<interp xml:id="SCpériode_interprétative">Explication... . </interp>

<interp xml:id="SCpériode_narrative">Explication... . </interp>

<interp xml:id="SCréférence">Explication... Adam:2005:86-97. </interp>

<interp xml:id="I1">introduction</interp>
</interpGrp>

<!-- **Segmentation du texte analysé en énoncés** -->
<spanGrp xml:id="Seg1" type="Segmentation" ana="#Énoncé" xml:base="borges_adam.xml">
À Junín ou à Tapalqué, on raconte l'histoire suivante.
Un enfant disparut après un raid d'Indiens ;
on dit que les Indiens l'avaient enlevé.
Ses parents le cherchèrent inutilement ;
des années plus tard, un soldat qui venait de l'intérieur leur parla d'un Indien aux yeux couleur de ciel qui pouvait bien être leur fils.
Ils le rencontrèrent enfin (
la chronique a perdu les circonstances
et je ne veux pas inventer ce que je ne sais pas)
et ils crurent le reconnaître.
L'homme, marqué par le désert et la

```
      vie sauvage, ne comprenait déjà plus les mots de sa langue natale,</span>
    <span from="#w129" to="#w143" xml:id="é5b"> mais, indifférent et docile, il se
      laissa conduire à la maison. </span>
    <span from="#w144" to="#w150" xml:id="é6a">Il s'arrêta sur le seuil,</span>
    <span from="#w151" to="#w160" xml:id="é6b">peut-être parce que les autres s'y
      arrêtèrent.</span>
    <span from="#w161" to="#w165" xml:id="é7a">Il regarda la porte,</span>
    <span from="#w166" to="#w174" xml:id="é7b">comme s'il ne la comprenait
      pas.</span>
    <span from="#w175" to="#w181" xml:id="é8a">Soudain, il baissa la tête,"></span>
    <span from="#w182" to="#w185" xml:id="é8b">poussa un cri,"></span>
    <span from="#w186" to="#w196" xml:id="é8c">traversa en courant le corridor et
      les deux vastes cours"></span>
    <span from="#w197" to="#w202" xml:id="é8d">et pénétra dans la cuisine."></span>
    <span from="#w204" to="#w214" xml:id="é9a">Sans hésiter, il plongea le bras dans
      la hotte enfumée </span>
    <span from="#w215" to="#w230" xml:id="é9b">et sortit le petit couteau à manche
      de corne qu'il avait caché là,</span>
    <span from="#w231" to="#w236" xml:id="é9c">lorsqu'il était enfant.</span>
    <span from="#w237" to="#w241" xml:id="é10a">Ses yeux brillèrent de joie</span>
    <span from="#w242" to="#w246" xml:id="é10b">et ses parents pleurèrent, </span>
    <span from="#w248" to="#w255" xml:id="é10c">parce qu'ils avaient retrouvé leur
      fils.</span>
    <span from="#w257" to="#w265" xml:id="é11a">Ce souvenir fut peut-être suivi par
      d'autres,</span>
    <span from="#w266" to="#w275" xml:id="é11b">mais l'Indien ne pouvait vivre entre
      quatre murs</span>
    <span from="#w276" to="#w288" xml:id="é11c">et un jour il partit à la recherche
      de son désert.</span>
    <span from="#w289" to="#w292" xml:id="é12a">Moi je voudrais savoir</span>
    <span from="#w293" to="#w302" xml:id="é12b">ce qu'il ressentit en cet instant de
      vertige</span>
    <span from="#w303" to="#w312" xml:id="é12c">où le passé et le présent se
      confondirent ; </span>
    <span from="#w313" to="#w316" xml:id="é12d">moi je voudrais savoir</span>
    <span from="#w317" to="#w328" xml:id="é12e">si le fils perdu renaquit et mourut
      en cette extase,</span>
    <span from="#w329" to="#w335" xml:id="é12f">ou s'il parvint à
      reconnaître,</span>
    <span from="#w336" to="#w352" xml:id="é12g">ne fût-ce qu'à la manière d'un
      nouveau-né ou d'un chien,</span>
    <span from="#w353" to="#w358" xml:id="é12f_fin">les parents et la maison.</span>
</spanGrp>

<!-- Segmentation du texte analysé en phrases -->
<spanGrp xml:id="Seg2" type="Segmentation" ana="#Phrase">
    <span from="#é1" xml:id="P1"/>
    <span from="#é2a" to="#é2b" xml:id="P2"/>
    <span from="#é3a" to="#é3b" xml:id="P3"/>
    <span from="#é4a" to="#é4d" xml:id="P4"/>
    <span from="#é5a" to="#é5b" xml:id="P5"/>
    <span from="#é6a" to="#é6b" xml:id="P6"/>
    <span from="#é7a" to="#é7b" xml:id="P7"/>
    <span from="#é8a" to="#é8d" xml:id="P8"/>
    <span from="#é9a" to="#é9c" xml:id="P9"/>
    <span from="#é10a" to="#é10c" xml:id="P10"/>
    <span from="#é11a" to="#é11c" xml:id="P11"/>
```

```
   <span from="#é12a" to="#é12f_fin" xml:id="P12"/>
</spanGrp>

<!-- Blocs de l'analyse compositionnelle -->

<div type="StructureCompositionnelle" ana="#SCsequence_narrative"
   xml:id="séquence_narrative_1" n="séquence narrative">
   <div type="StructureCompositionnelle" ana="#SCsituation_initiale" xml:id="Pn1">
      <span type="StructureCompositionnelle" from="#é2a" to="#é3a"
         xml:id="Pn1.é2a-é3a"/>
   </div>
   <div type="StructureCompositionnelle" xml:id="intrigue_1">
      <div type="StructureCompositionnelle" ana="#SCnoeud" xml:id="Pn2">
         <span type="StructureCompositionnelle" from="#é3b" to="#é4a"
            xml:id="Pn2.é3b-é4a"/>
         <span type="StructureCompositionnelle" from="#é4d" xml:id="Pn2.é4d"/>
      </div>
      <div type="StructureCompositionnelle" ana="#SCaction" xml:id="Pn3">
         <span type="StructureCompositionnelle" from="#P5" to="#P7"
            xml:id="Pn3.P5-P7">(Ré)Action</span>
      </div>
      <div type="StructureCompositionnelle" ana="#SCdénouement" xml:id="Pn4">
         <span type="StructureCompositionnelle" from="#P8" to="#P9"
            xml:id="Pn4.P8-P9"/>
      </div>
   </div>
   <div type="StructureCompositionnelle" ana="#SCsolution_finale" xml:id="Pn5">
      <span type="StructureCompositionnelle" from="#P10" xml:id="Pn5.P10"/>
   </div>
</div>

<div type="StructureCompositionnelle" ana="#SCsequence_narrative"
   xml:id="entrée-préface">
   <span type="StructureCompositionnelle" from="#é1" xml:id="Pn0">Cadre
      médiatif</span>
   <span type="StructureCompositionnelle" from="#é4b" to="#é4c" xml:id="Pn0a"
      >Évaluation commentative</span>
</div>

<div type="StructureCompositionnelle" ana="#SCpériode_argumentative"
   xml:id="P11_argumentative">
   <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-proposition_p"
      >premier argument </span>
   <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-proposition_q"
      >second argument</span>
   <span type="StructureCompositionnelle" from="#é11c"
      xml:id="é11c-conclusion_non_c">renversement de la conclusion implicite du
      retour définitif à la maison</span>
</div>

<div type="StructureCompositionnelle" ana="#SCpériode_narrative"
   xml:id="P11_narrative">
   <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-Pn1">Situation
      initiale Pn1</span>
   <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-Pn2">Noeud
      Pn2</span>
   <span type="StructureCompositionnelle" from="#é11c" xml:id="é11c-
```

```
Pn4">Dénouement
                    Pn4</span>
                </div>

                <div type="StructureCompositionnelle" ana="#SCpériode"
xml:id="simple_période_P11">
                    <alt mode="incl" targets="#P11_argumentative #P11_narrative" weights="0.5 0.5"/>
                </div>

                <div type="StructureCompositionnelle" ana="#SCpériode" xml:id="évaluation_finale">
                    <span type="StructureCompositionnelle" from="#P12" xml:id="PnΩ">Évaluation
                        finale. «Cette prose périodique dominée par le rythme contribue au
                        glissement de genre du récit factuel au récit poétique.» (Adam 2005:
                        211)</span>
                </div>

                <!-- Bloc supérieur : plan du texte -->

                <div type="StructureCompositionnelle" ana="#SCplan_de_texte"
                    xml:id="plan_de_texte_du_Captif">
                    <ab>
                        <ptr target="#entrée-préface"/>
                        <ptr target="#séquence_ narrative_1"/>
                        <ptr target="#simple_période_P11"/>
                        <ptr target="#évaluation_finale"/>
                    </ab>
                </div>

            </div>
        </body>
    </text>
</TEI>
```

Le fichier commence par un entête TEI. Vient ensuite le corps du texte (<*body*>) avec une division d'analyse : * *

Dans l'analyse, on trouve d'abord des <*interpGrp*> qui rassemblent les catégories interprétatives correspondant aux valeurs des attributs *ana* utilisés dans l'analyse. Vient ensuite un groupe d'empans textuels (<*spanGrp*>) décrivant les énoncés découpés par Adam. Pour faciliter la lecture, le contenu des balises <*span*> reprendra le texte intégral dépouillé de son balisage.

L'élément <spanGrp>, qui encadre la segmentation en propositions-énoncées, contient un certain nombre d'attributs.

- xml:id="Seg1" : identificateur de la segmentation ;
- type="Segmentation" : type des <*span*> ;

- ana="#Énoncé" : renvoi vers un élément <*interp*> expliquant la sémantique du découpage ;
- xml:base="borges_adam.xml" : adresse du document sur lequel porte le découpage.

Outre le découpage du texte en 36 propositions-énoncés, l'analyse de Jean-Michel Adam fait aussi référence au découpage du texte en 12 phrases typographiques numérotées. La délimitation de ces phrases peut s'opérer en référence aux identificateurs de mots. Cependant, dans l'exposé d'Adam, les phrases sont plutôt utilisées comme des regroupements de propositions-énoncés. Voilà pourquoi on a utilisé les identificateurs d'énoncés pour représenter ces phrases dans les balises <*span*> de la deuxième segmentation. Il est à noter qu'Adam, avec P11, rassemble les énoncés d'une phrase en espagnol que le traducteur a décidé de rendre en deux phrases en français.

Viennent ensuite les blocs d'*analyse compositionnelle* que nous explicitons dans les paragraphes qui suivent.

6.2.2.1 Analyse de la structure compositionnelle du texte

Ces divers segments phrastiques ou propositionnels sont organisés à des fins d'analyse en plusieurs regroupements périodiques et un regroupement séquentiel. La structure compositionnelle, plus spécifiquement le plan du texte, est présentée sous la forme d'un arbre. On pourrait donc emprunter le formalisme TEI des <*eTree*> (*embedding tree*) pour représenter la structure sous forme de graphe. Cependant, pour éviter de multiplier les formalismes, nous allons plutôt réutiliser les éléments <*span*> à l'intérieur d'une structure classique d'emboitement (élément <*div*>). Les valeurs de l'attribut *ana* seront employées pour pointer vers la définition, dans des éléments <*interp*>, des divers composants de la structure compositionnelle. Les valeurs de ces attributs seront préfixées des lettres *SC* pour indiquer qu'elles font partie d'un vocabulaire associé à l'analyse de la structure compositionnelle du texte.

Le processus d'analyse procède d'un va-et-vient entre la reconnaissance d'éléments macrostructurels et leur décomposition en structures plus fines jusqu'aux propositions-énoncés. La couche d'annotation la plus proche du texte catégorise des segments en fonction d'un paradigme d'organisation du texte de niveau supérieur. Ainsi, en annonçant que le premier paragraphe se présente comme une *séquence narrative complète*, on s'attend à ce que

les énoncés soient caractérisés en termes de composantes narratives. C'est ce que traduisent les éléments <*span*> et <*div*> qui suivent.

```
<div type="StructureCompositionnelle" ana="#SCsequence_narrative"
   xml:id="séquence_narrative_1" n="séquence narrative">
   <div type="StructureCompositionnelle" ana="#SCsituation_initiale" xml:id="Pn1">
      <span type="StructureCompositionnelle" from="#é2a" to="#é3a"
         xml:id="Pn1.é2a-é3a"/>
   </div>
   <div type="StructureCompositionnelle" xml:id="intrigue_1">
      <div type="StructureCompositionnelle" ana="#SCnoeud" xml:id="Pn2">
         <span type="StructureCompositionnelle" from="#é3b" to="#é4a"
            xml:id="Pn2.é3b-é4a"/>
         <span type="StructureCompositionnelle" from="#é4d" xml:id="Pn2.é4d"/>
      </div>
      <div type="StructureCompositionnelle" ana="#SCaction" xml:id="Pn3">
         <span type="StructureCompositionnelle" from="#P5" to="#P7"
            xml:id="Pn3.P5-P7">(Ré)Action</span>
      </div>
      <div type="StructureCompositionnelle" ana="#SCdénouement" xml:id="Pn4">
         <span type="StructureCompositionnelle" from="#P8" to="#P9"
            xml:id="Pn4.P8-P9"/>
      </div>
   </div>
   <div type="StructureCompositionnelle" ana="#SCsolution_finale" xml:id="Pn5">
      <span type="StructureCompositionnelle" from="#P10" xml:id="Pn5.P10"/>
   </div>
</div>
```

La composition structurelle emprunte donc ici la forme classique de l'emboitement des éléments TEI <*div*>. Le contenu textuel des blocs est constitué de références à des segments dont les pointeurs, une fois évalués, conduiront finalement à des empans textuels dans le document analysé contenu dans une ressource externe. Les valeurs de l'attribut *ana* renvoient à des explications sur l'interprétation de chaque structure compositionnelle. On réfère ici à des éléments <*interp*>. Mais, on aurait pu aussi référer à un système catégoriel défini explicitement dans une structure de traits.

Poursuivons, avec le même formalisme, l'analyse structurelle des autres énoncés.

La séquence narrative est précédée d'un premier énoncé qualifié d'*entrée-préface*. Une parenthèse dans la quatrième phrase est vue par Adam comme un renforcement du cadre médiatif de l'entrée-préface dont voici la description TEI.

```
<div type="StructureCompositionnelle" ana="#SCsequence_narrative"
   xml:id="entrée-préface">
   <span type="StructureCompositionnelle" from="#é1" xml:id="Pn0">Cadre
      médiatif</span>
   <span type="StructureCompositionnelle" from="#é4b" to="#é4c" xml:id="Pn0a"
      >Évaluation commentative</span>
```

```
</div>
```

Poursuivons la représentation TEI de l'analyse d'Adam pour remonter progressivement vers le plan de texte. La phrase P11 est qualifiée de *période ternaire* interprétable de deux points de vue, soit une *période argumentative*, soit une *séquence narrative incomplète* qualifiée de *récit avorté*. Chacun de ces points de vue sera d'abord décrit indépendamment. Nous utiliserons la balise TEI <*alt*> pour réunir ces deux interprétations.

```
<div type="StructureCompositionnelle" ana="#SCpériode_argumentative"
   xml:id="P11_argumentative">
   <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-proposition_p"
      >premier argument </span>
   <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-proposition_q"
      >second argument</span>
   <span type="StructureCompositionnelle" from="#é11c"
      xml:id="é11c-conclusion_non_c">renversement de la conclusion implicite du
      retour définitif à la maison</span>
</div>

<div type="StructureCompositionnelle" ana="#SCpériode_narrative"
   xml:id="P11_narrative">
   <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-Pn1">Situation
      initiale Pn1</span>
   <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-Pn2">Noeud
      Pn2</span>
   <span type="StructureCompositionnelle" from="#é11c" xml:id="é11c-Pn4">Dénouement
      Pn4</span>
</div>

<div type="StructureCompositionnelle" ana="#SCpériode" xml:id="simple_période_P11">
   <alt mode="incl" targets="#P11_argumentative #P11_narrative" weights="0.5 0.5"/>
</div>
```

La TEI présente l'élément <*alt*> comme une façon de rendre compte de l'incertitude de la transcription des passages d'un manuscrit et de la multiplicité possible de ces transcriptions. Mais, il ne sous semble pas abusif d'utiliser ces éléments à un niveau d'interprétation plus abstrait que le décryptage des mots d'un manuscrit. L'élément <*alt*> contient un attribut *mode* qui permet d'indiquer si la disjonction est inclusive ou exclusive d'un point de vue logique. On peut aussi utiliser l'attribut *weights* pour marquer un pourcentage de vraisemblance de chacun des termes de la disjonction. Dans son livre, Adam se contente de qualifier cette phrase de *période narrativo-argumentative*. Nous avons donc distribué les poids de façon égalitaire entre les deux analyses.

On passe finalement à l'analyse de la dernière phrase qualifiée d'*évaluation finale de récit*.

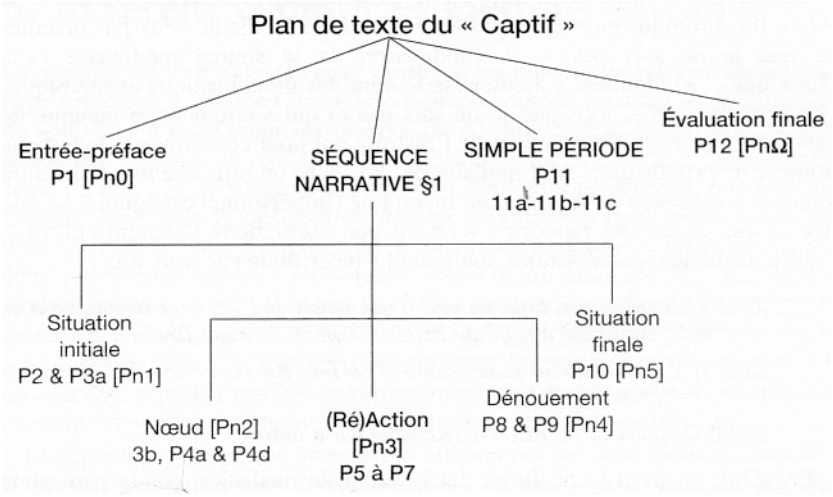
```
<div type="StructureCompositionnelle" ana="#SCpériode" xml:id="évaluation_finale">
   <span type="StructureCompositionnelle" from="#P12" xml:id="PnΩ">Évaluation
      finale. «Cette prose périodique dominée par le rythme contribue au
      glissement de genre du récit factuel au récit poétique.» (Adam 2005:
      211)</span>
</div>
```

Rassemblant les diverses analyses partielles déjà présentées, voici le plan du texte qui présente l'organisation des séquences et des périodes pour l'ensemble du récit.

```
<div type="StructureCompositionnelle" ana="#SCplan_de_texte"
   xml:id="plan_de_texte_du_Captif">
   <ab>
      <ptr target="#entrée-préface"/>
      <ptr target="#séquence_ narrative_1"/>
      <ptr target="#simple_période_P11"/>
      <ptr target="#évaluation_finale"/>
   </ab>
</div>
```

La structure hiérarchique s'exprime ici par un élément <*div*> analysé comme *SCplan_de_texte* et qui englobe les structures précédentes insérées par référence au moyen des balises <*ptr*> (pointeur). La présence de l'élément *bloc arbitraire* <*ab*>, qui a un statut similaire à l'élément paragraphe <*p*>, ne représente pas un emboitement dans le plan de texte. Il s'agit simplement d'une contrainte syntaxique de la TEI qui ne permet pas d'avoir un élément <*ptr*> directement sous un élément <*div*>. L'élément <*ptr*> n'ajoute pas de nouveaux emboitements. Il s'agit simplement d'une référence à une structure existante qui pourrait tout aussi bien prendre la place du pointeur.

Comme on l'a fait pour l'analyse fonctionnelle de la phrase en termes de thèmes et de rhèmes, on pourrait construire une représentation du plan de texte sous forme de graphe traduisant les éléments structuraux que nous venons de présenter. Une vue partielle du plan est d'ailleurs présentée par Adam dans son schéma *Plan de texte du « Captif »* (Adam2005:211).



Dans le graphe que nous pouvons construire à partir de l'analyse présentée, chacun des nœuds correspond à un élément <*div*> ou <*span*>, les arcs entre nœuds traduisant l'emboitement des éléments ou le parcours des pointeurs pour les éléments <*alt*> et <*ptr*>. On peut produire automatiquement ce graphe au moyen de la feuille de style XSLT *StructureCompositionnelle-graph.xsl* reproduite en annexe dans l'encadré 6.2.2f.

