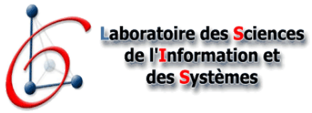


Université d'Aix-Marseille

Ecole Doctorale 184

Faculté des Sciences et Technique

LSIS UMR CNRS 7296 / Dimag
CLéO, OpenEdition

Thèse présentée pour obtenir le grade universitaire de docteur

Spécialité : Informatique

Chahinez BENKOUSSAS

Approches non supervisées pour la recommandation de lectures et la mise en relation automatique de contenus au sein d'une bibliothèque numérique

Soutenue le 14/12/2016 devant le jury :

Pr. Mohand BOUGHANEM	Université Toulouse III	Rapporteur
Pr. Sylvie CALABRETTO	LIRIS-INSA Lyon	Rapporteur
Pr. Antoine DOUCET	Université La Rochelle	Examinateur
Pr. Frédéric BECHET	Université Aix-Marseille	Examinateur
Pr. Milagros FERNANDEZ-GAVILANES	Univesité Vigo	Examinateur
Pr. Patrice BELLOT	Université Aix-Marseille	Directeur de thèse

Remerciement

Je remercie très chaleureusement Mr Mohand BOUGHANEM et Mme Sylvie CALABRETTO pour l'honneur qu'ils m'ont accordé en acceptant d'être rapporteurs de cette thèse. Je remercie également Mr Antoine DOUCET, Mr Frédéric BECHET et Mme Milagros FERNANDEZ-GAVILANES d'avoir accepté d'être membres du jury qui a examiné ce travail.

Je souhaite remercier mon encadrant de thèse Mr Patrice BELLOT, pour m'avoir fait confiance et laissé des libertés d'initiative tout au long de la thèse et surtout pour ses qualités humaines. J'aimerais aussi exprimer en particulier ma gratitude pour la confiance qu'il m'a accordé en me proposant ce sujet de thèse, et aussi pour son attention et son écoute pendant les moments difficiles.

Un remerciement particulier à mes collègues de bureau Radhia, Sherine et Hussam qui m'ont aidé durant ma thèse, sans oublier bien sûr mes collègues qui ont travaillé sur le projet INTER-TEXTES, Elodie et Anais, aussi au Cléo (Centre pour l'édition éléctronique ouverte) et à ses équipes, notamment Arnaud et Mathieu, pour le soutien technique durant ce travail de thèse.

Aussi, je remercie l'ensemble des doctorants, chercheurs, enseignants, personnels que j'ai côtoyé au sein du laboratoire LSIS.

Une pensée chaleureuse à mes parents, mon frère et mes sœurs et à ma belle-famille. Une pensée affectueuse pour mon mari Amine, qui m'a soutenu et surtout supporté durant la rédaction de ce manuscrit.

À mon PAPA, ma MAMAN et mon cher époux.

Résumé

Cette thèse s'inscrit dans le domaine de la recherche d'information (RI) et la recommandation de lecture. Elle a pour objets :

— La création de nouvelles approches de recherche de documents utilisant des techniques de combinaison de résultats, d'agrégation de données sociales et de reformulation de requêtes ;

— La création d'une approche de recommandation utilisant des méthodes de RI et les graphes entre les documents

Deux collections de documents ont été utilisées. Une collection qui provient de l'évaluation CLEF (tâche Social Book Search - SBS) et la deuxième issue du domaine des sciences humaines et sociales (OpenEdition, principalement Revues.org). La modélisation des documents de chaque collection repose sur deux types de relations :

— Dans la première collection (CLEF SBS), les documents sont reliés avec des similarités calculées par Amazon qui se basent sur plusieurs facteurs (achats des utilisateurs, commentaires, votes, produits achetés ensemble, etc.) ;

— Dans la deuxième collection (OpenEdition), les documents sont reliés avec des relations de citations (à partir des références bibliographiques).

Nous montrons que les approches proposées apportent dans la plupart des cas un gain dans les performances de recherche et de recommandation. Le manuscrit est structuré en deux parties. La première partie «état de l'art» regroupe une introduction générale, un état de l'art sur la RI et sur les systèmes de recommandation. La deuxième partie «contributions» regroupe un chapitre sur la détection de comptes rendus de lecture au sein de la collection OpenEdition (Revues.org), un chapitre sur les méthodes de RI utilisées sur des requêtes complexes et un dernier chapitre qui traite l'approche de recommandation proposée qui se base sur les graphes.

Mots-clés : recherche d'information, recommandation, modèles de recherche d'information, graphes, bibliothèque numérique, réseau de citations, classification automatique.

Abstract

This thesis deals with the field of information retrieval and the recommendation of reading. It has for objects:

— The creation of new approach of document retrieval and recommendation using techniques of combination of results, aggregation of social data and reformulation of queries;

— The creation of an approach of recommendation using methods of information retrieval and graph theories.

Two collections of documents were used. First one is a collection which is provided by CLEF (Social Book Search - SBS) and the second from the platforms of electronic sources in Humanities and Social Sciences OpenEdition.org (Revues.org). The modelling of the documents of every collection is based on two types of relations:

— For the first collection (SBS), documents are connected with similarity calculated by Amazon which is based on several factors (purchases of the users, the comments, the votes, products bought together, etc.);

— For the second collection (OpenEdition), documents are connected with relations of citations, extracted from bibliographical references.

We show that the proposed approaches bring in most of the cases gain in the performances of research and recommendation. The manuscript is structured in two parts. The first part "state of the art" includes a general introduction, a state of the art of information retrieval and recommender systems. The second part "contributions" includes a chapter on the detection of reviews of books in Revues.org; a chapter on the methods of IR used on complex queries written in natural language and last chapter which handles the proposed approach of recommendation which is based on graph.

Keywords: information retrieval, recommendation, information retrieval models, graphs, digital library, citation's network, automatic classification.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Cadre général

« Nul ne peut atteindre l'aube sans passer par le chemin de la nuit. »

Khalil Gibran

Sommaire

Une large