Feuille de style transformant en graphe l'analyse du plan de texte du récit de Borges (fichier *StructureCompositionnelle.xml*) (6.2.2f, exemple)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
   xmlns:TEI="http://www.tei-c.org/ns/1.0">
   <xsl:strip-space elements="*"/>
   <!--StructureCompositionnelle-graph
      transforme une annotation structurelle de type StructureCompositionnelle
      (plan de texte) en un graphe orientée représentant cette structure -->
   <xsl:output method="xml" indent="yes"/>
   <xsl:variable name="newline">
      <xsl:text>
</xsl:text>
   </xsl:variable>

   <xsl:template match="/">
      <xsl:call-template name="teiHeader"/>
   </xsl:template>

   <xsl:template name="teiHeader">
      <xsl:apply-templates mode="teiHeader"/>
   </xsl:template>

   <xsl:template match="title" mode="teiHeader">
      <xsl:copy>
```

```
        <xsl:value-of select="."/>
        <xsl:value-of select="$newline"/>
        <xsl:text> Présentation sous forme de graphe.</xsl:text>
    </xsl:copy>
</xsl:template>

<xsl:template match="@*|node()" mode="teiHeader">
    <xsl:copy>
        <xsl:apply-templates select="@*" mode="teiHeader"/>
        <xsl:choose>
            <xsl:when test="TEI:body">
                <xsl:apply-templates mode="body" select="."/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:apply-templates mode="teiHeader"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:copy>
</xsl:template>

<xsl:template match="TEI:body" mode="body">
    <xsl:copy>
        <xsl:for-each select="@*">
            <xsl:copy/>
        </xsl:for-each>
        <xsl:apply-templates mode="body"/>
    </xsl:copy>
</xsl:template>

<xsl:template match="TEI:div[@type='Analyse' and @subtype='StructureCompositionnelle']"
    mode="body">
    <!-- interp -->
    <xsl:for-each select="TEI:interpGrp">
        <xsl:apply-templates mode="teiHeader" select="."/>
    </xsl:for-each>
    <xsl:value-of select="$newline"/>

    <!-- Extraction des span -->
    <xsl:for-each select="/descendant-or-self::TEI:spanGrp">
        <xsl:apply-templates mode="teiHeader" select="."/>
    </xsl:for-each>
    <xsl:value-of select="$newline"/>

    <xsl:for-each select="/descendant-or-self::TEI:span">
        <xsl:if test="not(ancestor::TEI:spanGrp)">
            <xsl:apply-templates mode="teiHeader" select="."/>
        </xsl:if>
    </xsl:for-each>
    <xsl:value-of select="$newline"/>

    <!-- graph -->
    <xsl:variable name="graph_nom">
        <xsl:value-of select="@xml:id"/>
    </xsl:variable>
    <xsl:element name="graph">
        <xsl:attribute name="type">
            <xsl:text>directed</xsl:text>
```

```
            </xsl:attribute>
            <xsl:attribute name="xml:id">
               <xsl:value-of select="$graph_nom"/>
            </xsl:attribute>
            <!-- label -->
            <xsl:element name="label">
               <xsl:value-of select="@subtype"/>
            </xsl:element>
            <xsl:value-of select="$newline"/>

            <!-- création des noeuds de StructureCompositionnelle -->
            <xsl:for-each
               select="descendant-or-self::TEI:div[@type='StructureCompositionnelle']|descendant-or-
self::TEI:span[@type='StructureCompositionnelle']">
               <xsl:call-template name="noeud-SC">
                  <xsl:with-param name="graph_nom">
                     <xsl:value-of select="$graph_nom"/>
                  </xsl:with-param>
               </xsl:call-template>
            </xsl:for-each>

            <!-- arc -->
            <xsl:call-template name="arcs-SC">
               <xsl:with-param name="graph_nom">
                  <xsl:value-of select="@xml:id"/>
               </xsl:with-param>
            </xsl:call-template>
         </xsl:element>
      </xsl:template>

      <xsl:template name="noeud-SC">
         <xsl:param name="graph_nom"/>
         <xsl:element name="node">
            <!-- node -->
            <xsl:attribute name="xml:id">
               <xsl:value-of select="$graph_nom"/>
               <xsl:text>_</xsl:text>
               <xsl:value-of select="@xml:id"/>
            </xsl:attribute>
            <xsl:if test="@from">
               <xsl:attribute name="value">
                  <xsl:value-of select="@xml:id"/>
               </xsl:attribute>
            </xsl:if>
            <xsl:if test="@n">
               <xsl:attribute name="n">
                  <xsl:value-of select="@n"/>
               </xsl:attribute>
            </xsl:if>
            <xsl:element name="label">
               <!-- label -->
               <xsl:if test="attribute::ana">
                  <xsl:attribute name="ana">
                     <xsl:value-of select="@ana"/>
                  </xsl:attribute>
               </xsl:if>
               <xsl:if test="attribute::n">
```

```
                <xsl:value-of select="normalize-space(@n)"/>
                <xsl:text> </xsl:text>
              </xsl:if>
              <xsl:text>[</xsl:text>
              <xsl:value-of select="@xml:id"/>
              <xsl:text>]</xsl:text>
            </xsl:element>
          </xsl:element>
          <!--<xsl:value-of select="$newline"/>-->
        </xsl:template>

        <xsl:template name="arcs-SC">
          <!-- création des arcs de StructureCompositionnelle -->
          <xsl:param name="graph_nom"/>
          <xsl:for-each select="TEI:div[@type='StructureCompositionnelle']">
            <!-- tete -->
            <xsl:variable name="tete">
              <xsl:value-of select="@xml:id"/>
            </xsl:variable>
            <!-- arcs vers les span et div fils -->
            <xsl:for-each select="TEI:span|TEI:div">
              <xsl:call-template name="arc-SC">
                <xsl:with-param name="graph_nom">
                  <xsl:value-of select="$graph_nom"/>
                </xsl:with-param>
                <xsl:with-param name="de">
                  <xsl:value-of select="$tete"/>
                </xsl:with-param>
                <xsl:with-param name="à">
                  <xsl:value-of select="@xml:id"/>
                </xsl:with-param>
              </xsl:call-template>
            </xsl:for-each>
            <!-- arcs vers les span et div fils référés par les ptr-->
            <xsl:for-each select="TEI:ab/TEI:ptr">
              <xsl:call-template name="arc-SC">
                <xsl:with-param name="graph_nom">
                  <xsl:value-of select="$graph_nom"/>
                </xsl:with-param>
                <xsl:with-param name="de">
                  <xsl:value-of select="$tete"/>
                </xsl:with-param>
                <xsl:with-param name="à">
                  <xsl:value-of select="substring-after(@target,'#')"/>
                </xsl:with-param>
              </xsl:call-template>
            </xsl:for-each>
            <!-- arcs vers les span et div fils référés par les alt-->
            <xsl:for-each select="TEI:alt">
              <xsl:call-template name="alt">
                <xsl:with-param name="graph_nom">
                  <xsl:value-of select="$graph_nom"/>
                </xsl:with-param>
                <xsl:with-param name="de">
                  <xsl:value-of select="$tete"/>
                </xsl:with-param>
                <xsl:with-param name="pointeurs">
```

```
                    <xsl:value-of select=" concat(normalize-space(@targets), ' ')"/>
                </xsl:with-param>
                <xsl:with-param name="poids">
                    <xsl:value-of select=" concat(normalize-space(@weights), ' ')"/>
                </xsl:with-param>
            </xsl:call-template>
        </xsl:for-each>

        <!-- appel récursif sur les div fils -->
        <xsl:call-template name="arcs-SC">
            <xsl:with-param name="graph_nom">
                <xsl:value-of select="$graph_nom"/>
            </xsl:with-param>
        </xsl:call-template>
    </xsl:for-each>
</xsl:template>

<xsl:template name="alt">
    <!--  Définit un arc du noeud $de aux noeuds $pointeurs -->
    <xsl:param name="graph_nom"/>
    <xsl:param name="de"/>
    <xsl:param name="pointeurs"/>
    <xsl:param name="poids" select="''"/>
    <xsl:choose>
        <xsl:when test="$pointeurs!=''">
            <xsl:variable name="premier_pointeur" select="substring-before($pointeurs, ' ')"/>
            <xsl:variable name="autres_pointeurs" select="substring-after($pointeurs, ' ')"/>
            <xsl:variable name="premier_poids" select="substring-before($poids, ' ')"/>
            <xsl:variable name="autres_poids" select="substring-after($poids, ' ')"/>
            <xsl:call-template name="arc-SC">
                <xsl:with-param name="graph_nom">
                    <xsl:value-of select="$graph_nom"/>
                </xsl:with-param>
                <xsl:with-param name="de">
                    <xsl:value-of select="$de"/>
                </xsl:with-param>
                <xsl:with-param name="à">
                    <xsl:value-of select="substring-after($premier_pointeur,'#')"/>
                </xsl:with-param>
                <xsl:with-param name="label">
                    <xsl:choose>
                        <xsl:when test="$premier_poids">
                            <xsl:value-of
                                select="concat('contient (alt ',@mode,' poids ',$premier_poids,')')"
                            />
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:value-of select="concat('contient (alt ',@mode,')')"/>
                        </xsl:otherwise>
                    </xsl:choose>
                </xsl:with-param>
            </xsl:call-template>
            <xsl:call-template name="alt">
                <xsl:with-param name="graph_nom">
                    <xsl:value-of select="$graph_nom"/>
                </xsl:with-param>
                <xsl:with-param name="de">
```

```
                        <xsl:value-of select="$de"/>
                    </xsl:with-param>
                    <xsl:with-param name="pointeurs">
                        <xsl:value-of select="$autres_pointeurs"/>
                    </xsl:with-param>
                    <xsl:with-param name="poids">
                        <xsl:value-of select="$autres_poids"/>
                    </xsl:with-param>
                </xsl:call-template>
            </xsl:when>
        </xsl:choose>
    </xsl:template>

    <xsl:template name="arc-SC">
        <!--  Définit un arc du noeud $de au noeud $à -->
        <xsl:param name="graph_nom"/>
        <xsl:param name="de"/>
        <xsl:param name="à"/>
        <xsl:param name="label" select="'contient'"/>
        <xsl:element name="arc">
            <!-- arc -->
            <xsl:attribute name="from">
                <xsl:text>#</xsl:text>
                <xsl:value-of select="$graph_nom"/>
                <xsl:text>_</xsl:text>
                <xsl:value-of select="$de"/>
            </xsl:attribute>
            <xsl:attribute name="to">
                <xsl:text>#</xsl:text>
                <xsl:value-of select="$graph_nom"/>
                <xsl:text>_</xsl:text>
                <xsl:value-of select="$à"/>
            </xsl:attribute>
            <xsl:element name="label">
                <!-- label -->
                <xsl:value-of select="$label"/>
            </xsl:element>
        </xsl:element>
        <!--<xsl:value-of select="$newline"/>-->
    </xsl:template>
    <xsl:template match="node()" mode="body">
        <xsl:apply-templates mode="body"/>
    </xsl:template>

    <xsl:template name="segmentation">
        <xsl:value-of select="$newline"/>
        <xsl:apply-templates mode="teiHeader" select="ab[@type='Segmentation']"/>
        <xsl:value-of select="$newline"/>
    </xsl:template>

</xsl:stylesheet>
```

On trouvera, dans l'encadré suivant, la description en format TEI du graphe représentant le plan de texte du récit de Borges.



Graphe d'annotation structurelle (*StructureCompositionnelle-graph.xml*). (6.2.2g, exemple)

```
<?xml version="1.0" encoding="utf-8"?><?oxygen RNGSchema="http://www.tei-
c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="xml"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Analyse due à J.M. Adam du texte LE CAPTIF : Borges, El Hacedor. Traduction
            de J.-M. Adam</title>
      </titleStmt>
      <publicationStmt>
        <p>Publié par...</p>
      </publicationStmt>
      <sourceDesc>
        <p>...</p>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <interpGrp type="Unités_discursives">
        <interp xml:id="Énoncé">On considèrera comme énoncé...</interp>
        <interp xml:id="Phrase">On entendra par phrase typographique...</interp>
      </interpGrp>
      <interpGrp type="StructureCompositionnelle">
        <interp xml:id="plan_de_texte">Le plan du texte fait partie de la structure
            compositionnelle qui organise la cohésion d'une suite linéaire de séquences
            (Adam2005:chapitre 6).</interp>
        <interp xml:id="séquence">Les séquences sont des unités textuelles complexes,
            composées d'un nombre limité de paquets de propositions-énoncés. Elles
            constituent des réseaux relationnels hiérarchiques formant des entités
            relativement autonomes présentant des agencements dits narratifs,
            argumentatif, explicatif, dialogal, etc. (Adam2005:chapitre 5). </interp>
        <interp xml:id="SCséquence_narrative">La séquence narrative est... .</interp>
        <interp xml:id="SCsituation_initiale">La situation initiale est un des
            composants de la séquence narrative...</interp>
        <interp xml:id="SCnoeud">Le noeud est un des composants de la séquence
            narrative...</interp>
        <interp xml:id="SCaction">L'action (ou la réaction) est un des composants de la
            séquence narrative...</interp>
        <interp xml:id="SCdénouement">Le dénouement est un des composants de la séquence
            narrative...</interp>
        <interp xml:id="SCsolution_finale">La solution finale est un des composants de
            la séquence narrative...</interp>
        <interp xml:id="SCpériode">Selon Aristote, la période est une forme d'élocution
            qui renferme en elle-même un commencement et une fin, ainsi qu'une étendue
            qui se laisse embrasser d'un coup d'oeil» Rhétorique III, cité par Adam
            2006, p. 141. </interp>
        <interp xml:id="SCpériode_interprétative">Explication... . </interp>
        <interp xml:id="SCpériode_narrative">Explication... . </interp>
        <interp xml:id="SCréférence">Explication... Adam:2005:86-97. </interp>
        <interp xml:id="I1">introduction</interp>
      </interpGrp>
```

```
<spanGrp xml:id="Seg1" type="Segmentation" ana="#Énoncé"
xml:base="borges_adam.xml">
<span from="#w2" to="#w13" xml:id="é1">À Junín ou à Tapalqué, on raconte
l'histoire suivante.</span>
<span from="#w14" to="#w23" xml:id="é2a">Un enfant disparut après un raid
d'Indiens ;</span>
<span from="#w24" to="#w33" xml:id="é2b">on dit que les Indiens l'avaient
enlevé.</span>
<span from="#w34" to="#w39" xml:id="é3a">Ses parents le cherchèrent inutilement
;</span>
<span from="#w40" to="#w71" xml:id="é3b">des années plus tard, un soldat qui
venait de l'intérieur leur parla d'un Indien aux yeux couleur de ciel qui
pouvait bien être leur fils.</span>
<span from="#w72" to="#w76" xml:id="é4a">Ils le rencontrèrent enfin ( </span>
<span from="#w77" to="#w83" xml:id="é4b">la chronique a perdu les
circonstances</span>
<span from="#w84" to="#w97" xml:id="é4c">et je ne veux pas inventer ce que je ne
sais pas) </span>
<span from="#w98" to="#w103" xml:id="é4d">et ils crurent le reconnaitre.</span>
<span from="#w104" to="#w128" xml:id="é5a">L'homme, marqué par le désert et la
vie sauvage, ne comprenait déjà plus les mots de sa langue natale,</span>
<span from="#w129" to="#w143" xml:id="é5b"> mais, indifférent et docile, il se
laissa conduire à la maison. </span>
<span from="#w144" to="#w150" xml:id="é6a">Il s'arrêta sur le seuil,</span>
<span from="#w151" to="#w160" xml:id="é6b">peut-être parce que les autres s'y
arrêtèrent.</span>
<span from="#w161" to="#w165" xml:id="é7a">Il regarda la porte,</span>
<span from="#w166" to="#w174" xml:id="é7b">comme s'il ne la comprenait
pas.</span>
<span from="#w175" to="#w181" xml:id="é8a">Soudain, il baissa la tête,"&gt;</span>
<span from="#w182" to="#w185" xml:id="é8b">poussa un cri,"&gt;</span>
<span from="#w186" to="#w196" xml:id="é8c">traversa en courant le corridor et
les deux vastes cours"&gt;</span>
<span from="#w197" to="#w202" xml:id="é8d">et pénétra dans la cuisine."&gt;</span>
<span from="#w204" to="#w214" xml:id="é9a">Sans hésiter, il plongea le bras dans
la hotte enfumée </span>
<span from="#w215" to="#w230" xml:id="é9b">et sortit le petit couteau à manche
de corne qu'il avait caché là,</span>
<span from="#w231" to="#w236" xml:id="é9c">lorsqu'il était enfant.</span>
<span from="#w237" to="#w241" xml:id="é10a">Ses yeux brillèrent de joie</span>
<span from="#w242" to="#w246" xml:id="é10b">et ses parents pleurèrent, </span>
<span from="#w248" to="#w255" xml:id="é10c">parce qu'ils avaient retrouvé leur
fils.</span>
<span from="#w257" to="#w265" xml:id="é11a">Ce souvenir fut peut-être suivi par
d'autres,</span>
<span from="#w266" to="#w275" xml:id="é11b">mais l'Indien ne pouvait vivre entre
quatre murs</span>
<span from="#w276" to="#w288" xml:id="é11c">et un jour il partit à la recherche
de son désert.</span>
<span from="#w289" to="#w292" xml:id="é12a">Moi je voudrais savoir</span>
<span from="#w293" to="#w302" xml:id="é12b">ce qu'il ressentit en cet instant de
vertige</span>
<span from="#w303" to="#w312" xml:id="é12c">où le passé et le présent se
confondirent ; </span>
<span from="#w313" to="#w316" xml:id="é12d">moi je voudrais savoir</span>
<span from="#w317" to="#w328" xml:id="é12e">si le fils perdu renaquit et mourut
```

```
                en cette extase,</span>
            <span from="#w329" to="#w335" xml:id="é12f">ou s'il parvint à
                reconnaitre,</span>
            <span from="#w336" to="#w352" xml:id="é12g">ne fût-ce qu'à la manière d'un
                nouveau-né ou d'un chien,</span>
            <span from="#w353" to="#w358" xml:id="é12f_fin">les parents et la maison.</span>
          </spanGrp>
          <spanGrp xml:id="Seg2" type="Segmentation" ana="#Phrase">
            <span from="#é1" xml:id="P1"/>
            <span from="#é2a" to="#é2b" xml:id="P2"/>
            <span from="#é3a" to="#é3b" xml:id="P3"/>
            <span from="#é4a" to="#é4d" xml:id="P4"/>
            <span from="#é5a" to="#é5b" xml:id="P5"/>
            <span from="#é6a" to="#é6b" xml:id="P6"/>
            <span from="#é7a" to="#é7b" xml:id="P7"/>
            <span from="#é8a" to="#é8d" xml:id="P8"/>
            <span from="#é9a" to="#é9c" xml:id="P9"/>
            <span from="#é10a" to="#é10c" xml:id="P10"/>
            <span from="#é11a" to="#é11c" xml:id="P11"/>
            <span from="#é12a" to="#é12f_fin" xml:id="P12"/>
          </spanGrp>

          <span type="StructureCompositionnelle" from="#é2a" to="#é3a" xml:id="Pn1.é2a-é3a"/>
          <span type="StructureCompositionnelle" from="#é3b" to="#é4a" xml:id="Pn2.é3b-é4a"/>
          <span type="StructureCompositionnelle" from="#é4d" xml:id="Pn2.é4d"/>
          <span type="StructureCompositionnelle" from="#P5" to="#P7" xml:id="Pn3.P5-
P7">(Ré)Action</span>
          <span type="StructureCompositionnelle" from="#P8" to="#P9" xml:id="Pn4.P8-P9"/>
          <span type="StructureCompositionnelle" from="#P10" xml:id="Pn5.P10"/>
          <span type="StructureCompositionnelle" from="#é1" xml:id="Pn0">Cadre
                médiatif</span>
          <span type="StructureCompositionnelle" from="#é4b" to="#é4c" xml:id="Pn0a">Évaluation
commentative</span>
          <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-
proposition_p">premier argument </span>
          <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-proposition_q">second
argument</span>
          <span type="StructureCompositionnelle" from="#é11c" xml:id="é11c-
conclusion_non_c">renversement de la conclusion implicite du
                retour définitif à la maison</span>
          <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-Pn1">Situation
                initiale Pn1</span>
          <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-Pn2">Noeud
                Pn2</span>
          <span type="StructureCompositionnelle" from="#é11c" xml:id="é11c-Pn4">Dénouement
                Pn4</span>

          <span type="StructureCompositionnelle" from="#P12" xml:id="PnΩ">Évaluation
                finale. «Cette prose périodique dominée par le rythme contribue au
                glissement de genre du récit factuel au récit poétique.» (Adam 2005:
                211)</span>

          <graph type="directed" xml:id="ana1">
            <label>StructureCompositionnelle</label>

            <node xml:id="ana1_séquence_narrative_1" n="séquence narrative">
```

```
  <label ana="#SCsequence_narrative">séquence narrative [séquence_narrative_1]</label>
</node>
<node xml:id="ana1_Pn1">
  <label ana="#SCsituation_initiale">[Pn1]</label>
</node>
<node xml:id="ana1_Pn1.é2a-é3a" value="Pn1.é2a-é3a">
  <label>[Pn1.é2a-é3a]</label>
</node>
<node xml:id="ana1_intrigue_1">
  <label>[intrigue_1]</label>
</node>
<node xml:id="ana1_Pn2">
  <label ana="#SCnoeud">[Pn2]</label>
</node>
<node xml:id="ana1_Pn2.é3b-é4a" value="Pn2.é3b-é4a">
  <label>[Pn2.é3b-é4a]</label>
</node>
<node xml:id="ana1_Pn2.é4d" value="Pn2.é4d">
  <label>[Pn2.é4d]</label>
</node>
<node xml:id="ana1_Pn3">
  <label ana="#SCaction">[Pn3]</label>
</node>
<node xml:id="ana1_Pn3.P5-P7" value="Pn3.P5-P7">
  <label>[Pn3.P5-P7]</label>
</node>
<node xml:id="ana1_Pn4">
  <label ana="#SCdénouement">[Pn4]</label>
</node>
<node xml:id="ana1_Pn4.P8-P9" value="Pn4.P8-P9">
  <label>[Pn4.P8-P9]</label>
</node>
<node xml:id="ana1_Pn5">
  <label ana="#SCsolution_finale">[Pn5]</label>
</node>
<node xml:id="ana1_Pn5.P10" value="Pn5.P10">
  <label>[Pn5.P10]</label>
</node>
<node xml:id="ana1_entrée-préface">
  <label ana="#SCsequence_narrative">[entrée-préface]</label>
</node>
<node xml:id="ana1_Pn0" value="Pn0">
  <label>[Pn0]</label>
</node>
<node xml:id="ana1_Pn0a" value="Pn0a">
  <label>[Pn0a]</label>
</node>
<node xml:id="ana1_P11_argumentative">
  <label ana="#SCpériode_argumentative">[P11_argumentative]</label>
</node>
<node xml:id="ana1_é11a-proposition_p" value="é11a-proposition_p">
  <label>[é11a-proposition_p]</label>
</node>
<node xml:id="ana1_é11b-proposition_q" value="é11b-proposition_q">
  <label>[é11b-proposition_q]</label>
</node>
<node xml:id="ana1_é11c-conclusion_non_c" value="é11c-conclusion_non_c">
```

```
  <label>[é11c-conclusion_non_c]</label>
</node>
<node xml:id="ana1_P11_narrative">
  <label ana="#SCpériode_narrative">[P11_narrative]</label>
</node>
<node xml:id="ana1_é11a-Pn1" value="é11a-Pn1">
  <label>[é11a-Pn1]</label>
</node>
<node xml:id="ana1_é11b-Pn2" value="é11b-Pn2">
  <label>[é11b-Pn2]</label>
</node>
<node xml:id="ana1_é11c-Pn4" value="é11c-Pn4">
  <label>[é11c-Pn4]</label>
</node>
<node xml:id="ana1_simple_période_P11">
  <label ana="#SCpériode">[simple_période_P11]</label>
</node>
<node xml:id="ana1_évaluation_finale">
  <label ana="#SCpériode">[évaluation_finale]</label>
</node>
<node xml:id="ana1_PnΩ" value="PnΩ">
  <label>[PnΩ]</label>
</node>
<node xml:id="ana1_plan_de_texte_du_Captif">
  <label ana="#SCplan_de_texte">[plan_de_texte_du_Captif]</label>
</node>
<arc from="#ana1_séquence_narrative_1" to="#ana1_Pn1">
  <label>contient</label>
</arc>
<arc from="#ana1_séquence_narrative_1" to="#ana1_intrigue_1">
  <label>contient</label>
</arc>
<arc from="#ana1_séquence_narrative_1" to="#ana1_Pn5">
  <label>contient</label>
</arc>
<arc from="#ana1_Pn1" to="#ana1_Pn1.é2a-é3a">
  <label>contient</label>
</arc>
<arc from="#ana1_intrigue_1" to="#ana1_Pn2">
  <label>contient</label>
</arc>
<arc from="#ana1_intrigue_1" to="#ana1_Pn3">
  <label>contient</label>
</arc>
<arc from="#ana1_intrigue_1" to="#ana1_Pn4">
  <label>contient</label>
</arc>
<arc from="#ana1_Pn2" to="#ana1_Pn2.é3b-é4a">
  <label>contient</label>
</arc>
<arc from="#ana1_Pn2" to="#ana1_Pn2.é4d">
  <label>contient</label>
</arc>
<arc from="#ana1_Pn3" to="#ana1_Pn3.P5-P7">
  <label>contient</label>
</arc>
<arc from="#ana1_Pn4" to="#ana1_Pn4.P8-P9">
```

```
            <label>contient</label>
          </arc>
          <arc from="#ana1_Pn5" to="#ana1_Pn5.P10">
            <label>contient</label>
          </arc>
          <arc from="#ana1_entrée-préface" to="#ana1_Pn0">
            <label>contient</label>
          </arc>
          <arc from="#ana1_entrée-préface" to="#ana1_Pn0a">
            <label>contient</label>
          </arc>
          <arc from="#ana1_P11_argumentative" to="#ana1_é11a-proposition_p">
            <label>contient</label>
          </arc>
          <arc from="#ana1_P11_argumentative" to="#ana1_é11b-proposition_q">
            <label>contient</label>
          </arc>
          <arc from="#ana1_P11_argumentative" to="#ana1_é11c-conclusion_non_c">
            <label>contient</label>
          </arc>
          <arc from="#ana1_P11_narrative" to="#ana1_é11a-Pn1">
            <label>contient</label>
          </arc>
          <arc from="#ana1_P11_narrative" to="#ana1_é11b-Pn2">
            <label>contient</label>
          </arc>
          <arc from="#ana1_P11_narrative" to="#ana1_é11c-Pn4">
            <label>contient</label>
          </arc>
          <arc from="#ana1_simple_période_P11" to="#ana1_P11_argumentative">
            <label>contient (alt incl poids 0.5)</label>
          </arc>
          <arc from="#ana1_simple_période_P11" to="#ana1_P11_narrative">
            <label>contient (alt incl poids 0.5)</label>
          </arc>
          <arc from="#ana1_évaluation_finale" to="#ana1_PnΩ">
            <label>contient</label>
          </arc>
          <arc from="#ana1_plan_de_texte_du_Captif" to="#ana1_entrée-préface">
            <label>contient</label>
          </arc>
          <arc from="#ana1_plan_de_texte_du_Captif" to="#ana1_séquence_ narrative_1">
            <label>contient</label>
          </arc>
          <arc from="#ana1_plan_de_texte_du_Captif" to="#ana1_simple_période_P11">
            <label>contient</label>
          </arc>
          <arc from="#ana1_plan_de_texte_du_Captif" to="#ana1_évaluation_finale">
            <label>contient</label>
          </arc>
        </graph>
      </body>
    </text>
  </TEI>
```

On aurait aussi pu développer les phrases jusqu'aux énoncés, mais ce n'était pas nécessaire pour illustrer notre propos. Évidemment, on peut faire appel à une librairie graphique pour produire une image du graphe. Si la librairie n'accepte pas directement la description TEI, une autre feuille XSLT pourra traduire le document XML dans la syntaxe de la librairie.

6.2.2.2 Énonciation narrative et source du savoir.

D'autres dimensions sont évoquées par Adam pour l'analyse du récit de Borges. Ainsi, dans une section de son ouvrage (Adam 2005:212), il aborde «*la question de la prise en charge énonciative (PdV) des propositions*». L'argumentaire prend la forme d'un commentaire analytique qui suit le déroulement du récit de Borges en référant à des passages. La mise en forme TEI de cette partie de l'analyse pourrait donc prendre la forme de paragraphes en prose encadrés d'un <*div*> qui reprend la division formelle de l'ouvrage d'Adam en sections coiffées de sous-titres. Les références aux énoncés du récit emprunteront la syntaxe de l'élément <*span*> pour pointer de manière rigoureuse sur les énoncés du récit inscrits dans le document source externe.

6.2.2.3 Référent évolutif et identité narrative.

Dans cette section, Adam aborde la question de l'*identité narrative* à travers l'examen des «*reprises du référent du personnage principal*» (Adam 2005:213). Cet examen l'amènera à parler d'un *référent évolutif* qui se modifie tout au long de la chaine co-référentielle et anaphorique. Dans l'exposé en prose de l'auteur, il suffit donc de remplacer les renvois aux énoncés par des balises <*span*> qui réfèrent précisément aux passages du texte portant sur le personnage principal, tantôt décrit comme l'indien, tantôt comme le fils retrouvé. Pour éviter de reprendre le contenu du texte explicatif d'Adam, nous nous contenterons, dans l'exemple qui suit, de citer les mots qui précèdent immédiatement les références au récit de Borges.

```
<div type="Analyse" ana="#référence" xml:base="borges_adam.xml" xml:id="référent_personnage_principal">
<p> La question de l'identité narrative...
... L'amorce de la chaîne par <span from="#w14" to="#w15">un enfant</span>
... pronominalisations ...<span from="#w30">l'</span>
... <span from="#w36">le</span>
... Mais une nouvelle chaîne <span from="#w57" to="#w63">un Indien aux yeux couleur de ciel</span>
... pronom ...<span from="#w73">le</span>
... <span from="#w101">le</span>
... hyperonyme... <span from="#w104" to="#w105">l'homme</span>
.. reprises pronominales neutres en «il» …
      <spanGrp>
          <span from="#w135"/>
          <span from="#w144"/>
          <span from="#w168"/>
          <span from="#w177"/>
```

```
            <span from="#w207"/>
            <span from="#w226"/>
            <span from="#w232"/>
        </spanGrp>
... <span from="#w253" to="#w254">leurs fils</span>
...
</p>
</div>
```

Même si l'analyse prend ici la forme d'un commentaire plutôt que celle d'une formalisation, la référence sous forme d'élément <*span*> permet de pointer sur l'objet textuel. Bien qu'on ne l'illustre pas ici, la référence pourrait utiliser l'attribut *xml:id* pour permettre au logiciel de maintenir un double pointage allant de l'analyse au récit et du récit à l'analyse. Même si, au sens strict, la double référence appartient à la sémantique de l'élément <*link*>, elle pourrait être construite dynamiquement par le logiciel qui mettra en forme la navigation entre le récit et les documents d'annotation. Ajoutons aussi que nous aurions pu utiliser l'attribut *ana* pour lier la référence à sa catégorie dans le contexte d'une typologie de la coréférence.

6.2.2.4 Une fable sur le temps, la mémoire et l'oubli.

La dernière section de l'analyse d'Adam fait un pas de plus dans l'interprétation en lui donnant une portée plus large ancrée dans l'œuvre de Borges. Les références portent donc non seulement sur le texte étudié, mais aussi sur des citations tirées de d'autres textes de Borges. Par exemple, il cite une conférence à Buenos Aires en 1978 éditée chez Folio-Essais en 1985. Si on avait une version électronique segmentée en mots de cette conférence, on pourrait référer directement à l'extrait cité comme on l'a fait pour *Le Captif*. À défaut, il est quand même possible de formaliser un peu la référence en renvoyant à une entrée électronique de la bibliographie.

Du point de vue informatique, les techniques de balisage TEI déjà illustrées conviennent tout à fait pour rendre compte de l'ancrage de l'explication en prose aux textes intégraux. Cela suppose, bien sûr, que les les écrits de Borges soient disponibles en format numérique sous forme de ressources électroniques pouvant être référées par des URI. Le balisage TEI de ces ressources n'est pas strictement obligatoire dans la mesure où divers schémas de pointage peuvent être utilisés. Il reste que la cohérence de l'ensemble sera largement favorisée par le fait que les documents analysés et les documents d'analyse sont tous en TEI. Aussi, le découpage en mots, s'il n'est pas strictement obligatoire, facilitera grandement la cohérence et la simplicité d'implantation des mécanismes de pointage. En pratique, on doit constater que

les grands efforts de numérisation que l'on retrouve dans les projets de bibliothèques en ligne procèdent, à des fins d'indexation, à des opérations de reconnaissance optique des caractères passant des textes numérisés en format image à des textes en format caractère. Ce processus de reconnaissance optique des caractères et d'indexation plein texte fournit déjà un découpage des documents en mots. C'est le cas, par exemple, dans le projet Gallica 2 à la bibliothèque nationale de France.

6.3 Quelques mots de conclusion

L'objectif de cet exercice de mise en forme TEI d'un certain nombre de modèles en analyse textuelle des discours était de mesurer, à travers une syntaxe externe des données, le niveau de complexité interne qui sera requis pour une représentation informatique de ces analyses structurelles. Il s'agit, rappelons-le de concevoir une extension possible du modèle lexique/occurrences de SATO pour lui adjoindre un modèle de segments dynamiques apte à représenter les dimensions structurelles de l'analyse de discours, au-delà de la lexicométrie traditionnelle.

Pour peu que cette incursion exploratoire en analyse textuelle du discours soit représentative du domaine, on peut penser que la simplicité syntaxique relative des représentations exposées ici nous fournira des exemples pertinents pour la modélisation informatique des structures internes des données, et pour leur présentation dans un modèle usager. C'est ce que nous examinons dans le chapitre suivant.

Alors que nous avons choisi de nous concentrer sur des problèmes de linguistique textuelle, nous constatons que l'emploi de la TEI dans des branches plus classiques de la linguistique est aussi d'actualité. Ainsi, par exemple, Przepiórkowski (2009), propose d'utiliser la TEI pour l'annotation syntaxique.

> The aim of the paper is to show that a subset of Text Encoding Initiative Guidelines is a reasonable choice as a standard for stand-off XML encoding of syntactically annotated corpora. The proposed TEI schema — actually employed in the National Corpus of Polish — is compared to other such candidate standards, including TIGER-XML, SynAF and PAULA. (Przepiórkowski 2009, abstract)

On notera également que si nous avons emprunté nos exemples à des situations variées d'analyse linguistique, en particulier dans sa dimension textuelle, la première utilisation pratique de l'annotation structurelle concernera probablement davantage la représentation des liens explicites dans le texte. Un cas typique est le corpus des procès-verbaux du Comité d'instruction publique (CIP) donné en exemple (cf. 4.10b) pour illustrer notre proposition du de dépôt de données adapté à la constitution de corpus de recherche (Daoust et coll. 2008). Ce corpus, fruit d'une édition critique, contient un imposant dispositif de notes, d'index analytiques, et de renvois de toutes sortes. Ce réseau de relations correspond à autant de structures reliant des empans textuels. La traduction de ces références dans le formalisme des documents d'annotation en XML-TEI sera la porte d'entrée d'une implantation informatique permettant la navigation hypertextuelle et la définition de sous-textes basés sur ces mises en relations. Ces sous-textes seront, à leur tour, la porte d'entrée vers les analyseurs textométriques.

Bibliographie du chapitre 6

Adam, 2005. Adam, J.-M. *La linguistique textuelle, Introduction à l'analyse textuelle des discours*. Armand Colin, Paris ISBN 2-220-26752-5.

ATONET, 2005. *Réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur* (ATONET) : http://www.atonet.net

Bakhtine, 1984. Bakhtine, M. *Esthétique de la création verbale*. Paris, Gallimard, 1984.

Charolles, 1993. Charolles, M. Les plans d'organisation du discours et leur interaction, in Moirand, S. et alii, ed., *Parcours linguistiques de discours spécialisés.*, Berne, Peter Lang, 1993 : 301-314.

Daoust, Marcoux et Viprey, 2010. Daoust F. ; Marcoux, Y. ; Viprey, J.-M. L'annotation structurelle, in *Actes des JADT-2010. http://www.ledonline.it/ledonline/JADT-2010/allegati/JADT-2010-1145-1156_057-Daoust.pdf*

Daoust, 2009. Daoust F. Système d'analyse de texte par ordinateur, SATO, Manuel de référence, version 4.3. Centre d'analyse de texte par ordinateur, UQAM, 2007; modifié en 2009. http://www.ling.uqam.ca/sato/satoman-fr.html

Daoust et coll. 2008. Daoust, F.; Duchastel, J.; Marcoux, Y.; Rizkallah, E. JADT-2008. Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche, in *Actes des JADT-2008*, vol. 1, pp- 355-367, Presses universitaires de Lyon, 2008. ISBN 978-2-7297-0810-8. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2008/pdf/daoust-duchastel-marcoux-rizkallah.pdf

Daoust et Marcoux, 2006. Daoust F. et Marcoux Y. Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés, in *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp 327-340, Presses universitaires de Franche-Comté, 2006. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/029.pdf

Fleury, 2009. Fleury, S. *Le métier textométrique* (Trameur). Centre de textométrie – CLA^{2}T, U. Paris 3 Sorbonne nouvelle, http://tal.univ-paris3.fr/trameur/

Habert, 1998. Habert B. *Des mots complexes possibles aux mots complexes existants : l'apport des corpus*, *Mémoire présenté pour l'obtention d'une habilitation à diriger des recherches.* Document de synthèse, Université Lille III - Charles de Gaulle *http://www.limsi.fr/Individu/habert/Publications/Fichiers/hdr/node4.html*

Lebart, 2005. Lebart, L. *Data and Text Mining.* École nationale supérieure de télécommunications, Paris, http://www.enst.fr/egsh/lebart/

Przepiórkowski, 2009. Przepiórkowski A. TEI P5 as an XML Standard for Treebank Encoding *The Eighth International Workshop on Treebanks and Linguistic Theories* *http://tlt8.unicatt.it/FullPaper/D_2.pdf*

Reinert, 2002. Reinert, M. *ALCESTE, Manuel de référence.* Université de Saint-Quentin-en-Yvelines, CNRS.

Salem, A. et coll. 2003. *Manuel Lexico 3*. http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/lexicoWWW/team.htm

TEI Consortium, 2007. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium, eds. http://www.tei-c.org/Guidelines/P5/

Viprey, 2005. Viprey, J-M. *DiaTag–Astartex.*. Université de Franche-Comté. http://laseldi.univ-fcomte.fr/document/viprey/page_JMV.htm

W3C, 2009a. *XQuery Update Facility* 1.0. W3C W3C Candidate Recommendation 09 June 2009. http://www.w3.org/TR/2007/REC-xquery-20070123/

W3C, 2007a. *XML Path Language (XPath) 2.0.* W3C Recommendation 2007. http://www.w3.org/TR/2007/REC-xpath20-20070123/

W3C, 2007b. *XQuery 1.0: An XML Query Language.* W3C Recommendation 2007. http://www.w3.org/TR/2007/REC-xquery-20070123/

W3C, 2007c. *XSL Transformations (XSLT) Version 2.0.* W3C Recommendation 2007. http://www.w3.org/TR/2007/REC-xslt20-20070123/

Weinrich, 1973. Weinrich, H. *Le temps.* Paris, Seuil (1964), 1973. Cité par Adam 2005.

7 Modèles informatiques pour l'exploitation de la structure formelle et des segments dynamiques.

7.1 Introduction

Après avoir examiné diverses représentations TEI pour l'annotation externe de structures textuelles, il convient d'évaluer diverses stratégies informatiques pour exploiter ce type de structures. Rappelons que l'annotation externe réfère ici à des documents d'annotation sous forme de fichiers XML annotant le document source. Dans le chapitre précédent, ce qui nous a d'abord intéressé, ce sont les segments dynamiques construits sur des documents à des fins d'analyse. Mais, on a aussi vu que la référence à ces segments analytiques peut s'accompagner de commentaires en format libre accompagnés de références aux extraits commentés. Dans le cas d'une analyse donnée, il appartiendra à l'analyste de lexicaliser ces commentaires, comme étant des éléments du corpus, ou de les conserver comme des annotations élaborées.

Même si on s'est d'abord intéressé à l'annotation structurelle, on doit aussi pouvoir gérer la structure formelle des documents qui expose, de façon statique, l'organisation des documents. Un certain nombre de caractéristiques distinguent ces deux types de structures.

La structure formelle est une donnée primaire qui rend compte de l'organisation hiérarchique du document soumis à l'analyse. Cette structure est réputée non modifiable et soumise à une arborescence unique généralement construite directement sur le contenu textuel d'un document. Les structures dynamiques, au contraire, sont variables et susceptibles de se modifier en cours d'analyse. Elles sont aussi multiples et s'organisent en hiérarchies indépendantes et concurrentes. Comme on l'a montré, la nature dynamique des structures

analytiques peut se représenter par des pointeurs sur des éléments référant au texte analysé. Ainsi, il devient possible de construire des arborescences autonomes sur ces éléments agissant comme des relais vers le texte analysé. D'un point de vue documentaire, et pour rendre compte de la dynamique même du travail d'analyse-annotation, on a proposé de considérer ces structures dynamiques comme des documents externes d'annotation.

Malgré ces différences entre la structure formelle d'un document et les structures dynamiques construites par l'analyse, il est à noter que, d'un point de vue syntaxique, la structure hiérarchique peut aussi être transformée en annotation externe constituant, en quelque sorte, une annotation initiale par rapport au texte linéaire. Cette transformation implique que l'on puisse pointer sur des contenus textuels selon le niveau de granularité le plus pertinent pour l'analyse : caractère, morphème, mot, syntagme, phrase, etc.

Dans son état actuel, SATO, sous l'angle XML, organise le texte comme une suite d'éléments simples faisant l'objet d'annotations externes sous forme de propriétés apparentées à des structures de traits atomiques. Dotés d'identifiants uniques (valeur de l'attribut *xml:id*), les éléments XML contenant ces chaines de caractères (occurrences ou *token* en anglais) lexicalisées (formes lexicales ou *type* en anglais) fournissent les ancrages de base à l'annotation structurelle.

La façon classique d'exploiter des documents XML consiste à partir du sommet des arborescences pour repérer des sous-arbres selon diverses contraintes : des contraintes de dominance dans la hiérarchie des nœuds, des contraintes de proximité pour des nœuds de même niveau ou des contraintes reliées aux attributs des nœuds ou à leur contenu textuel. Nous pouvons intégrer le décompte de ces éléments structuraux à nos méthodes textométriques classiques en termes de nombre de segments textuels ou d'unités recouverts par ces sous-arbres, ou en termes de nombre de sous-arbres présents dans des parties du corpus. En d'autres termes, il est relativement naturel d'ajouter la possibilité de compter des structures à notre pratique actuelle de dénombrement de mots, de valeurs de propriété (attributs), de formes lexicales, de cooccurrents ou de collocations.

Donc, ce schéma classique XML, où l'on traverse la structure de façon descendante, du sommet vers les segments, est pertinent pour l'analyse textuelle telle que nous pratiquons. Mais ce schéma doit aussi être complété par une approche ascendante qui nous fait voir la structure *à partir du bas*, c'est-à-dire à partir des contextes que l'on fréquente par une lecture

cursive ou selon des parcours variables guidés par divers outils d'exploration des données. Ces parcours vont du simple appel de notes à l'exploitation de l'annotation structurelle, de la simple concordance à la lecture de contextes dirigés par des classifications, des distances lexicales, des plans factoriels, des proximités de cooccurrences, etc. Ces cheminements peuvent, en fait, déjà être assimilés à des parcours de structures et de graphes tissant des relations entre contextes. L'enjeu de ces parcours de type hypertextuel est de pouvoir remonter du contexte vers les autres chemins marqués susceptibles de nous y conduire. L'exploitation de ces chemins à partir de la localité textuelle est donc tout aussi pertinente que leur exploitation *par le haut*. Le modèle d'implantation informatique devra donc pouvoir rendre compte de façon efficace de ces deux types de parcours sur de grands ensembles textuels.

Comme nous l'avons déjà exposé, notre projet de gestion de segments construits dynamiquement au cours de l'analyse, dans un travail fluide de mise en relation, a d'abord pris forme dans le contexte du projet AlexATO en 1993. C'était donc bien avant la mouvance XML qui propose aujourd'hui des formalismes faisant office de normes. Ces normes ont eu l'avantage d'amener le développement de langages et d'outils logiciels. Mais elles sont aussi basées sur des modèles qui privilégient la structure de dominance, voire de dominance unique dans le cas du modèle le plus classique. La question qui se posait à l'époque, et qui se pose encore aujourd'hui, est la conciliation entre des formalismes qui considèrent les relations entre segments comme des attributs, par rapport à ceux qui privilégient la relation de dominance et s'expriment en termes de parcours des arbres. en s'inspirant davantage de l'idée de *chemins* dans un graphe. La première approche, de nature plus algébrique, s'inspire du modèle relationnel, rigoureux du point de vue formel et pour lequel il existe des implantations optimisées. Un autre argument plaidant en faveur de l'extension du modèle pour prendre en compte les relations hiérarchiques était d'offrir un formalisme unique faisant appel à un langage unique. C'est en pensant à cette approche que nous avons formulé nos premières hypothèses de travail dans les années 1990. Nous commencerons donc d'abord par la présentation de cette approche pour ensuite réactualiser le débat dans les termes d'aujourd'hui.

7.2 Le modèle algébrique

La question d'un modèle de gestion des segments dynamiques, soulevée en 1993 dans le contexte du projet AlexATO, visait à formuler un modèle algébrique de traitement permettant de calculer en temps réel des segments condensant de multiples niveaux d'annotation et de structuration. On retrouve dans les publications scientifiques de l'époque (voir plus bas) plusieurs articles manifestant une intention similaire, surtout dans un contexte de recherche documentaire dépassant la simple indexation plein texte.



Nos hypothèses de 1993 sur les segments dynamiques (7.2a, remarque)

Il peut être intéressant de relire certaines des notes de travail de 1993. On verra, en particulier, que la question de l'annotation sur les segments est posée d'abord en relation avec l'annotation à plat, mais annotation tout de même. Les notes de l'époque utilisent d'ailleurs le terme de segments catégoriels. On notera malgré tout que la représentation de la macro-structure du texte est déjà perçue comme une application naturelle des segments catégoriels. La proposition d'annotation structurelle, que nous défendons aujourd'hui comme constitutive des segments dynamiques est donc une évolution conséquente par rapport à nos intuitions du début des années 1990.

Qu'est-ce qu'un segment?

En termes généraux, on entend par segment textuel une suite continue de mots. Il peut s'agir d'un chapitre, d'une phrase, d'une tirade, d'un syntagme, etc.

Dans SATO, lorsque plusieurs mots consécutifs partagent une même valeur de propriété, on procède, lors de l'édition du texte à une *mise en évidence* :

Maître***locuteur=lafont** Corbeau***locuteur=lafont**

devient

***locuteur=lafont** Maître Corbeau

À l'inverse, lorsque SATO, lors de la génération du corpus, rencontre une fonction de catégorisation de segment (affectation globale), il comprend qu'il s'agit d'une *factorisation* et effectue la distribution du *facteur* sur chacun des mots qui suit.

C'est dire que, dans sa représentation du texte, SATO ne connaît pas le segment. En ce sens, ce type de segment, que l'on pourrait dire catégoriel, est un artifice d'annotation. C'est un segment virtuel en ce sens qu'il ne se matérialise pas dans une structure informatique explicite.

Le deuxième type de segment manipulé par SATO est un résultat d'un dispositif (algorithme) de segmentation. Ces segments construits n'ont pas de pérennité dans la représentation SATO du texte. Ce sont davantage des points de vue de lecture. Souvent, on va se servir de ces points de vue de lecture pour marquer les unités de base de SATO (lexèmes et occurrences) ou produire des mesures sur le texte.

Finalement, il y a dans SATO une utilisation, très spécifique cependant, de segments désignés. Ce sont donc des segments explicites qui ont une autonomie de représentation. Il s'agit des extraits que l'on construit à partir du *catégorisateur plein écran*. Ces structures ne sont pas intégrées à l'ensemble du logiciel. On ne peut que les afficher ou les imprimer. On peut également s'en servir dans l'algorithme de comparaison d'extraits (commande MARQUER). La référence de pagination est aussi, dans SATO, un segment désigné mais dont la structure n'est pas généralisée.

L'extension des structures segmentaires dans SATO vise à répondre à 3 types de besoins.

En termes de description textuelle, il ne semble pas très naturel de penser la représentation de la macro-structure en termes de segments virtuels. La macro-structure s'apparente en effet à la pagination par sa dimension référentielle et parce qu'elle qualifie souvent de grands segments pour lesquels la *mise en évidence* n'est pas très appropriée.

En termes d'analyse de texte, on perçoit un besoin de structuration des rapports entre segments : rapports d'emboitement, d'intersection, de succession, liens hypertextuels.

En termes informatiques, l'utilisation des segments permettrait d'éviter la saturation des propriétés qui sont actuellement toujours condensées au niveau du mot ou du

lexème. D'un autre côté, la gestion des segments risque d'exiger davantage d'efforts de calcul. D'où l'intérêt de les développer dans la perspective d'une implantation sur ordinateur à traitement parallèle.

Une extension naturelle de SATO en termes de représentation segmentaire serait de conserver la factorisation des propriétés. En d'autres mots, une propriété segmentaire ne serait pas distribuée explicitement sur les mots du segment. Inversement, un segment ne pourrait exister qu'en tant que suite possédant une même catégorisation (propriété factorisée). Il s'agit en somme de donner une existence physique au *segment virtuel* sans en changer la nature algébrique.

Cela implique donc que les segments construits ne pourront exister, en dehors de la procédure qui les a construits, que s'ils sont catégorisés. Ils deviennent alors des segments catégoriels et c'est à ce titre qu'ils seront interrogeables.

On peut associer aux segments catégorisés la notion de constructeurs. Les constructeurs ont pour objet de construire des graphes à permettant de relier les segments entre eux. Par exemple, un constructeur de type hiérarchique aurait pour objet de représenter la structure d'emboitement entre les segments partageant une même structure catégorielle. Un constructeur de type hypertextuel vise plutôt à lier entre eux des segments disjoints possédant une même valeur catégorielle. Par exemple, un constructeur *référence* pourrait être utilisé pour lier une citation à l'ouvrage cité.

Hypothèse d'implantation

Le constructeur construit un graphe à partir d'une propriété sur les segments. D'un point de vue formel, il n'est pas utile de représenter le graphe de façon explicite dans la base textuelle. Il vaudrait mieux disposer d'un dispositif informatique qui construise le graphe au moment où on en a besoin. Cela implique que l'on puisse facilement rassembler tous les segments qui se trouvent dans le voisinage d'un mot donné.

Le voisinage peut être défini comme un segment calculé qui englobe un mot donné et dont la définition s'obtient à partir du numéro d'ordre de l'occurrence (son

adresse).

En utilisant ces voisinages calculés comme tête d'index sur les segments catégoriels dont l'intersection avec le voisinage est non nulle, on peut envisager une implantation efficace des segments et des constructeurs associés. Dans SATO-ACTE, nous possédons déjà une librairie d'index appelé FIR (Fichier des identificateurs et des références). Cette librairie a été développée par les informaticiens affectés à la gestion informatisée des bibliothèques de l'UQAM.

Malheureusement, le FIR actuel n'utilise que des références de 4 octets, ce qui se prête mal à la représentation directe d'un segment. Aussi, nous avons pris entente avec les développeurs du FIR qui ont accepté de nous fournir les fonctionnalités dont nous avions besoin en échange de services linguistiques de la part de notre équipe.

Au niveau des hypothèses d'implantation, nos notes de 1993 amènent, à travers l'idée du constructeur, l'hypothèse que la structure pourrait être construite dynamiquement en conformité avec des schémas de construction s'appuyant sur les segments catégoriels. Évidemment, l'idée qu'on puisse se passer de structures de données permanentes pour la conservation des graphes était très ambitieuse et en fait assez irréaliste si on considère des exemples comme le plan de texte du récit de Borges. La reconstruction du plan à partir des segments impliquerait l'existence d'un constructeur si précis qu'il ressemblerait à s'y méprendre au graphe que l'on voulait éviter de conserver dans une structure de données permanente. De plus, la fouille de tels graphes, s'ils devaient être reconstruits à la volée, imposerait une charge de calcul considérable. Nonobstant cette réserve importante, l'idée du constructeur, vu davantage comme un assistant à la construction des annotations structurelles, sera sans doute à reprendre dans le cadre d'un environnement ergonomique d'annotation structurelle.

Par ailleurs, la lecture d'articles de cette époque nous a permis de constater que ces pistes de recherche étaient en émergence dans les années 1990, en particulier dans une perspective de recherche documentaire. Ainsi, l'article de Clarke, Gormack et Burkowski (1994) est très représentatif de cette tendance qui avance un modèle qui n'impose pas un schéma unique et

fixé d'avance. L'approche est basée sur un modèle de données simple appelé *GC-list*, ou liste généralisée de concordances. Ces listes constituent des index de segments textuels. Le modèle propose une représentation du texte dans laquelle les unités sont numérotées. Ces unités sont construites selon divers alphabets : un pour les unités textuelles jugées pertinentes et un autre pour les balises. Un troisième alphabet, assimilable à un anti-dictionnaire (*stoplist*), écarte les unités jugées peu significatives du point de vue de la recherche documentaire. Un empan textuel peut donc être décrit par les numéros des unités débutant et terminant la séquence textuelle. Ces listes de contextes constituent des index sur lesquels sont appliqués divers opérateurs algébriques de segments.

Le point fort du modèle est qu'il substitue la relation d'emboitement à la relation de hiérarchie typique de la représentation en arbre. La combinaison d'opérateurs de segments, moins de dix en fait, permet de proposer une algèbre agissant à titre de langage intermédiaire pour formuler des requêtes. Par exemple, dans un texte de théâtre, on vise à répondre à une question du type *trouver les réparties qui contiennent tel ou tel mot dans la première ligne, mais mais pas tel autre dans la seconde*, ou encore, *trouver les réparties qui contiennent tel mot et qui apparaissent dans une scène contenant une certaine ligne de texte*. Évidemment, même si la réponse à ces requêtes repose sur des règles algébriques indépendantes du schéma de données, la pertinence de la règle implique l'existence d'un modèle structurel spécifique. Il reste que la constitution de la base n'exigeant pas de schéma, il est possible d'y déposer des documents de structures quelconques. L'autre avantage du modèle proposé est sa compatibilité avec les primitifs du modèle relationnel. Il faut cependant noter que le modèle algébrique, du moins celui présenté par Clarke, Gormack et Burkowski, ne supporte pas l'indirection, c'est-à-dire des liens hypertextuels, par exemple entre l'appel et le texte de la note.

Avec l'arrivée des nouveaux formalismes XML, d'autres articles proposent des extensions aux langages de requêtes plus traditionnels.

> This paper presents structural recursion as the basis of the syntax and semantics of query languages for semistructured data and XML. We describe a simple and powerful query language based on pattern matching and show that it can be expressed using structural recursion, which is introduced as a top-down, recursive function, similar to the way XSL is defined on XML trees. (Buneman et coll. 2000)

7.3 Le modèle arborescent

Ces dernières années ont vu apparaitre des propositions de langages permettant d'exploiter de façon directe des documents XML. Dans ce contexte, le format XML n'est plus simplement vu comme un format d'échange, mais plutôt comme un format natif pouvant être interrogé et exploité directement. Ainsi, la norme Xpath, permettant de désigner des parties de documents en s'appuyant sur le parcours de la structure arborescente des documents XML, servira de base pour des langages permettant de transformer des documents (cf. XSLT) et de les interroger par des langages de requête (cf. Xquery) apparentés aux langages traditionnels d'interrogation de base de données. L'implantation efficace de requêtes sur des bases de document XML implique l'utilisation de mécanismes sophistiqués d'indexation et d'optimisation permettant d'accéder à des structures sans qu'il soit requis de parcourir tous les documents à partir du nœud racine.

Selon des recensions récentes de Steven Bird et autres (Lai & Bird, 2009, 2004), pas moins de douze langages ont été développés pour la fouille de banques d'arbres linguistiques. Notons *CorpusSearch* (Randall 2008), *Finite structure query* (fsq) (Kepser 2003), *PML-TQ* (Pajas and Štěpánek, 2009),Tgrep2 (Rohde, 2001), *TIGER* (König and Lezius, 2001). Même si ces arbres linguistiques recouvrent généralement de courts empans textuels, par exemple des syntagmes ou des phrases, cette recension reste pertinente pour les arbres d'annotation en général. On y trouve les approches que nous avons qualifié d'algébriques. Elles font appel à des variables, des quantifieurs, des opérateurs booléens et des négations. On y trouve aussi des langages qui utilisent une syntaxe à base de *path*. D'ailleurs, Bird favorise maintenant cette approche.

> From our consideration of the actual queries used in the various languages mentioned above, we observe that descriptions of structure almost always involve paths (as also observed by Palm (1999). Paths are routinely used to identify particular subtrees relative to the root and to describe binary relationships between tree nodes (...). (Lai and Bird, 2009)

Bird propose d'ailleurs une extension à *Xpath* (*LPath*) qui s'intègre aux divers langages basées sur *XPath* et il en propose une implantation utilisant SQL.

> XPath is a language for describing paths in trees, and is popular for the tree-structured document markup of the XML world (Clark and DeRose, 1999). It provides a well-understood starting point for investigation of modal-style languages for linguistic tree query. LPath and LPath+ are linguistically motivated extensions to XPath (Bird et coll., 2006; Lai, 2005). An interpreter converts LPath expressions into equivalent SQL expressions over annotation graphs stored in a relational database (Bird and Liberman, 2001; Bird et coll., 2006). An open-source implementation is available as part of the Natural Language Toolkit Bird et coll. (2009), and a graphical interface is described by Bird and Lee (2007).

Un des problèmes soulevé par Bird et coll. (2006) concernant *Xpath* est la difficulté de naviguer horizontalement alors que des constructions syntaxiques diverses accentuent la distance dans l'arbre.

> XPath and XQuery support vertical navigations of a tree using the parent, ancestor, child and descendant axes, and certain horizontal navigations using the following and preceding sibling axes, and the following and preceding axes. However, other horizontal navigations which are important to linguistic queries, are lacking or cannot be easily expressed in Xpath. Bird et coll. (2006)

Bien sûr, cette remarque concernant les recherches de nature linguistique est tout à fait pertinente pour la textométrie qui s'intéresse particulièrement aux phénomènes de proximité, contextes, cooccurrences, etc. Voici comment les auteurs présentent leurs extensions.

> By adding certain horizontal navigation axes, we have both primitives and transitive closures for vertical and horizontal navigation, filling a gap in the XPath axis set. We also include subtree scoping and edge alignment which we will show are required by linguistic queries. (...)
>
> LPath provides three substantive extensions to XPath: the immediate following axis (and its converse), a scoping operator, and tree edge alignment. First, the immediate following axis, ->, is the natural one-step version of the XPath following axis, -->. We can consider this axis as taking a step to constituents immediately right of the current node. (...) Second, a scoping operator, denoted by braces {}, constrains navigations to the subtree that is rooted at a given node. The query inside the scoping braces is evaluated locally on the subtree, and cannot escape to the outside context of

> the enclosing tree. For example, /...P{...Q} finds some node Q only if it occurs inside the subtree rooted at P. Finally, left and right tree edge alignment, denoted by ^ and $ respectively, combine with the scoping operator and permit queries to constrain a node to be leftmost (rightmost) within a constituent (cf TGrep >>, and >>‘). Bird et coll. (2006)

Dans leur article de 2009, Lai et Bird présentent une extension supplémentaire pour ce qu'ils nomment Lpath+ et poursuivent sur le modèle d'implantation.

> LPath+ extends LPath by adding atomic closures to the language, e.g. (/N)* matches arbitrary length paths to descendents via nodes labelled N. (...)
>
> As already mentioned, one of our goals is to provide an efficient linguistic tree query tool. LPath and LPath+ seem to meet our linguistic requirements, but we also need to establish the formal expressiveness of these languages in order to determine the type of technology needed to implement them. As we will see, our path-based approach to linguistic tree query can be implemented using efficient and well-understood technology, namely SQL and relational databases. Lai et Bird (2009).

Finalement, en 2010, Ghodke et Bird reviennent sur les questions d'implémentation en présentant une technique d'indexation et de requête pouvant être mise à profit, selon leurs dires, par une variété de langages d'interrogation. L'article explore ces nouvelles méthodes pour augmenter la capacité de la fouille d'arbres en utilisant un moteur de recherche d'information (*IR engine*) et un mécanisme d'indexation des nœuds. Pour leur expérimentation, ils utiliseront le logiciel ouvert *Lucene*. Le mécanisme proposé reprend l'approche standard consistant à conserver un vecteur de position dans la référence du terme indexé et à utiliser ce vecteur pour calculer les contraintes structurelles. C'est ce genre de stratégie qui est utilisée, par exemple, dans le système de base de données XML comme eXist (Meier, 2003). Dans ce cas, les auteurs proposent d'utiliser le schéma *LPath* de description des nœuds (Bird et coll., 2006). Selon ce schéma, on attribue 4 nombres à chacun des nœuds numérotés dans un parcours pré-ordre des arbres (Bird et coll., 2006). On a d'abord les positions gauche et droite de l'empan textuel dominé (fermé à gauche, ouvert à droite) exprimées par les numéros des mots dans l'ordre séquentiel. Le troisième nombre est le niveau du nœud en termes de profondeur dans l'arbre (en commençant par 1 pour la racine). Finalement, le dernier nombre est le numéro du nœud parent. Ces 4 nombres pris

conjointement permettent de calculer la position d'un nœud dans l'arbre sans avoir à le traverser.

Ces n-tuples peuvent être représentés par des tables relationnelles ou dans les index (*fichiers inverses*) d'un engin de recherche d'information. Le modèle de traitement avec l'approche relationnelle est le suivant.

> Tree nodes can be stored in a relational database using a table structure (Bird et coll., 2006). Each treebank would have a single table for all nodes where each node's information is stored in a tuple. The node name is stored along with other position information and the sentence id. Every node tuple also has a unique primary key. The parent id column is a foreign key, referencing the parent node's id, speeding up parent/child join operations. In practice, queries are translated from higher level linguistic query languages such as LPath into SQL automatically, allowing users to use a convenient syntax, rather than query using SQL. (Ghodke et Bird, 2010)

Dans leur article, Ghodke et Bird rappellent que Zhang et coll. (2001) ont montré que l'approche relationnelle serait moins efficace que l'utilisation des index des engins de recherche d'information. Pour expérimenter cette approche sur de très grandes banques d'arbres linguistiques, les auteurs considèrent la phrase comme l'unité documentaire, ce qui est quand même une situation très spécifique aux arbres syntaxiques. Ils utilisent deux types d'index. On a des index dits de fréquence qui contiennent, pour chaque unité catégorielle, la liste des identificateurs de phrases couvertes par un nœud de cette catégorie et le nombre de nœuds de cette catégorie dans la phrase. On a aussi, pour chaque catégorie de nœuds, des index de position contenant les tuples de nombres. Les index de fréquences sont parcourus séquentiellement pour repérer les documents (phrases) qui contiennent tous les attributs requis. Pour ces phrases, on vérifie les contraintes à partir des index de position. Les documents eux-mêmes n'ont pas à être consultés. Les auteurs admettent qu'en réduisant le problème au seul repérage des phrases, ils réduisent de beaucoup la charge de calcul.

On constate donc que l'opposition entre les approches *algébriques* et les approches basées sur le parcours dans des arbres s'éclate en fait en multiples considérations convergeant finalement vers le problème central de l'efficacité des implantations. Avec les extensions apportées à l'une et l'autre des approches, on en arrive à une certaine équivalence logique avec la possibilité de traduction des syntaxes concrètes vers des systèmes de calcul éprouvés : bases

relationnelles ou systèmes de recherche d'information. Il reste la question des interfaces usagers qui se pose de façon relativement autonome et dont nous reparlerons plus loin.



À propos des index d'intervalles (7.3a, remarque)

Du point de vue algorithmique, une autre façon de poser le problème d'accès aux segments textuels nous est suggérée par la perspective spatiale. À partir d'un point, on voudra savoir quelles sont les intervalles, les plans ou régions dans l'espace qui contiennent ce point. En somme, quels intervalles on quelles régions sont traversées par ce point. Pour nous, le point correspond à un mot sur la trame textuelle. D'un point de vue algorithmique, la réponse à la question est souvent posée en termes d'index d'intervalles.

Les formalismes de type *Xquery,* fournissent une solution pour l'accès aux arbres d'annotation à partir de la racine des arbres. Ils permettent donc de répondre à des requêtes qui visent à répertorier des segments textuels en fonction de contraintes sur les structures qui les dominent. Mais, dans la mesure où les arbres d'annotation structurelle ne comportent, à titre d'éléments terminaux, que des références à des empans textuels, les contraintes sur les occurrences et leurs propriétés lexicales ne pourront s'exprimer, dans la requête Xquery, que par une relation d'ordre sur les valeurs des attributs *from* et *to* de l'élément *span* terminal. Pour optimiser une telle fouille, il faudrait disposer d'une forme d'index par intervalle.

Aussi, dans le contexte du parcours linéaire du texte, la question qui se pose est de savoir quelles sont les structures qui recouvrent un élément, typiquement une occurrence, ne serait-ce que pour en faire état à titre de *propriété structurelle,* sans même qu'il soit nécessaire de poser des contraintes sur la structure arborescente. L'accès aux arbres *par le bas,* c'est-à-dire par les feuilles des arbres d'annotation, s'avère donc nécessaire et il convient de trouver des algorithmes efficaces pour y répondre. Ce problème semble congruent à la recherche d'intervalles recouvrant un point dans un espace linéaire. Dans les publications scientifiques anglophones, ce problème est souvent décrit par l'expression *Interval Stabbing.*

> *Interval stabbing,* also known as the one-dimensional *point enclosure problem* is one of the most fundamental problems in computational geometry and has been studied for decades. Let l_a be the left endpoint and r_a be the right endpoint of an interval a. We address the following

static setting:

Let I be a given set of n intervals with $l_a, r_a \in Q := \{1, ..., O(n)\}$ for every $a \in I$. An interval $a \in I$ is *stabbed* by a point $q \in Q$ if $q \in a$. We want to construct simple and lightweight data structures that answer the following queries on I efficiently:

1. *Interval Stabbing Problem*: Given a query point $q \in Q$, report all intervals in I that are stabbed by q.

2. *Interval Intersection Problem*: Given a query interval $[l_q, r_q]$ with $l_q, r_q \in Q$, report all intervals $i \in I$ with $[l_i, r_i] \cap [l_q, r_q] \neq \emptyset$;.

3. *Interval Cover Problem*: Given an interval $q \in I$, report all intervals in I that contain the interval q.

4. Multiple Query Problems: These problems extend each of the problems 1-3 by allowing multiple queries $q_1 < \ldots < q_t, \forall i : q_i \in Q$, at the same time. The query points have to be given as a sorted list while the output consists of the intervals that are stabbed by at least one q_i (without double occurences).
(Schmidt, 2009)

Plusieurs algorithmes ont été proposés pour résoudre ce genre de problème. Certains font appel à des structures arborescentes : arbres d'intervalles (Edelsbrunner 1980, McCreight, 1980), arbres de segments (Bentley 1977), arbres de priorité (McCreight 1985), et autres (Schmidt, 2009) etc. D'autres algorithmes font appel à des systèmes de filtrage (Chazelle 1986) et de listes (Pugh 1990, Hanson et Johnson 1992).

En ce qui nous concerne, c'est le premier problème de la nomenclature de Schmidt qui nous intéresse. On peut aussi ajouter une variante itérative à ce problème consistant à calculer de façon efficace l'ensemble des intervalles autour du point $q+1$ à partir du résultat de la requête sur le point q. De façon plus formelle, le problème pourrait se formuler comme suit : obtenir l'ensemble I_{q+1} des intervalles autour d'un point $q+1 \in Q$ à partir de l'ensemble I_q des intervalles autour d'un point $q \in Q$ ayant été obtenu par une requête préalable. Dans nos applications, le parcours linéaire du texte, avec des sauts d'un contexte à l'autre par exemple, est la situation la plus courante. Les problèmes d'optimisation dans ce genre de situations sont déjà moins important puisqu'un simple parcours linéaire donnera déjà des résultats plus

performants que dans le cas général de requête sur un tirage aléatoire de points

7.4 Un modèle de données en couches multiples

Les logiciels de textométrie, tels SATO, permettent des parcours textuels basés sur des contraintes lexicales et contextuelles (filtres) construisant à la volée des contextes de référence. Les modalités de construction des frontières des contextes tiennent peu compte, cependant, de la structure hiérarchique des documents. Ce qui nous intéresse dans les systèmes de gestion des documents XML, ce n'est pas tant la recherche en texte intégral que le repérage de segments soumis à des contraintes structurelles. On veut aussi pouvoir restaurer, à partir des occurrences, la représentation des structures dominant ces occurrences.

La question est donc la suivante. Compte-tenu du fait que des logiciels comme SATO disposent déjà de modes efficaces d'accès au lexique et aux occurrences avec leurs propriétés respectives, peut-on représenter la composante structurelle de l'annotation comme une couche séparée au-dessus du plan lexique-occurrences?

Certes, l'alternative serait de considérer d'emblée le corpus comme un document XML que l'on peut interroger directement par les outils XML. Dans ce cas, au lieu d'être considéré comme un format d'exportation, d'échange et de conservation, XML serait considéré comme un format natif général s'appliquant à tout type de contenu semi-structuré. Dans ce contexte, des extensions de type *LPath+* peuvent permettre de mieux tenir des contraintes de séquentialité.

Mais, il faut admettre que l'apport de l'annotation structurelle en ATO reste encore à explorer et qu'un changement de paradigme qui se ferait au détriment des habitudes de travail actuelles n'est peut-être pas souhaitable. Il faut aussi tenir compte du fait que le modèle actuel bénéficie de structures logiques et de mécanismes d'optimisation conçus spécifiquement pour l'analyse interprétative des corpus, notamment le rapport au lexique. La question de l'évaluation de la charge de calcul reliée à l'utilisation des formalismes XML généraux reste ouverte et on voit que la recherche à ce niveau est dynamique. Mais, ce n'est pas là notre objet immédiat de

recherche. Nous nous intéressons plutôt à la gestion de segments définis dynamiquement pour annoter des corpus pour lesquels on dispose déjà de modèles de traitement et d'annotation *à plat*, c'est-à-dire des modèles qui portent sur les unités terminales, dans leur dimension lexicale et contextuelle.

Pour les fins de la modélisation, faisons éclater le corpus en diverses couches. Il y a d'abord la couche de base, qui correspond à qu'André Salem appelle la trame du texte (voir Söze-Duval Keyser 2008), métaphore de l'image tramée, c'est-à-dire décomposée dans un ensemble dénombrable fini de pixels. La trame du texte peut être plus ou moins fine : caractère, morphème, mot, énoncé, ligne... Mais, quelque soit le niveau de granularité choisi pour ce partitionnement du flux de caractères, l'annotation structurelle va d'abord s'appuyer sur ce découpage en atomes pour se construire. Par exemple, si on découpe le texte en mots, l'analyse morphologique sur les mots demeure accessible, mais ce sont les mots qui seront d'abord comptés ou rassemblés en divers agencements structuraux, et qui serviront de pivot pour l'analyse des phénomènes internes aux mots. Dans cette section, nous voulons illustrer comment se construit, par couches successives, le corpus et ses annotations, suggérant que des outils distincts pourraient être utilisés pour gérer ces couches, selon leur nature. Pour ce faire, nous procèderons par des exemples.

Pour les fins du modèle, nous utiliserons la balise *w* du TEI pour exprimer les unités (token) de cette partition de base du flux de caractères. C'est ce modèle TEI simplifié qui a été proposé, sous le nom de *propositions Sacacomie*, par le réseau ATONET. Voici un exemple d'une version tramée du texte de Borges dépouillée de toute annotation.



Version numérisée tramée du texte en format TEI (*fichier borges_trame.xml*). (7.4a, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt>
<title>LE CAPTIF : Borges, El Hacedor. Traduction de J.-M. Adam</title>
</titleStmt>
<publicationStmt> <p>Publié par...</p></publicationStmt>
 <notesStmt>
   <note>Des annotations structurelles et analytiques sur le texte figurent dans des fichiers
séparés.</note>
  </notesStmt>
```

```
<sourceDesc> <bibl>...</bibl></sourceDesc>
</fileDesc>
<encodingDesc>
<refsDecl>
<p>Texte tramé n'utilisant que des balises w et p. La présence d'un blanc entre w est une
représentation normalisée de la mise en forme du texte original.</p>
</refsDecl>
</encodingDesc>
</teiHeader>
<text>
<body>
<p><w xml:id="w2">À</w> <w xml:id="w3">Junín</w> <w xml:id="w4">ou</w> <w
xml:id="w5">à</w> <w xml:id="w6">Tapalqué</w><w xml:id="w7">,</w> <w
xml:id="w8">on</w> <w xml:id="w9">raconte</w> <w xml:id="w10">l'</w><w
xml:id="w11">histoire</w> <w xml:id="w12">suivante</w><w xml:id="w13">.</w> <w
xml:id="w14">Un</w> <w xml:id="w15">enfant</w> <w xml:id="w17">disparut</w> <w
xml:id="w18">après</w> <w xml:id="w19">un</w> <w xml:id="w20">raid</w> <w
xml:id="w21">d'</w><w xml:id="w22">Indiens</w> <w xml:id="w23">;</w> <w
xml:id="w24">on</w> <w xml:id="w25">dit</w> <w xml:id="w26">que</w> <w
xml:id="w27">les</w> <w xml:id="w28">Indiens</w> <w xml:id="w30">l'</w><w
xml:id="w31">avaient</w> <w xml:id="w32">enlevé</w><w xml:id="w33">.</w> <w
xml:id="w34">Ses</w> <w xml:id="w35">parents</w> <w xml:id="w36">le</w> <w
xml:id="w37">cherchèrent</w> <w xml:id="w38">inutilement</w> <w xml:id="w39">;</w>
<w xml:id="w40">des</w> <w xml:id="w42">années</w> <w xml:id="w43">plus</w> <w
xml:id="w44">tard</w><w xml:id="w45">,</w> <w xml:id="w46">un</w>
 <!-- ...-->
</p>
<!-- *{(Jorge Luis Borges, El Hacedor (J 960). Traduction de J.-M. Adam, Adam2005:203-204)}
 -->
</body>
</text>
</TEI>
```

Et voici ce que pourrait donner une première version de l'information structurelle construite sur la version épurée du texte de Borges. Pour rendre le document plus lisible, nous avons mis en commentaire la transcription non balisée du contenu textuel référencé à travers les balises <*span*>. Dans cet exemple, nous reprenons temporairement les balises <*milestone/*> de la représentation embarquée du corpus pour montrer le passage direct vers une forme débarquée d'annotation, sachant, comme nous l'avons montré au chapitre précédent, qu'il y a d'autres façons de procéder pour l'annotation débarquée.



Annotation structurelle (avec balises frontières référentielles et analytiques) sur la trame de Borges (fichier *borges_struct_v1.xml*). (7.4b, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
      <teiHeader>
        <fileDesc>
          <titleStmt>
            <title>LE CAPTIF : Borges, El Hacedor. Traduction de J.-M. Adam</title>
          </titleStmt>
          <publicationStmt> <p>Publié par...</p></publicationStmt>
          <notesStmt>
            <note>Annotations structurelle sur le document borges_trame.xml.</note>
          </notesStmt>
          <sourceDesc> <p> ... </p></sourceDesc>
        </fileDesc>
        <encodingDesc>
          <refsDecl>
   <p>Les balises «milestone n="valeur-de propriété" unit="nom-de-propriété"» concernent les mots qui suivent la balise jusqu'à l'apparition d'un nouveau « milestone » de même «unit».</p>
   <p>Les références de pagination utilisent les balises pb (début de page), lb(début de ligne) et l'attribut n de la balise w (word).</p>
   <p>milestone Én symbol "él" "é2a" "é2b" "é3a" "é3b" "é4a" "é4b" "é4c" "é4d" "é5a" "é5b" "é6a" "é6b" "é7a" "é7b" "é8a" "é8b" "é8e" "é8d" "é9a" "é9b" "é9c" "él0a" "él0b" "él0c" "élla" "éllb" "é11c" "é12a" "é12b" "é12c" "é12d" "é12e" "é12f" "é12g" "é12f fin"</p>
          </refsDecl>
        </encodingDesc>
      </teiHeader>
      <text>
        <body>

          <pb n="borges_adam/203"/>
          <p xml:base="borges_trame.xml"><lb n="1"/><milestone unit="Én" n="él"/>
            <span  from="#w2" to="#w13"/>
            <!--À Junín ou  à Tapalqué, on raconte l'histoire suivante.-->
            <milestone unit="Én" n="é2a"/>
            <span from="#w14" to="#w15"/>
            <!--Un enfant -->
            <lb n="2"/>
            <span from="#w17" to="#w23"/>
            <!--disparut  après un raid d'Indiens ; -->
            <milestone unit="Én" n="é2b"/>
            <span  from="#w24" to="#w33"/>
            <!--on dit que les Indiens l'avaient enlevé.-->
            <lb n="3"/>
            <span from="#w34" to="#w39"/>
            <!--Ses parents  le cherchèrent inutilement ;-->
            <milestone unit="Én" n="é3a"/>
            <span from="#w40" to="#w46"/>
            <!--des années plus tard, un -->
            <!-- ...-->
          </p>
        </body>
      </text>
    </TEI>
```

En utilisant une base de données XML supportant Xquery, on pourrait faire une requête pour obtenir les divers segments (*span*) englobant une occurrence (*w*) donnée ou un empan donné,

comme une concordance, une page, etc. en filtrant sur le contenu des attributs *from* et *to*. Ce qui nous intéresse ici, ce sont les segments et les arbres qui dominent une suite de mots préalablement sélectionnés, s'ils y a lieu. Les balises <*w*> n'ont pas à faire partie des objets à rapporter puisque l'information qu'elles portent se trouve déjà dans la représentation matricielle de SATO ou son équivalent XML en termes de structures de traits. La syntaxe XML nous oblige à utiliser des identifiants symboliques comme valeurs des attributs *from* et *to*. On comprendra cependant qu'ils seront associés de façon bijective aux valeurs entières représentant les numéros d'occurrences, ou d'entrées lexicales du texte en format numérique. Une représentation XML explicite de ces numéros en termes d'attributs à valeurs entières permettrait par ailleurs l'utilisation des opérateurs de comparaison numérique standard (plus grand, plus petit, égal, différent) de *XPath* et *XQuery*.

On remarquera aussi dans notre exemple que les balises <*milestone*/>, servent de repères pour un découpage en segments faisant partie d'une analyse, comme illustré dans le chapitre Linguistique textuelle et TEI. Nous les nommerons *balises frontières analytiques* pour les distinguer des *balises frontières référentielles* indiquant la référence de l'édition électronique à la forme de l'édition papier en termes de pages et de lignes.

Comme les balises frontières analytiques sont un point de vue sur le texte, il serait pertinent de les dégager de la structure formelle en termes de pages, paragraphes et lignes.



Annotation structurelle (avec balises frontières référentielles) sur la trame de Borges (fichier *borges_struct_v2.xml*). (7.4c, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
   <teiHeader>
      <fileDesc>
         <titleStmt>
            <title>LE CAPTIF : Borges, El Hacedor. Traduction de J.-M. Adam</title>
         </titleStmt>
         <publicationStmt>
            <p>Publié par...</p>
         </publicationStmt>
         <notesStmt>
            <note>Annotations structurelle sur le document borges_trame.xml.</note>
         </notesStmt>
         <sourceDesc>
            <p> ... </p>
         </sourceDesc>
      </fileDesc>
```

```
        <encodingDesc>
          <refsDecl>
            <p>Les références de pagination utilisent les balises pb (début de page), lb(début
              de ligne) et l'attribut n de la balise w (word).</p>
          </refsDecl>
        </encodingDesc>
      </teiHeader>
      <text>
        <body>
          <div type="texte" xml:base="borges_trame.xml">
            <pb n="borges_adam/203"/>
            <p>
              <lb n="1"/>
              <span from="#w2" to="#w15"/>
              <!-- À Junín ou  à Tapalqué, on raconte l'histoire suivante. Un enfant -->
              <lb n="2"/>
              <span from="#w17" to="#w33"/>
              <!-- disparut  après un raid d'Indiens ;  on dit que les Indiens l'avaient enlevé. -->
              <lb n="3"/>
              <span from="#w34" to="#w46"/>
              <!-- Ses parents  le cherchèrent inutilement ; des années plus tard, un -->
              <!-- ...-->
            </p>
          </div>
          <spanGrp xml:id="Seg1" type="Segmentation" ana="#Énoncé"
xml:base="borges_adam.xml">
            <span from="#w2" to="#w13" xml:id="é1">À Junín ou à Tapalqué, on raconte
l'histoire  suivante.</span>
            <span from="#w14" to="#w23" xml:id="é2a">Un enfant disparut après un raid
d'Indiens ;</span>
            <span from="#w24" to="#w33" xml:id="é2b">on dit que les Indiens l'avaient
enlevé.</span>
            <span from="#w34" to="#w39" xml:id="é3a">Ses parents le cherchèrent
inutilement ;</span>
            <!-- ...-->
          </spanGrp>

        </body>
      </text>
    </TEI>
```

Même s'ils sont utilisés pour représenter la structure formelle du texte, les balises frontières référentielles <*lb/*> et <*pb/*> sont aussi de type *milestone*, ce qui ne facilite pas la fouille arborescente. Pour retrouver la page qui contient un mot, on doit trouver dans la division *texte* le <*span*> qui contient le mot. La requête devra ensuite sélectionner la première balise <*pb*> à la gauche du <*span*>. Le nombre de nœuds à parcourir pourrait être assez important. C'est l'inconvénient des balises frontières. En effet, dans la représentation arborescente des documents XML, l'exploitation des balises frontières implique qu'on explore l'arbre horizontalement plutôt que verticalement. Or, la sémantique derrière ces balises <*pb/*> et

<*lb/*> comprend implicitement une notion d'emboitement. Ainsi, <*pb/*>, qui marque la page, contient <*lb/*> qui marque la ligne. La page est elle-même partie d'un texte dans le corpus. On pourrait donc expliciter cette hiérarchie par un emboitement explicite de <*span*> de pages et de <*span*> de lignes.

```
<div type="pagination"  xml:base="borges_adam.xml">
        <span type ="page" from="#w2" to "#w148" n="borges_adam/203">
                <spanGrp xml:id="Seg1" type="Segmentation" ana="#Ligne">
                <span from="#w2" to="#w15" n="1"/>
                <span from="#w17" to="#w33" n="2"/>
                <span from="#w34" to="#w46" n="3"/>
                </spanGrp>
        </span>
</div>
```

On pourrait aussi s'en tenir à une approche purement syntaxique sans interprétation de la sémantique d'emboitement de ces éléments <*pb/*> et <*lb/*> dérivés de la syntaxe des <*milestone*>. Dans ce cas, chaque groupe de <*milestone*> serait traduit en <*spanGrp*> comme on l'a fait pour les énoncés.

```
<spanGrp xml:id="Seg1" type="Segmentation" ana="#Page" xml:base="borges_adam.xml">
        <span type ="page" from="#w2" to="#w148" n="borges_adam/203"/>
</spanGrp>

 <spanGrp xml:id="Seg2" type="Segmentation" ana="#Ligne" xml:base="borges_adam.xml">
        <span from="#w2" to="#w15" n="1"/>
        <span from="#w17" to="#w33" n="2"/>
        <span from="#w34" to= "#w46" n="3"/>
</spanGrp>
```

Mais, on pourrait préférer la première représentation en ce qu'elle permet d'expliciter syntaxiquement la nature hiérarchique d'emboitement des lignes dans les pages. Finalement, dans notre exemple trop simple, la seule structure explicitement hiérarchique est la structure en paragraphe. Voici une troisième version du document d'annotation dépouillée de toutes ses balises frontières.



Annotation structurelle (sans balises frontières) sur la trame de Borges (fichier *borges_struct_v3.xml*). (7.3d, exemple)

```
<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
   <teiHeader>
```

```
        <fileDesc>
          <titleStmt>
            <title>LE CAPTIF : Borges, El Hacedor. Traduction de J.-M. Adam</title>
          </titleStmt>
          <publicationStmt>
            <p>Publié par...</p>
          </publicationStmt>
          <notesStmt>
            <note>Annotations structurelle sur le document borges_trame.xml.</note>
          </notesStmt>
          <sourceDesc>
            <p> ... </p>
          </sourceDesc>
        </fileDesc>
        <encodingDesc>
          <refsDecl>
            <p></p>
          </refsDecl>
        </encodingDesc>
      </teiHeader>
      <text>
        <body>
          <p xml:base="borges_adam.xml">
              <span from="#w2" to="#w46"/>
          </p>

          <spanGrp xml:id="Seg1" type="Segmentation" ana="#Page"
xml:base="borges_adam.xml">
            <span type="page" from="#w2" to="#w148" n="borges_adam/203"/>
          </spanGrp>

          <spanGrp xml:id="Seg2" type="Segmentation" ana="#Ligne"
xml:base="borges_adam.xml">
            <span from="#w2" to="#w15" n="1"/>
            <span from="#w17" to="#w33" n="2"/>
            <span from="#w34" to="l#w46" n="3"/>
          </spanGrp>

          <spanGrp xml:id="Seg3" type="Segmentation" ana="#Énoncé"
xml:base="borges_adam.xml">
            <span from="#w2" to="#w13" xml:id="é1">À Junín ou à Tapalqué, on raconte l'histoire
              suivante.</span>
            <span from="#w14" to="#w23" xml:id="é2a">Un enfant disparut après un raid d'Indiens
              ;</span>
            <span from="#w24" to="#w33" xml:id="é2b">on dit que les Indiens l'avaient
              enlevé.</span>
            <span from="#w34" to="#w39" xml:id="é3a">Ses parents le cherchèrent inutilement
              ;</span>
            <!-- ...-->
          </spanGrp>

        </body>
      </text>
    </TEI>
```

Dans l'exemple qui suit, nous intégrons dans un seul document externe toutes les annotations sur la trame de Borges.



Annotation cumulative sur la trame de Borges (fichier *borges_struct_v4.xml*). (7.4e, exemple)

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Analyse due à J.M. Adam du texte LE CAPTIF : Borges, El Hacedor. Traduction
          de J.-M. Adam</title>
      </titleStmt>
      <publicationStmt>
        <p>Publié par...</p>
      </publicationStmt>
      <notesStmt>
        <note><p>Annotations structurelles sur le document borges_trame.xml comprenant :
          <list>
            <item>la structure formelle (paragraphes, pagination et lignes physiques) ;</item>
            <item>la segmentation en énoncés et en phrases typographique ;</item>
            <item>l'analyse de la structure compositionnelle du récit (plan de texte) ;</item>
          </list>.</p></note>
      </notesStmt>
      <sourceDesc>
        <p> ... </p>
      </sourceDesc>
    </fileDesc>
    <encodingDesc>
      <refsDecl>
        <p/>
      </refsDecl>
    </encodingDesc>
  </teiHeader>
  <text>
    <body>
<!-- Structure formelle du corpus -->
      <p xml:base="borges_adam.xml">
        <span from="#w2" to="#w46"/>
      </p>

<!-- Structure référentielle du corpus -->
      <spanGrp xml:id="Seg1" type="Segmentation" ana="#Page"
xml:base="borges_adam.xml">
        <span type="page" from="#w2" to="#w148" n="borges_adam/203"/>
      </spanGrp>

      <spanGrp xml:id="Seg2" type="Segmentation" ana="#Ligne"
xml:base="borges_adam.xml">
        <span from="#w2" to="#w15" n="1"/>
        <span from="#w17" to="#w33" n="2"/>
        <span from="#w34" to="l#w46" n="3"/>
      </spanGrp>
```

```
<!-- Analyse de la structure compositionnelle -->
        <!-- Définition des catégories interprétatives -->
        <div type="Analyse" subtype="StructureCompositionnelle" xml:id="ana1">
          <interpGrp type="Unités_discursives">
            <interp xml:id="Énoncé">On considèrera comme énoncé...</interp>
            <interp xml:id="Phrase">On entendra par phrase typographique...</interp>
          </interpGrp>

          <interpGrp type="StructureCompositionnelle">
            <interp xml:id="plan_de_texte">Le plan du texte fait partie de la structure
              compositionnelle qui organise la cohésion d'une suite linéaire de séquences
              (Adam2005:chapitre 6).</interp>

            <interp xml:id="séquence">Les séquences sont des unités textuelles complexes,
              composées d'un nombre limité de paquets de propositions-énoncés. Elles
              constituent des réseaux relationnels hiérarchiques formant des entités
              relativement autonomes présentant des agencements dits narratifs,
              argumentatif, explicatif, dialogal, etc. (Adam2005:chapitre 5). </interp>

            <interp xml:id="SCséquence_narrative">La séquence narrative est... .</interp>

            <interp xml:id="SCsituation_initiale">La situation initiale est un des
              composants de la séquence narrative...</interp>

            <interp xml:id="SCnoeud">Le noeud est un des composants de la séquence
              narrative...</interp>

            <interp xml:id="SCaction">L'action (ou la réaction) est un des composants de la
              séquence narrative...</interp>

            <interp xml:id="SCdénouement">Le dénouement est un des composants de la séquence
              narrative...</interp>

            <interp xml:id="SCsolution_finale">La solution finale est un des composants de
              la séquence narrative...</interp>

            <interp xml:id="SCpériode">Selon Aristote, la période est une forme d'élocution
              qui renferme en elle-même un commencement et une fin, ainsi qu'une étendue
              qui se laisse embrasser d'un coup d'oeil» Rhétorique III, cité par Adam
              2006, p. 141. </interp>

            <interp xml:id="SCpériode_interprétative">Explication... . </interp>

            <interp xml:id="SCpériode_narrative">Explication... . </interp>

            <interp xml:id="SCréférence">Explication... Adam:2005:86-97. </interp>

            <interp xml:id="I1">introduction</interp>
          </interpGrp>

          <!-- Segmentation du texte analysé en énoncés -->
          <spanGrp xml:id="Seg3" type="Segmentation" ana="#Énoncé"
xml:base="borges_adam.xml">
            <span from="#w2" to="#w13" xml:id="é1">À Junín ou à Tapalqué, on raconte
              l'histoire suivante.</span>
            <span from="#w14" to="#w23" xml:id="é2a">Un enfant disparut après un raid
```

d'Indiens ;
on dit que les Indiens l'avaient enlevé.
Ses parents le cherchèrent inutilement ;
des années plus tard, un soldat qui venait de l'intérieur leur parla d'un Indien aux yeux couleur de ciel qui pouvait bien être leur fils.
Ils le rencontrèrent enfin (
la chronique a perdu les circonstances
et je ne veux pas inventer ce que je ne sais pas)
et ils crurent le reconnaître.
L'homme, marqué par le désert et la vie sauvage, ne comprenait déjà plus les mots de sa langue natale,
 mais, indifférent et docile, il se laissa conduire à la maison.
Il s'arrêta sur le seuil,
peut-être parce que les autres s'y arrêtèrent.
Il regarda la porte,
comme s'il ne la comprenait pas.
Soudain, il baissa la tête,">
poussa un cri,">
traversa en courant le corridor et les deux vastes cours">
et pénétra dans la cuisine.">
Sans hésiter, il plongea le bras dans la hotte enfumée
et sortit le petit couteau à manche de corne qu'il avait caché là,
lorsqu'il était enfant.
Ses yeux brillèrent de joie
et ses parents pleurèrent,
parce qu'ils avaient retrouvé leur fils.
Ce souvenir fut peut-être suivi par d'autres,
mais l'Indien ne pouvait vivre entre quatre murs
et un jour il partit à la recherche de son désert.
Moi je voudrais savoir
ce qu'il ressentit en cet instant de vertige
où le passé et le présent se confondirent ;
moi je voudrais savoir
si le fils perdu renaquit et mourut en cette extase,
ou s'il parvint à reconnaître,
ne fût-ce qu'à la manière d'un nouveau-né ou d'un chien,
les parents et la maison.

```
</spanGrp>

<!-- Segmentation du texte analysé en phrases -->
<spanGrp xml:id="Seg4" type="Segmentation" ana="#Phrase">
   <span from="#é1" xml:id="P1"/>
   <span from="#é2a" to="#é2b" xml:id="P2"/>
   <span from="#é3a" to="#é3b" xml:id="P3"/>
   <span from="#é4a" to="#é4d" xml:id="P4"/>
   <span from="#é5a" to="#é5b" xml:id="P5"/>
   <span from="#é6a" to="#é6b" xml:id="P6"/>
   <span from="#é7a" to="#é7b" xml:id="P7"/>
   <span from="#é8a" to="#é8d" xml:id="P8"/>
   <span from="#é9a" to="#é9c" xml:id="P9"/>
   <span from="#é10a" to="#é10c" xml:id="P10"/>
   <span from="#é11a" to="#é11c" xml:id="P11"/>
   <span from="#é12a" to="#é12f_fin" xml:id="P12"/>
</spanGrp>

<!-- Blocs de l'analyse compositionnelle -->

<div type="StructureCompositionnelle" ana="#SCsequence_narrative"
   xml:id="séquence_narrative_1" n="séquence narrative">
   <div type="StructureCompositionnelle" ana="#SCsituation_initiale" xml:id="Pn1">
      <span type="StructureCompositionnelle" from="#é2a" to="#é3a"
         xml:id="Pn1.é2a-é3a"/>
   </div>
   <div type="StructureCompositionnelle" xml:id="intrigue_1">
      <div type="StructureCompositionnelle" ana="#SCnoeud" xml:id="Pn2">
         <span type="StructureCompositionnelle" from="#é3b" to="#é4a"
            xml:id="Pn2.é3b-é4a"/>
         <span type="StructureCompositionnelle" from="#é4d" xml:id="Pn2.é4d"/>
      </div>
      <div type="StructureCompositionnelle" ana="#SCaction" xml:id="Pn3">
         <span type="StructureCompositionnelle" from="#P5" to="#P7"
            xml:id="Pn3.P5-P7">(Ré)Action</span>
      </div>
      <div type="StructureCompositionnelle" ana="#SCdénouement" xml:id="Pn4">
         <span type="StructureCompositionnelle" from="#P8" to="#P9"
            xml:id="Pn4.P8-P9"/>
      </div>
   </div>
   <div type="StructureCompositionnelle" ana="#SCsolution_finale" xml:id="Pn5">
      <span type="StructureCompositionnelle" from="#P10" xml:id="Pn5.P10"/>
   </div>
</div>

<div type="StructureCompositionnelle" ana="#SCsequence_narrative"
   xml:id="entrée-préface">
   <span type="StructureCompositionnelle" from="#é1" xml:id="Pn0">Cadre
      médiatif</span>
   <span type="StructureCompositionnelle" from="#é4b" to="#é4c" xml:id="Pn0a"
      >Évaluation commentative</span>
</div>

<div type="StructureCompositionnelle" ana="#SCpériode_argumentative"
   xml:id="P11_argumentative">
   <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-proposition_p"
```

```
                >premier argument </span>
              <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-proposition_q"
                >second argument</span>
              <span type="StructureCompositionnelle" from="#é11c"
                xml:id="é11c-conclusion_non_c">renversement de la conclusion implicite du
                retour définitif à la maison</span>
            </div>

            <div type="StructureCompositionnelle" ana="#SCpériode_narrative"
              xml:id="P11_narrative">
              <span type="StructureCompositionnelle" from="#é11a" xml:id="é11a-Pn1">Situation
                initiale Pn1</span>
              <span type="StructureCompositionnelle" from="#é11b" xml:id="é11b-Pn2">Noeud
                Pn2</span>
              <span type="StructureCompositionnelle" from="#é11c" xml:id="é11c-
Pn4">Dénouement
                Pn4</span>
            </div>

            <div type="StructureCompositionnelle" ana="#SCpériode"
xml:id="simple_période_P11">
              <alt mode="incl" targets="#P11_argumentative #P11_narrative" weights="0.5 0.5"/>
            </div>

            <div type="StructureCompositionnelle" ana="#SCpériode" xml:id="évaluation_finale">
              <span type="StructureCompositionnelle" from="#P12" xml:id="PnΩ">Évaluation
                finale. «Cette prose périodique dominée par le rythme contribue au
                glissement de genre du récit factuel au récit poétique.» (Adam 2005:
                211)</span>
            </div>

            <!-- Bloc supérieur : plan du texte -->

            <div type="StructureCompositionnelle" ana="#SCplan_de_texte"
              xml:id="plan_de_texte_du_Captif">
              <ab>
                <ptr target="#entrée-préface"/>
                <ptr target="#séquence_ narrative_1"/>
                <ptr target="#simple_période_P11"/>
                <ptr target="#évaluation_finale"/>
              </ab>
            </div>
          </div>

          <div  type="Analyse" ana="#référence" xml:base="borges_adam.xml"
xml:id="référent_personnage_principal">
            <p> La question de l'identité narrative...
              ... L'amorce de la chaîne par <span from="#w14" to="#w15">un enfant</span>
              ... pronominalisations ...<span from="#w30">l'</span>
                ... <span from="#w36">le</span>
                ... Mais une nouvelle chaîne <span from="#w57" to="#w63">un Indien aux yeux
couleur de ciel</span>
                ... pronom ...<span from="#w73">le</span>
                ... <span from="#w101">le</span>
                ... hyperonyme... <span from="#w104" to="#w105">l'homme</span>
                .. reprises pronominales neutres en «il» …
                <spanGrp>
```

```
                        <span from="#w135"/>
                        <span from="#w144"/>
                        <span from="#w168"/>
                        <span from="#w177"/>
                        <span from="#w207"/>
                        <span from="#w226"/>
                        <span from="#w232"/>
                    </spanGrp>
                    ... <span from="#w253" to="#w254">leurs fils</span>
                    ...
                </p>
            </div>
        </body>
    </text>
</TEI>
```

Cette dernière version du document d'annotation contient à la fois des annotations structurelles liées à l'édition électronique du texte et d'autres qui traduisent un point de vue analytique sur le texte. Ces annotations pourraient être consignées dans des documents indépendants. Ainsi, l'annotation structurelle directement reliée à l'établissement de l'édition électronique, pourrait être consignée dans un document d'annotation qu'on pourrait qualifier de *primaire* dans la hiérarchie de dépendance des annotations — voir diagrammes UML de Salmon-Alt, Romary et Pierrel (2004). Les annotations analytiques pourraient être considérées comme des points de vue consignés dans des documents indépendants que l'analyste pourrait intégrer ou non dans son plan d'analyse. Cette modularité est permise par l'organisation de l'annotation structurelle en documents et structures distinctes pouvant ou non entretenir des relations de dépendance. D'un point de vue documentaire, ces dépendances devraient être définies dans le référentiel de données sous la forme, par exemple, de relations RDF (cf. Daoust et coll. 2008).

Cette façon de découper l'annotation structurale en sections ou en documents, avec des éléments qui font référence au document annoté, lève une partie importante de la motivation derrière l'approche algébrique, à savoir la nécessité de faire coexister une multiplicité de schémas d'annotation. Ces documents pourraient être gérés par des systèmes de base de données XML dont beaucoup se présentent comme indépendants des schémas. C'est le cas notamment du *Berkeley DB XML* qui permet de gérer des *contenants* qui sont des collections de documents XML avec des structures différentes, même si les index d'optimisation des requêtes trouveront avantage à gérer des documents partageant des éléments.

Comme les schémas décrivant les structures d'annotation structurelle vont varier selon le type d'annotation impliqué, les *patrons de fouille* des propriétés structurelles dépendront, comme pour les propriétés atomiques de SATO, du type et du vocabulaire spécifique de chaque structure d'annotation.

Les requêtes sur l'annotation structurelles peuvent avoir plusieurs types de résultats. Comme dans le cas des propriétés atomiques de SATO et des opérateurs algébriques proposés par Clarke, Cormack et Burkowski, le résultat du filtrage pourrait se traduire par la sélection d'un ensemble de contextes (ou de formes lexicales dans le cas d'une annotation lexicale) satisfaisant les contraintes de chaque propriété et de l'expression qui combinent ces contraintes par divers opérateurs. Comme on l'a vu, c'est aussi le cas de l'expérimentation de Ghodke et Bird (2010) qui ramène des phrases. Pour les filtrages impliquant des segments textuels, la conjonction s'interprète comme le dépistage des empans textuels contenus dans les segments rapportés par chacune des requêtes. La requête pourrait aussi avoir pour objectif de ramener le nombre de structures contenues ou en intersection avec un sous-texte ou un contexte donné. On pourrait également être intéressé à obtenir l'ensemble des arbres ou types d'arbres qui satisfont les contraintes spécifiées.

Par rapport aux opérations textométriques, du moins celles que l'on trouve dans SATO, l'annotation structurelle se présente essentiellement comme une couche supplémentaire de description des données qui s'intègre dans le cadre expérimental existant : recherche de contextes, décomptes d'objets dans des contextes, typologie des objets avec leur répartition dans le corpus ou des sous-textes. Les structures dépistées peuvent être interprétées en tant que telles. Elles peuvent servir à guider des parcours dans le texte avec navigation hypertextelle. Elles peuvent également servir d'intrants à divers calculs statistiques au même titre qu'un nombre d'occurrences, de formes lexicales ou de valeurs catégorielles.

L'enjeu de l'implantation des segments dynamiques et de l'annotation structurelle ne vise donc pas à changer de paradigme d'analyse mais plutôt à développer une couche logicielle spécifique s'ajoutant au traitement existant des unités atomiques dans leurs dimensions lexicales et contextuelles. C'est aussi cette couche logicielle qui doit prévoir la gestion des liens (balises <link>) permettant la création de liens hypertextuels, par exemple entre la note et son appel.

Au niveau de l'affichage des contextes, ce qui peut être présenté directement, ce sont les frontières des divers segments référés par les éléments contenant des pointeurs vers le texte annoté. En d'autres termes, on peut recréer l'équivalent des *milestone TEI* lors de l'affichage des mots en contexte. Une certaine forme de mise à plat de la structure pourrait être envisagée à titre de description de la borne de chacun des segments. Ainsi, dans l'exemple qui suit, on donne le chemin vers l'élément terminal de la structure *struct_compo* en concaténant les identificateurs de nœds (*xml-id*), séparés par des /, jusqu'au commentaire final, contenu explicatif du de la balise *span*, mis entre parenthèses.

<milestone unit="struct_compo" n="plan_de_texte_du_Captif/entrée-préface/Pn0 (Cadre médiatif)"/>

Dans la syntaxe actuelle de SATO, cette frontière de segment pourrait s'exprimer sous forme de propriété libre ayant un caractère informatif.

**struct_compo="plan_de_texte_du_Captif/entrée-préface/Pn0 (Cadre médiatif)"*

Dans notre exemple, on a utilisé la séquence des attributs *xml:id* suivie, entre parenthèses, du contenu explicatif du <*span*> terminal. Mais, d'autres attributs plus explicatifs auraient pu être choisis pour signaler l'apparition d'un segment analysé. Au-delà de ce signalement, il s'agirait de produire à la volée les représentations arborescentes dans une fenêtre d'annotation, comme on le fait actuellement dans l'ergonomie de SATO pour les propriétés atomiques. On pourrait, au choix de l'utilisateur, afficher la structure en XML ou sa représentation graphique. Évidemment, la multiplicité des structures d'analyse pourrait générer un affichage difficile à lire. Mais, comme pour les propriétés actuelles de SATO, l'usager pourrait sélectionner les propriétés qu'il désire afficher, sachant qu'en cliquant sur un mot, toute l'information serait révélée dans la fenêtre prévue à cet effet.

Ces considérations nous amènent donc, tout naturellement, à esquisser quelques hypothèses concernant les interfaces possibles pour manipuler les annotations structurelles.

7.5 La question des interfaces usagers.

Parallèlement à la recherche sur les langages de requête sur les documents XML et les forêts d'arbres, et parallèlement aussi à la recherche sur les algorithmes d'implantation de ces langages, un certain nombre de publications abordent la question des interfaces permettant à

l'utilisateur de formuler des requêtes de façon relativement naturelle. On conviendra en effet que la maitrise de beaucoup des langages de requête proposés est difficile pour les non-spécialistes. Quand on considère que la seule maitrise des expressions rationnelles est déjà une difficulté pour un nombre significatif des utilisateurs de systèmes d'analyse de texte par ordinateur, on conviendra que l'apprentissage des formalismes de requêtes sur les arbres puisse poser un réel problème, même dans le cas des langages qui s'expriment sur la base plus intuitive de parcours dans des arbres (XPath et ses dérivés).

L'utilisation d'interfaces graphiques permettant de dessiner des prototypes d'arbres à chercher est une des voies proposées pour permettre à l'usager de formuler des requêtes sans maitriser le langage de requêtes sous-jacent. Ces approches sont inspirées de leur équivalent dans le domaine des bases de données relationnelles : *Query by Example* (Zloof, 1977). Un exemple est *XQBE* (Braga et coll. 2005a).

> XQBE is based on the use of trees, coherent with the hierarchical XML data model. XQBE was designed with both the objectives of being intuitive and of being directly mapped to XQuery, thus representing a GUI capable of running on top of any XQuery engine.

On trouvera aussi une adaptation de XQBE destinée à servir d'interface graphique pour la génération de feuilles XSLT (Braga et coll. 2005b). Bird et Lee ont aussi proposé ce type d'interface dans une formule taillée sur mesure pour la fouille d'arbres linguistiques (Bird et Lee 2007)

> We describe a new approach to tree query which we call “Query by Annotation”. Users express a query by annotating a tree, and the annotation is compiled into an expression in a path language. The result trees are overlaid with the original query, permitting the user to see why they match. Since queries and results are annotated trees, users can easily refine and resubmit their queries. (Bird et Lee 2007)

Même si Bird et Lee s'appuient sur *LPath*, l'approche *Query by Annotation* (*QBA*) est, selon leurs dires, indépendante du langage de référence. Avec *QBA*, la requête et les résultats empruntent un même formalisme d'arbres annotés, ce qui facilite la mise au point des requêtes. Les auteurs proposent une approche graphique qui génère une requête LPath envoyés sur un serveur qui le traduit en SQL et retourne les arbres repérés. QBA est implanté en Python et PyQt. NLTK (Bird et coll., 2009) est utilisée pour compiler les requêtes LPath.

Dans le cadre du *Prague Dependency Treebank 2.0* (Haji. et coll., 2006), abrégé sous le nom de *PDT 2.0*, Pajas et Štěpánek (2009) présentent leur système d'interrogation de corpus annotés syntaxiquement.

> The system consists of a powerful query language with natural support for cross-layer queries, a client interface with a graphical query builder and visualizer of the results, a command-line client interface, and two substitutable query engines: a very efficient engine using a relational database (suitable for large static data), and a slower, but paralel-computing enabled, engine operating on treebank files (suitable for "live" data).

Tout en reprenant un certain nombre de propositions de Bird, en particulier le modèle relationnel et de représentation des nœuds pour les grands corpus, les chercheurs tchèques présentent un système intégré en code ouvert qui offre plusieurs solutions originales, notamment au niveau de l'interface. Il faut dire que le modèle de données pour la représentation des arbres de *PDT 2.0* est particulièrement élaboré et comporte des caractéristiques qui peuvent se rapprocher des données en ATO, notamment l'annotation à couches multiples et multilingues. Le système s'appelle *PML Tree Query* (*PML-TQ*). Pour la gestion des arbres, le système propose deux système. Le premier suit le modèle relationnel de Bird. Il est réservé aux grandes banques stabilisées sur lesquelles on ne fait pas de mises à jour. Le système propose aussi un deuxième moteur de recherche sans index, plus lent pour la consultation, mais plus rapide pour la mise à jour. Ces données peuvent être traitées en parallèle sur des ordinateurs distincts. Ce moteur est donc consacré aux arbres en cours de construction.

Le langage d'interrogation s'appuie sur un modèle de données générique qui et un langage XML baptisé *Prague Markup Language* (PML) (Pajas et .Štěpánek 2006). Les auteurs indiquent qu'il est facile de convertir vers PML divers formats d'arbres, ce qui peut être fait à la volée par des transformations XSLT. Ainsi, l'éditeur graphique *TrEd*, conçu pour manipuler des données en PML, pourra être utilisé, à une conversion près sur divers types de formats de données. Pajas et .Štěpánek indiquent qu'il est aussi possible d'utiliser une feuille XSLT pour convertir un schéma exprimé en PML vers un langage plus standard comme Relax NG, ce qui permet aux outils de validation XML existants. Comme on le fait pour SATO, l'approche générale du PDT mise sur l'annotation en couches multiples utilisant l'annotation débarquée. La particularité du projet tchèque est l'adoption d'un langage spécifique de définition de

schéma, le PML, justifié, selon leur point de vue, par l'insuffisance des modèles existants, du moins dans leur version de l'époque.

> Rather than being targeted to one particular annotation schema or being a set of specifically targeted encoding conventions, PML is an open system, where a new type of annotation can be introduced easily by creating a simple XML file called PML schema, which describes the annotation by means of declaring the relevant data types and possibly assigning certain roles to these data types. The roles in the context of PML are just labels from a pre-defined set that can be used to mark the declarations according to their purpose. For instance, the roles indicate which data structures represent the nodes of the trees, how the node data structures are nested to form a tree, which field in a data structure carries its unique ID (if any), or which field carries a link to the annotated data or other layers of annotation, and so on. PML schema can define all kinds of annotations varying from linear annotations through constituency or dependency trees, to complex graph-oriented annotation systems. (Pajas et Štěpánek 2008)

Ce que l'on constate à la lecture des articles de Pajas et Štěpánek, c'est que PML est aussi intimement lié au fonctionnement des outils logiciels développés pour le PDT.

> The fundamental toolkit for PML comprises of a validator (based on compiling PML schemas to RelaxNG grammars accompanied by Schematron rules), and API, consisting of a Perl library (basic interfaces for Java and C++ are planned). The input/output functions of the library are modular and can work with local files as well as with remote resources accessible via HTTP, FTP or SSH protocols (with pluggable support for other protocols). Additionally, the library supports on the-fly XSLT-based format conversions that can be easily plugged in via a simple configuration file. Consequently, the API can transparently handle even non-PML data formats. (Pajas et Štěpánek 2009)

Dans cette chaine de traitement, un module très important du point de vue de l'interface usager est l'éditeur d'arbres *TrEd*.'

> The basic editing capabilities of TrEd allow the user to easily modify the tree structure with dragand-drop operations and to easily edit the associated data. Although this is sufficient for most annotation tasks, the annotation process can be

> greatly accelerated by a set of custom extension functions, called macros, written in Perl. Macros are usually created to simplify the most common tasks done by the annotators. They can be called either from menu or by keyboard shortcuts.

Les auteurs indiquent que TrEd est déjà utilisé par plusieurs projets en dehors de la PDT. Les fonctions de TrEd peuvent être appelées en mode commande et permettent l'écriture de *macros* sous forme de petits programmes Perl.

L'autre grand outil d'interface développé pour la PDT est *NetGraph* (Mírovský, 2006), le module graphique traditionnellement utilisé pour l'interrogation de la PDT. Cette application a l'avantage de se présenter sous la forme d'applet Java utilisable par le Web. Le nouvel outil d'interrogation *PML-TQ* fait, pour sa part, largement appel à l'éditeur *TrEd*.

Concernant l'annotation structurelle dans un contexte d'analyse de texte assistée par ordinateur, il est aussi possible de concevoir des modes d'utilisation simplifiés pour exploiter des schémas de données prédéterminées. Une interface par formulaires fournissant des champs prédéfinis de paramétrage pourrait permettre de faire basculer du côté serveur la tâche de production des requêtes *Xquery* ou autres. La mise en place de chaines de traitement simplifiées se présente donc d'abord comme l'écriture d'interfaces destinées à manipuler des structures de données particulières. Par exemple, si on prévoit une structure destinée précisément à représenter la chaine référentielle, il sera possible de prévoir des requêtes exploitant ces structures avec des contraintes spécifiques réduisant le jeu des possibilités offertes à l'utilisateur. Donc, au-delà de la généralité du moteur de gestion de l'annotation structurelle, on pourrait prévoir la mise au point de schémas d'annotation accompagnés d'interfaces d'interrogation spécifiques. L'usager qui voudrait utiliser un type déjà défini d'annotation structurelle pourrait appliquer des scénarios encadrant l'interface avec le moteur de recherche.

Comme on a pu l'entrevoir dans les paragraphes précédents, le dynamisme de la recherche sur les langages d'interrogation des arbres et les algorithmes d'optimisation trouve son pendant dans le domaine des interfaces graphiques susceptibles d'amoindrir les difficultés d'utilisation des langages eux-mêmes. Certes, le traitement en SATO des annotations structurelles, ne pose pas les mêmes défis que la mise au point et l'interrogation de millions d'arbres linguistiques. En fait, notre défi est davantage d'intégrer harmonieusement cette nouvelle couche d'annotation à celle, largement plus coutumière en ATO, de l'annotation à plat faisant appel à

un langage de requête simple et pour lequel, par conséquent, l'interface graphique est moins nécessaire. Dans la mesure, cependant, où l'on voudra conserver la syntaxe actuelle du filtrage SATO, il faudra trouver une façon de la combiner avec le nouveau formalisme de fouille des structures, tant sur le plan de la syntaxe du langage de commande que sur le plan de l'intégration des modules informatiques respectifs.

Bibliographie du chapitre 7

Adam, 2005. Adam, J.-M. *La linguistique textuelle, Introduction à l'analyse textuelle des discours*. Armand Colin, Paris ISBN 2-220-26752-5.

ATONET, 2005. *Réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur* (ATONET) : http://www.atonet**.**net

Bentley, 1977. Bentley, J. L. Solutions to Klee's rectangle problems. Tech. report, Carnegie-Mellon Univ., Pittsburgh, PA, 1977.

Bird et coll., 2009. Bird, Steven; Klein, Ewan; Loper. Edward; *Natural Language Processing with Python*. O'Reilly, 2009. http://www.nltk.org/book.

Bird et Lee, 2007. Bird, Steven; Lee, Haejoong. Graphical query for linguistic treebanks. In *10th Conference of the Pacific Association for Computational Linguistics,* pages 22–30, 2007.

Bird et coll., 2006. Bird, Steven ; Chen, Yi; Davidson, Susan; Lee, Haejoong; Zheng, Yifeng. Designing and evaluating an Xpath dialect for linguistic queries. In *22nd International Conference on Data Engineering (ICDE)*, pages 52–61, 2006.

Bird et Liberman, 2001. Bird, S. and Liberman, M. A formal framework for linguistic annotation. *Speech Communication,* 33:23-60. Speech Communication, Jan. 2001

Boncz et coll. 2006. P. A. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, J. Teubner. MonetDB/XQuery: A Fast XQuery Processor Powered by a Relational Engine. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Chicago, IL, USA, June 2006.

Braga et coll., 2005a. Braga, Daniele; Campi, Alessandro; Ceri, Stefano. XQBE (XQuery By Example): A visual interface to the standard XML query language. ACM Transactions on Database Systems, 30(2):398–443.

Braga et coll., 2005b. Braga, Daniele; Campi, Alessandro; Cappa, Roberto; Salvi, . XSLT By Example. *Special interest tracks and posters of the 14th international conference on World Wide Web*. Pages: 1158 – 1159. ISBN:1-59593-051-5

Buneman et coll. 2000. Buneman, Peter; Fernandez, Mary F.; Suciu, Dan. UnQL: a query language and algebra for semistructured data based on structural recursion. *VLDB Journal: Very Large Data Bases*, Vol. 9, No. 1. pp. 76-110.]

Chazelle, 1986. Chazelle, B. Filtering search: A new approach to query answering. *SIAM J.* Comput. 15(3), 703–724 (1986)

Clark et DeRose, 1999. Clark, James ;DeRose, Steve. *XML Path language (XPath)*. W3C, 1999. http://www.w3.org/TR/xpath.

Clarke, Cormack, Burkowski, 1994. C.L.A. Clarke, G.V. Cormack, and F.J. Burkowski, *An algebra for structured text search and a framework for its implementation*, Department of Computer Science, University of Waterloo, Canada, Technical Report CS-94-30, August 1994.

Daoust, 2009. Daoust F. Système d'analyse de texte par ordinateur, SATO, Manuel de référence, version 4.3. Centre d'analyse de texte par ordinateur, UQAM, 2007; modifié en 2009. http://www.ling.uqam.ca/sato/satoman-fr.html

Daoust et coll. 2008. Daoust, F.; Duchastel, J.; Marcoux, Y.; Rizkallah, E. JADT-2008. Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche, in *Actes des JADT-2008*, vol. 1, pp- 355-367, Presses universitaires de Lyon, 2008. ISBN 978-2-7297-0810-8. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2008/pdf/daoust-duchastel-marcoux-rizkallah.pdf

Daoust et Marcoux, 2006. Daoust F. et Marcoux Y. Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés, in *Les Cahiers de la MSH Ledoux no. 3, Actes des JADT-2006*, vol. 1, pp 327-340, Presses universitaires de Franche-Comté, 2006. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/029.pdf

Edelsbrunner, H. 1980. Dynamic data structures for orthogonal intersection queries. Tech. Report F59, Inst. Informationsverarb., Tech. Univ. Graz, 1980.

Fleury, 2009. Fleury, S. *Le métier textométrique* (Trameur). Centre de textométrie – CLA^{2}T, U. Paris 3 Sorbonne nouvelle, http://tal.univ-paris3.fr/trameur/

Ghodke et Bird, 2010. Ghodke, S., & Bird, S. Fast query for large treebanks. In Human Language Technologies: *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, USA. Handle 10187/7142 [http://repository.unimelb.edu.au/10187/7142]

Haji et coll. 2006. Haji, Jan et coll. *The Prague Dependency Treebank 2.0*. CD-ROM. Linguistic Data Consortium (CAT: LDC2006T01).

Hanson et Johnson, 1992. Hanson, E. N. ; Johnson, T. The Interval Skip List: A Data Structure for Finding All Intervals That Overlap a Point. *Lecture Notes in Computer Science*, Volume 519/1991, Springer Berlin / Heidelberg. ISSN 0302-9743 (Print) 1611-3349 (Online).

Kepser, 2003 Stephan Kepser.. Finite structure query: A tool for querying syntactically annotated corpora. In *EACL 2003: The 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 179–186, 2003.

König et Lezius, 2001. E. König and W. Lezius. The TIGER language - a description language for syntax graphs. part 1: *User's guidelines. Technical report*, University of Stuttgart, Stuttgart, Germany, 2001. URL citeseer.ist.psu.edu/article/knig01tiger.html

Lai et Bird, 2009. Lai, C. and Bird, S. Querying Linguistic Trees. Journal of Logic, Language and Information. http://repository.unimelb.edu.au/10187/3307

Lai, 2005. Lai, Catherine. *A Formal Framework for Linguistic Tree Query*. Master's thesis, Department of Computer Science and Software Engineering, University of Melbourne, Australia, 2005.

Lai et Bird, 2004. Lai, C. and Bird, S. Querying and Updating Treebanks: A Critical Survey and Requirements Analysis, in *Proceedings, Australasian Language Technology Workshop*, pages 139–146, Macquarie University, Sydney. December 2004 http://repository.unimelb.edu.au/10187/1547

Lucene. http://lucene.apache.org/

McCreight, 1980. McCreight, E. M. Efficient algorithms for enumerating intersecting intervals and rectangles. Tech. Report CSL-80-9, Xerox Palo Alto Res. Center, Palo Alto, CA, 1980.

McCreight, 1985. McCreight, E. M. Priority search trees. SIAM Journal on Computing, 14(2):257–276, 1985.

Meier, 2003. Meier. Wolfgang. eXist: An open source native XML database. In *Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems*, pages 169–183. Springer-Verlag.

Mírovský, 2006. Mírovský, Jirí. Netgraph: A tool for searching in prague dependency treebank 2.0. In Haji. c, Jan and Joakim Nivre, editors, *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT)*, pages 211–222, Prague, Czech Republic.

Pajas et Štěpánek, 2009. Pajas, Petr; Štěpánek, Jan. System for querying syntactically annotated corpora. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 33–36. ACL. http://www.aclweb.org/anthology/P/P09/P09-4009.pdf

Pajas et Štěpánek, 2008. Pajas, Petr; Štěpánek, Jan. Recent Advances in a Feature-Rich Framework for Treebank Annotation, in *The 22nd International Conference on Computational Linguistics - Proceedings of the Conference,* Manchester, pp. 673-680, 2008 http://www.aclweb.org/anthology/C/C08/C08-1085.pdf

Pajas et Štěpánek, 2006. Pajas, Petr; Štěpánek, Jan. XML-based representation of multi-layered annotation in the PDT 2.0. In *Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006)*, pages 40–47.

Pugh, 1990. Pugh, W. Skip lists: A probabilistic alternative to balanced trees. Communications of the ACM 33 6 (1990), pp. 668–676.

Randall, 2008. Randall, Beth. *CorpusSearch 2 Users Guide*, 2008. http://corpussearch.sourceforge.net/CS-manual/Contents.html.

Rohde, 2001. Rohde, D. Tgrep2 user manual. http://tedlab.mit.edu/~dr/Tgrep2/tgrep2.pdf, 2001.

Salmon-Alt et coll. 2004. Salmon-Alt, S. ; Romary, L. ; Pierrel, J.-M. Un modèle générique d'organisation de corpus en ligne : application à la FreeBank, *Traitement Automatique des Langues*, Vol.45, n°3, pp. 145-169, 2004. ISSN 1248-9433

Schmidt, 2009. Schmidt, Jens M. Interval Stabbing Problems in Small Integer Ranges. In *Algorithms and Computation, 20th International Symposium*, *Proceedings 2009*. Y. Dong, D.-Z. Du, and O. Ibarra (Eds.). ISAAC , Vol. 5878 Springer (2009), p. 163-172. ISBN 978-3-642-10630-9.

Söze-Duval Keyser 2008. *Pour une textométrie opérationnelle*, http://www.cavi.univ-paris3.fr/lexicometrica/jadt/ressources-textometriques/textes/RTI6provisoire.pdf

TEI Consortium, 2007. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium, eds. http://www.tei-c.org/Guidelines/P5/

W3C, 2009a. *XQuery Update Facility* 1.0. W3C W3C Candidate Recommendation 09 June 2009. http://www.w3.org/TR/2007/REC-xquery-20070123/

W3C, 2007a. *XML Path Language (XPath) 2.0.* W3C Recommendation 2007. http://www.w3.org/TR/2007/REC-xpath20-20070123/

W3C, 2007b. *XQuery 1.0: An XML Query Language.* W3C Recommendation 2007. http://www.w3.org/TR/2007/REC-xquery-20070123/

W3C, 2007c. *XSL Transformations (XSLT) Version 2.0.* W3C Recommendation 2007. http://www.w3.org/TR/2007/REC-xslt20-20070123/

Zhang et coll., 2001. Zhang, Chun; Naughton, Jeffrey; DeWitt, David; Luo, Qiong; Lohman, Guy. On supporting containment queries in relational database management systems. In *SIGMOD '01: Proc. ACM SIGMOD international Conference on Management of Data*, pages 425–436, New York. ACM.

Zloof, 1977. Zloof, M. M. Query-by-example: A data base language. IBM Systems Journal, 16(4):324–343.

8 Conclusion

L'objectif initial de cette recherche doctorale était de proposer un modèle informatique pour représenter, construire et exploiter des structures textuelles pour l'analyse de corpus, ce que nous appelons maintenant l'annotation structurelle.

En réalité, ce projet était vu comme une suite logique à plus de trente ans de recherche et développement dans le domaine de l'analyse de texte assisté par ordinateur. Il est donc apparu nécessaire d'assoir notre projet prospectif sur un bilan de ces années de recherche qui, s'appuyant sur une certaine posture théorique concernant l'analyse textuelle, cristallisent des points de vue méthodologiques servis par le modèle informatique implanté dans le logiciel SATO.

Inscrit dans la mouvance multidisciplinaire des sciences du langage et de l'analyse de discours, notre regard demeure celui de l'informaticien d'abord préoccupé par une problématique de modélisation informatique. Inscrit au cœur des sciences humaines en général et des sciences du langage en particulier, ce travail de modélisation, s'il doit suivre sa propre logique disciplinaire, est aussi tributaire du creuset d'influences dans lequel il évolue. Le rappel historique de ces influences, à travers un examen sommaire des paradigmes théoriques et des projets de recherche auxquels nous avons pris part, est éclairant à plus d'un égard. D'abord, parce qu'il permet de comprendre la dialectique entre un modèle de calcul et les objectifs méthodologiques qu'il entend favoriser. L'un ne va pas sans l'autre. L'outil informatique est l'instrument de nos objectifs et de nos stratégies de recherche sociale. D'un autre côté, la sémantique d'utilisation de l'outil de calcul se comprend sous l'éclairage de ces démarches de construction et d'interprétation des données textuelles soumises à l'outil.

Notre rappel historique permet aussi de constater que le caractère novateur du *modèle SATO* et de son implantation n'est pas tributaire d'un paradigme passager. Si sa pertinence s'est maintenue malgré les différences d'école et s'il a pu être utilisé dans des perspectives si variées, c'est qu'il repose sur une cohérence interne qui autorise son évolution au-delà de l'évolution accélérée des contraintes informatiques. C'est en effet la cohérence du modèle qui autorise une évolution en largeur et en profondeur.

En largeur d'abord, en ce que le modèle s'intègre harmonieusement, en amont, avec un modèle documentaire qui pose la question des corpus de recherche dans le creuset de la circulation des textes modélisés à des fins de recherche. Il s'agit de voir les corpus en termes d'objets *annotables* et de documents d'annotation de nature diverses, mais interreliés et composables. Au niveau de la syntaxe concrète, notre préjugé favorable aux recommandations de la TEI permet, à tout le moins, de bénéficier d'une référence permettant de mettre à l'épreuve nos hypothèses documentaires.

C'est aussi la cohérence de notre *modèle SATO* qui permet, en aval, d'intégrer des formalismes d'annotation plus puissants pour rendre compte de la multiplicité des relations qui organisent le tissus textuel. Cette recherche doctorale aura donc été l'occasion de montrer comment l'*annotation structurelle* peut s'intégrer au modèle lexique/occurrences actuel et compléter l'annotation *à plat*.

Notre travail de recherche a aussi permis de montrer que la normalisation XML permet de rendre compte du modèle élargi, de sa dimension documentaire à l'annotation structurelle en passant par les structures de données actuelles de SATO, données qui peuvent s'exprimer, de façon naturelle, en termes de documents externes d'annotation XML-TEI. On notera cependant que cet effort de formalisation XML n'impose pas d'office que l'on doive opter pour un traitement XML en format natif. Il existe plusieurs approches de traitement selon le niveau de granularité des données, l'ampleur des documents, le type d'interrogation et de mise à jour. Pour notre part, nous proposons de réserver le traitement des occurrences et du lexique à une couche logicielle spécialisée telle SATO. Nous proposons de recourir à un logiciel de dépôt de données pour la partie documentaire de la chaine de traitement. Enfin, pour l'annotation structurelle, nous nous proposons d'examiner de façon plus approfondie les solutions assez nombreuses que l'on retrouve actuellement en tenant compte des paramètres qui nous sont spécifiques, en particulier l'importance des fonctions de mise à jour et de parcours séquentiel du texte.

Finalement, l'intégration de toutes ces approches dans une interface unique nous invite à maintenir notre choix d'interfaces Web avec, probablement, des modules complémentaires permettant de doter la navigation d'outils particuliers, en particulier pour la manipulation des graphes d'annotation structurelle. C'est aussi dans ce contexte d'intégration que l'on évaluera

l'état du code informatique existant pour voir comment le faire évoluer afin d'en faciliter l'entretien et la modularité.

Index

Dans l'index alphabétique, les entrées commençant par une majuscule correspondent à des noms de personnes. Les entrées en lettres capitales correspondent à des noms de logiciels ou de protocoles. Les entrées en minuscules renvoient à des notions discutées dans le corps du texte.

ACTE.....26, 117, 128, 130, 132, 134, 135, 138, 239, 354
Adam..10, 16, 17, 32, 49, 50, 201, 233, 290, 291, 292, 293, 296, 299, 300, 301, 302, 314, 315, 322, 326, 327, 328, 329, 330, 343, 344, 346, 347, 382
ALCESTE.....43, 69, 97, 98, 100, 101, 102, 103, 104, 294
Allen.....87, 113
alphabet.....56, 57, 119, 122, 123, 245, 272, 273, 274, 275, 316, 355
ALTO.....320, 321, 322
ambiguïté.....40, 170, 186, 187
analyse discriminante.....141, 146, 147, 164
analyse factorielle des correspondances.....61, 81, 104, 105, 108, 109, 199, 290
approche hypothético-déductive.....33, 86, 90
approche inductive.....20, 23, 33, 86, 90, 97, 127, 212
arborescence.....129, 131, 348, 349, 356, 360, 361, 367, 377
arbre. 11, 131, 289, 300, 327, 349, 350, 355, 356, 357, 358, 359, 360, 361, 366, 367, 376, 377, 378, 379, 380, 381
Armony.....196, 237
ASTARTEX.....294
ATONET..97, 109, 115, 118, 128, 209, 210, 211, 215, 217, 221, 226, 278, 294, 296, 316, 346, 363, 382
axe lexical.....7, 55, 60, 62, 83, 153, 154
axe paradigmatique.....38, 39, 40, 41
axe syntagmatique.....39, 40, 153, 154
Ayoub.....117, 180, 217, 219, 225, 226, 227, 229, 233, 234
Bakhtine.....17, 50, 289, 346
balisage.....6, 8, 47, 48, 54, 88, 97, 102, 128, 182, 196, 197, 212, 214, 221, 223, 226, 233, 242, 243, 252, 290, 293, 296, 326, 344
Barbaud.....118
Barnard.....47, 50
Barthes.....15
Battista.....50
Bauman.....235
Beauchemin.....24, 25, 26, 124, 185, 190, 235, 238
Bécue.....211
Bégin.....181, 184, 234
Benoît.....21, 50
Bentley.....361, 382
Benzécri.....100, 111
Bertrand-Gastaldy.....17, 18, 19, 36, 45, 50, 117, 139, 140, 141, 142, 143, 144, 180, 234, 235
bibliothèque virtuelle.....119, 345
Bird.....356, 357, 358, 359, 376, 378, 379, 382, 383, 384
Bolasco.....211
Boncz.....382
Bonhomme.....47, 50
Borges.....293, 314, 315, 316, 331, 336, 343, 344, 354, 363, 364, 366, 368, 370
Bouchard.....112
Bourque.....117, 180, 185, 190, 235, 280, 281
Bradley.....249, 287
Braga.....378, 382

Brill.....184, 235
Brunet.....198, 235
Buneman.....355, 383
Burkowski.....48, 50, 354, 355, 376
Burnard.....50, 211, 235, 240
catégorisation.....7, 22, 23, 28, 33, 35, 36, 60, 61, 69, 70, 71, 77, 89, 90, 92, 93, 99, 100, 121, 130, 141, 142, 169, 170, 174, 181, 183, 184, 186, 187, 188, 189, 199, 201, 205, 206, 247, 254, 259, 264, 270, 271, 272, 289, 291, 351, 353
Catégorisation.....169, 170, 240
Charmaz.....20, 51
Charolles.....290, 346
Chazelle.....361, 383
Chomsky.....21
Clark.....357, 383
Clarke.....354, 355, 376
classe lexicale.....65, 67, 120, 121, 123, 242
client-serveur.....116, 191, 193, 200, 201, 247, 248, 250, 251, 252, 253, 275, 276
constructeur.....353, 354
cooccurrence.....22, 77, 216, 276, 277, 278, 279, 280, 281, 282, 283, 350, 357
Cooccurrence.....280, 282
Corbin.....58, 111
Cormack.....376
corrélation.....22, 165, 189
Coulon.....124, 125, 126, 137, 235
Courtois.....176, 235
Cucumel.....164, 235
Daoust.…6, 8, 28, 36, 38, 40, 42, 43, 48, 51, 86, 88, 97, 98, 112, 116, 117, 119, 120, 123, 124, 130, 131, 137, 138, 147, 148, 149, 157, 158, 168, 180, 184, 185, 194, 195, 199, 209, 210, 212, 213, 216, 218, 221, 234, 235, 236, 237, 238, 239, 240, 244, 276, 278, 281, 287, 289, 291, 292, 293, 294, 296, 346, 347, 375, 383
David.....50
Delannoy.....270, 287
Della Faille.....11, 43, 51, 195, 237
dépôt de données.....212, 213, 217, 218, 221, 223, 224, 226, 235, 278, 292, 293, 346, 383, 387
DeRose.....383
DIATAG.....294
dispositif de lecture.....7, 32, 34, 147, 199
dispositif expérimental.....9, 28, 33, 35, 36, 54, 72, 121, 145, 154
distance du Chi2.....83, 89, 99, 106, 107, 108
DISTANCE du Chi2.....107
Dobrowolski.....112, 235
Drouin.....210
DTM.....97, 98, 105, 107, 108, 109, 110, 294
Duchastel.....43, 51, 97, 112, 117, 118, 124, 126, 127, 172, 180, 183, 185, 190, 194, 195, 196, 210, 235, 237, 238, 276, 280, 281, 287, 346
Dufresne.....112, 235, 238
Dupuis.....112, 117, 118, 157, 183, 184, 185, 236, 240
Dupuy.....124, 130, 131, 138, 172, 190, 238, 239

Edelsbrunner....361, 383
EIDOS....184, 187, 191
emboitement....302, 327, 328, 330, 331, 352, 353, 355, 368
encapsulage....268, 285
entête tei....215, 216, 222, 223, 228, 282, 295, 298, 303, 326
ergonomie....9, 41, 68, 71, 107, 125, 126, 142, 148, 250, 307, 354, 377
espace lexical....79, 83, 85, 99
Évrard....23, 24
FEDORA....223, 224, 238
Fielding....277
filtre....27, 64, 65, 73, 74, 76, 77, 78, 79, 83, 129, 131, 137, 158, 179, 201, 203, 226, 252, 262, 263, 273, 291, 361, 362, 376, 382
Filtre....77
Fleury....113, 210, 240, 347, 383
forme fléchie....58, 155, 172, 174, 176, 177, 179, 180
forme graphique....57, 58, 107, 121, 122, 176, 177
forme lexicale....7, 37, 40, 55, 56, 57, 60, 61, 65, 66, 67, 69, 70, 74, 76, 80, 81, 82, 83, 84, 89, 90, 92, 99, 106, 107, 108, 109, 110, 120, 154, 155, 170, 174, 175, 185, 203, 204, 205, 242, 244, 246, 272, 273, 274, 283, 289, 294, 316, 349, 376
FORTRAN....116, 120, 267
Foucault....14, 17, 51
Fraser....44, 51
FREEBANK....233, 240, 385
gabarit....116, 250, 251, 252, 255, 267
GALLICA....320, 345
Gélinas-Chebat....86, 87, 112, 172, 211, 235, 236, 238
Ghodke....358, 359, 376, 383
Giroux....50
Glaser....20, 51
Gormack....354, 355
grammaires locales....157, 159, 170, 184, 188
granularité....16, 214, 218, 224, 349, 363, 387
graphe....6, 22, 65, 177, 182, 213, 291, 292, 302, 307, 310, 314, 327, 330, 331, 332, 336, 343, 350, 353, 354, 387
Graphe....337
grappe de serveurs....116, 191, 209, 277
Grau....44, 53
Greimas....15
Grenon....219, 225, 227, 229, 234
Gross....176
Guilhaumou....13, 51
Guillaume....219, 220, 224, 225, 226, 227, 229
Habert....35, 44, 51, 58, 112, 158, 210, 238, 290, 347
Haejoong....379, 383
Halliday....18, 51
Hamilton-Smith....121, 238
Hanson....361, 383
Harman....16, 52

Heiden.....184, 198, 211, 238, 240, 249, 287
héritage.....65, 66, 67, 76, 111, 120, 121, 123, 241, 242, 247, 291
Hjorland.....20, 21
Hochon.....23, 24, 47, 52
Ide.....47, 52
idéateur.....20, 44, 46
indice de Salton.....82
indice Gunning.....82
indice GUNNING.....173
inférence.....26, 28, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 141, 142, 148, 237
Ingwersen.....20
intertextualité.....19, 25, 45, 53, 129, 213, 214, 218, 279, 292
Jacobson.....211
JEUDEMO.....119, 121, 122, 123, 239
Johnson.....361, 383
journalisation.....121, 247, 270
Kayser.....124, 125, 126, 137, 235
Kepser.....356, 383
König.....356, 384
Labonté.....274, 287
Lai.....356, 357, 358, 384
Lamalle.....113, 240
Lanteigne.....50
Laperrière.....20, 52
Laroche.....132, 148, 149, 164, 168, 172, 190, 236, 237, 238
Lebart.....57, 61, 97, 104, 107, 112, 211, 294, 347
Lebel.....112, 238
Leblanc.....210
lecture électronique.....28, 29, 30, 31, 32, 33, 34, 35, 138, 147, 180, 199
Lee.....357, 378, 382
Lemieux.....117, 180
Lerdorf.....287
Leventhal.....87, 113
lexicalisation.....58, 59, 60, 111, 144, 315, 316, 348, 349
LEXICO.....43, 97, 98, 104, 105, 106, 107, 198, 294
lexicométrie.....10, 16, 34, 35, 43, 59, 61, 83, 97, 104, 198, 221, 290, 345
Lezius.....356, 384
Liberman.....357, 382
lisibilité.....24, 73, 82, 118, 150, 151, 166, 167, 172, 174, 236, 238, 302
LISP.....76, 119, 130, 131, 134, 239
locution.....60, 111, 144, 145, 154, 159, 170, 175
macrostructure.....16, 201, 327
Maier.....18, 52
Maingueneau.....13, 15, 16, 17, 20, 42, 52, 239
Maldidier.....51
Marchand.....36, 50
Marcoux.....98, 112, 210, 213, 221, 235, 244, 278, 287, 289, 294, 296, 346, 347, 383
Martinez.....113, 210, 240, 276, 287

McCarthy....239
McCreight....361, 384
McKenzie....18, 52
Meier....358, 384
métadonnée....22, 212, 213, 215, 216, 222, 223, 224, 225, 226, 228, 229, 278
Meunier....38, 116, 118, 119, 234, 235, 239, 240
Michard....47, 52
milestone....76, 226, 227, 228, 316, 364, 366, 367, 368, 377
Mírovský....381, 384
Misuraca....211
modèle relationnel....350, 355, 379
morphologie....16, 24, 26, 27, 59, 89, 99, 125, 126, 127, 129, 130, 169, 175, 176, 177, 363
n-gramme....59, 61
Nazarenko....51
orienté objet....268, 270, 284, 285
Ouellet....148, 149, 168, 190, 236, 237
Ouellette....119, 121, 239
Pagola....144, 234, 235
Pajas....356, 379, 380, 384
Paquette....239
Paquin....24, 25, 26, 27, 28, 124, 130, 131, 132, 138, 234, 235, 238, 239
Parker-Pope....87, 113
PASCAL....116, 123, 131, 134, 183, 254, 267, 271, 275, 286
Passmore....287
patron de concordance....77, 78
Patton....20, 52
Pêcheux....15, 16, 52, 239
PERL....48, 70, 74, 98, 200, 201, 250, 253, 265, 275, 380, 381
PHP....200, 250, 258, 287
Pierrel....240, 375, 385
Pincemin....211
Plante....200, 239
Poirier....141, 239
polymorphisme....270
Portelance....118
Prévost....240
procédural....54, 130, 268
propriété. 7, 48, 56, 57, 61, 62, 63, 64, 65, 66, 67, 70, 71, 73, 74, 75, 76, 77, 78, 79, 80, 81, 83, 84, 88, 89, 92, 93, 96, 98, 99, 100, 109, 111, 114, 130, 140, 142, 143, 144, 146, 154, 157, 158, 169, 170, 172, 175, 177, 178, 179, 180, 201, 203, 205, 206, 207, 227, 229, 241, 242, 244, 245, 247, 265, 270, 271, 282, 289, 291, 315, 317, 349, 351, 352, 353, 360, 362, 365, 376, 377
Propriété....179
Proulx....181, 184, 187, 191, 234
Przepiórkowski....345, 347
Pugh....361, 384
PYTHON....200, 378, 382
Randall....356, 384
Rastier....17, 32, 45, 52

rdf....226, 240
RDF....215, 216, 222, 223, 226, 228, 229, 240, 279, 375
Reinert....69, 97, 100, 113, 211, 294, 347
reproductibilité....20, 34, 35, 72
Rizkallah....346, 383
Robin....51
Rockwell....199, 240, 249, 287, 288
Rohde....356, 385
Rolland....116, 118, 119, 120, 239, 240
Romary....240, 375, 385
Russell....121, 240
Sabah....44, 53
SACAO....115, 117, 124, 126, 127, 128, 172, 190, 192, 238
Saint-Denys Garneau....36, 50
Salem....51, 57, 61, 97, 104, 112, 113, 198, 210, 240, 294, 347, 363
Salmon-Alt....240, 385
Salton....82, 113
Schmidt....361, 385
segments dynamiques....11, 48, 181, 183, 345, 348, 351, 376
Silberztein....157, 184, 188, 190, 240
Söze-Duval Keyser....287, 363, 385
Sperberg....50, 52, 240
Strauss....20, 51
structures dynamiques....348, 349
textométrie....10, 11, 69, 227, 238, 241, 287, 346, 347, 349, 357, 362, 376, 383, 385
Thlivitis....45, 53
Todorov....15
Trybula....21, 22, 23
UNICODE....56, 59, 201, 209, 242, 246, 271, 272, 273, 274, 275
Van Dijk....13, 15, 53
Viprey....14, 38, 48, 53, 210, 211, 289, 293, 294, 346, 347
Weaver....18, 53
WEBLEX....198, 238, 249, 287
Weinrich....290, 347
Wirth....267
Witte....87, 113
XPATH....299, 356, 357, 366, 378
XQBE....378
Xquery....360, 365, 381
XQuery....357, 366, 378
XQUERY....356
XSLT....252, 267, 281, 307, 310, 331, 343, 356, 378, 379, 380
Zhang....359, 385
Zimina-Poirot....210
Zloof....378, 385
....51
Štěpánek....356, 379, 380, 384

Table des matières

Sommaire....2
Remerciements....3
1 Présentation....5
Résumé (1a, remarque)....5
2 Quels modèles de calcul pour quelles analyses textuelles?....12
2.1 L'analyse de discours....13
2.2 La perspective sémiotique....17
2.3 L'approche documentaire....19
2.4 La lecture experte....23
2.5 L'analyse de texte par ordinateur....28
L'analyse de texte assistée par ordinateur : lunettes de lecture des textes électroniques (2.5a, publication)....29
L'informaticien, le lecteur et le texte, l'approche SATO (2.5b, publication)....36
À propos de la variation linguistique (2.5c, remarque)....42
2.6 Quels modèles de calcul?....43
3 SATO : un modèle informatique pour la construction de dispositifs expérimentaux....54
3.1 Introduction....54
3.2 Le modèle lexique / occurrences....55
Texte dans le plan lexique/occurrence (3.2a, exemple)....55
Un même lexique... deux textes (3.2b, exemple)....56
Génération d'un corpus en SATO (3.2c, définition)....56
Texte avec figement de majuscule (3.2d, exemple)....57
Texte avec figement de locution (3.2e, exemple)....60
À propos de la consolidation terminologique (3.2f, remarque)....61
Texte augmenté de propriétés (3.2g, exemple)....62
Référence de pagination (3.2h, exemple)....63
Référence de pagination (3.2i, exemple)....63
Référence de pagination (3.2j, exemple)....64
Référence de pagination (3.2k, exemple)....64
Notion de propriété en SATO (3.2l, définition) 64
Utilisation de propriété lexicale dans le texte (3.2m, exemple)....67
3.3 L'ergonomie de SATO....68
Le bureau de SATO 4.3 (3.3a, figure)....68
L'onglet Fichier de l'interface bureau de SATO 4.3 (3.3b, notice technique)....69
L'interface d'analyse de SATO 4.3 (3.3c, figure)....70
Items du menu de catégorisation de SATO 4.3 (3.3d, définition)....70

3.4 Les opérations dans le plan lexique/occurrences....73
3.4.1 Opérations communes....73
Opérations sur le plan lexique/occurrence (3.4.1a, exemple)....73
Filtrage sur les caractères (3.4.1b, exemple)....74
Filtrage sur les valeurs de propriété (3.4.1c, exemple)....74
Opération d'exportation en SATO (3.4.1d, définition)....76
3.4.2 Opérations sur l'axe des occurrences....76
Bornes de contexte dans les concordances en SATO (3.4.2a, définition)....77
Filtre contextuel dans les concordances en SATO (3.4.2b, définition)....77
Sous-textes en SATO (3.4.2c, définition)....79
Analyseur COMPTAGE de SATO (3.4.2d, définition)....81
3.4.3 Opérations sur l'axe lexical....83
Analyseur DISTANCE de SATO (3.4.3a, définition)....83
3.4.4 Opérations combinant les deux axes....85
3.5 Un exemple d'analyse illustrant la construction d'une grille catégorielle....86
3.5.1 Introduction....86
3.5.2 Analyse exploratoire d'entrevues de groupe : les jeunes Français et le tabac....87
Codification SATO du corpus d'entrevues (3.5.2a, exemple)....88
Analyse de distance sur les formes lexicales brutes avant/après l'introduction de la brochure (3.5.2b, tableau I)....90
Analyse de distance sur les valeurs de la propriété thème avant/après l'introduction de la brochure (3.5.2c, tableau II)....93
Analyse de distance avant/après pour les fumeurs vs. les non-fumeurs (3.5.2d, tableau III)....95
Analyseur PARTICIPATION (thème=apparence) (3.5.2e, tableau IV)....96
Analyseur PARTICIPATION (thème=mort) (3.5.2f, tableau V)....96
3.5.3 Analyse exploratoire d'entrevues de groupe : quand ALCESTE, DTM, LEXICO et SATO se donnent la main....97
Classes produites par ALCESTE sur le corpus Initial (3.5.3a, tableau)....102
AFC produite par Lexico sur le corpus Participant (3.5.3b, figure)....105
Comparaison entre les spécificités et la distance du Chi2 (3.5.3c, tableau)....106
AFC produite par DTM sur le corpus Participant (3.5.3d, figure)....108
AFC produite par DTM sur le corpus Participant catégorisé (3.5.3e, figure)....109

AFC produite par DTM sur le corpus Participant réduit (3.5.3f, figure).....110

4 Trente ans de développement et d'utilisation du logiciel SATO 114
4.1 Introduction : de l'ordinateur central au PC, et du PC au traitement distribué.....114
Repères historiques (4.1a, remarque) 116

4.2 SATO, versions 1 et 2.....118
Jeudemo, logiciel contemporain à SATO 1. (4.2a, remarque).....121

4.3 Projet SACAO : ressources partagées sur ordinateur central.....124
4.4 Projet ACTE : influence du courant cognitiviste.....128
4.5 Le projet SOQUIJ.....139
Dépistage des locutions et des termes complexes (4.5a, notice technique).....145

4.6 Le projet SATO-CALIBRAGE.....147
SATO-CALIBRAGE : présentation d'un outil d'assistance au choix et à la rédaction de textes pour l'enseignement (4.6a, publication).....149

Procédure de génération de la base de données lexicales (4.6b, notice technique) . 174

4.7 Projet AlexATO : développement d'un modèle de traitement coopératif.....180
Les limites de l'architecture VOLVOX (4.7a, notice technique).....182

Un protocole pour la mise au au point d'algorithmes de désambiguïsation catégorielle (4.7b, publication).....185

4.8 Projet Visibilité : le pari d'une architecture Web.....191
4.9 Projet ATO-MCD : une infrastructure robuste pour l'ATO.....194
SATO-XML : une plateforme Internet ouverte pour l'analyse de texte assistée par ordinateur (4.9a, publication).....195

4.10 Projet ATONET.....210
Liste des membres d'ATONET en 2010 (4.10a, remarque).....210

Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche (4.10b, publication).....218

4.11 Bibliographie du chapitre 4.....233
5 Le modèle d'implantation de SATO.....241
5.1 Choix stratégiques 241
5.2 Le modèle de données de SATO.....243
Fichiers internes de SATO 4.3 (5.2a, définition).....245

Représentation interne du corpus dans SATO 4.3 (5.2b, définition).....246

5.3 Une architecture client-serveur.....247
5.4 Modèle de l'interface 253
Guide de programmation des interfaces HTML (5.3b, notice technique).....254

5.5 Un code informatique en évolution.....267
5.5.1 Description du programme.....267

5.5.2 Le passage à l'Unicode. 271

5.5.3 Des services Web au format XML-TEI....275
Un service Web pour l'analyse de la cooccurrence (5.5a, publication)....276
5.5.4 Vers un SATO en logiciel libre?....284
6 L'annotation structurelle....289
6.1 problématique....289
L'annotation structurelle (6.1a, publication, extrait)....289
6.2 Linguistique textuelle et TEI....292
Document d'annotation (6.2a, définition)....293
6.2.1. Perspective fonctionnelle de la phrase : la relation thème-rhème....295
6.2.1.1 Présentation de l'exemple....295
Marquage des occurrences par la balise w (document doc1.xml) (6.2.1a, exemple) 297
6.2.1.2 Identification des segments textuels....298
6.2.1.3 Annotation structurelle des relations thèmes-rhèmes et de la progression thématique....300
Marquage des occurrences par la balise w (document ThemeRheme.xml) (6.2.1b, exemple)....303
6.2.1.4 Annotation structurelle. : utilisation des structures de graphe....307
ThemeRheme-graph.xml (6.2.1c, exemple)....307
Feuille de style transformant en graphe l'analyse thème-rhème (fichier ThemeRheme-graph.xsl). (6.2.1d, exemple)....310
6.2.2. Analyse textuelle d'un récit de Jorge Luis Borges....314
Un récit de Borges : LE CAPTIF en format texte (6.2.2a, exemple)....315
Un récit de Borges : LE CAPTIF en format SATO (fichier borges_adam.sat). (6.2.2.b, exemple)....315
Un récit de Borges : LE CAPTIF en format TEI (fichier borges_adam.xml). (6.2.2c, exemple)....316
Format ALTO. (6.2.2d, exemple)....321
Document d'annotation structurelle (StructureCompositionnelle.xml). (6.2.2e, exemple)....322
6.2.2.1 Analyse de la structure compositionnelle du texte....327
Feuille de style transformant en graphe l'analyse du plan de texte du récit de Borges (fichier StructureCompositionnelle.xml) (6.2.2f, exemple)....331

Graphe d'annotation structurelle (StructureCompositionnelle-graph.xml). (6.2.2g, exemple)....337
6.2.2.2 Énonciation narrative et source du savoir....343
6.2.2.3 Référent évolutif et identité narrative....343
6.2.2.4 Une fable sur le temps, la mémoire et l'oubli....344
6.3 Quelques mots de conclusion....345
7 Modèles informatiques pour l'exploitation de la structure formelle et des segments dynamiques....348
7.1 Introduction....348
7.2 Le modèle algébrique....351
Nos hypothèses de 1993 sur les segments dynamiques (7.2a, remarque)....351
7.3 Le modèle arborescent....356
À propos des index d'intervalles (7.3a, remarque)....360
7.4 Un modèle de données en couches multiples....362
Version numérisée tramée du texte en format TEI (fichier borges_trame.xml). (7.4a, exemple)....363
Annotation structurelle (avec balises frontières référentielles et analytiques) sur la trame de Borges (fichier borges_struct_v1.xml). (7.4b, exemple)....364
Annotation structurelle (avec balises frontières référentielles) sur la trame de Borges (fichier borges_struct_v2.xml). (7.4c, exemple)....366
Annotation structurelle (sans balises frontières) sur la trame de Borges (fichier borges_struct_v3.xml). (7.3d, exemple)....368
Annotation cumulative sur la trame de Borges (fichier borges_struct_v4.xml). (7.4e, exemple)....370
7.5 La question des interfaces usagers....377
8 Conclusion....386
Index....389

Titre de la thèse / *Title*

Modélisation informatique de structures dynamiques de segments textuels pour l'analyse de corpus. / *Data-processing modeling of dynamic structures of textual segments for the analysis of corpus.*

Résumé / *Abstract*

L'objectif de la thèse est de proposer un modèle informatique pour représenter, construire et exploiter des structures textuelles. Le modèle proposé s'appuie sur une représentation du texte sous la forme d'un plan lexique/occurrences augmenté de systèmes d'annotations lexicales et contextuelles, modèle dont une implantation a été réalisée dans le logiciel SATO dont on présente les fonctionnalités et l'organisation interne. La présentation d'un certain nombre de travaux rendent compte du développement et de l'utilisation du logiciel dans divers contextes.

La prise en charge formelle des structures textuelles et discursives trouve un allié dans le langage de balisage XML et dans les propositions de la Text Encoding Initiative (TEI). Formellement, les structures construites sur les segments textuels correspondent à des graphes. Dans le contexte d'une analyse textuelle en élaboration, ces graphes sont multiples et partiellement déployés. La résolution de ces graphes, au sens du rattachement des nœuds à des segments textuels ou à des nœuds d'autres graphes, est un processus dynamique qui peut être soutenu par divers mécanismes informatiques. Des exemples tirés de la linguistique textuelle servent à illustrer les principes de l'annotation structurelle. Des considérations prospectives sur une implantation informatique d'un système de gestion de l'annotation structurelle sont aussi exposées.

The objective of the thesis is to propose a data-processing model to represent, build and exploit textual structures. The suggested model relies on a «type/token» form of text representation extended by systems of lexical and contextual annotations. This model's establishment was carried out in the SATO software -- of which the functionalities and the internal organization are presented. Reference to a number of works give an account of the development and use of the software in various contexts.

The formal assumption of the textual and discursive structures find an ally in the beaconing XML language and the proposals of the Text Encoding Initiative (TEI). Formally, the structures built on the textual segments correspond to graphs. In a development driven textual analysis context, these graphs are multiple and partially deployed. Their resolution, within the fastening of the nodes to textual segments or that of other graphs, is a dynamic process which can be sustained by various data-processing mechanisms. Examples drawn from textual linguistics are used to illustrate the principles of structural annotation. Prospective considerations for the data-processing establishment of a management system of the structural annotation are also exposed.

Mots clés / *Key Words*

Analyse de texte assistée par ordinateur ; analyse de discours ; modèle SATO ; annotation structurelle ; TEI ; textométrie. / *Computer Aided Text Analysis ; discourse analysis ; SATO model ; structural annotation ; TEI ; textometry.*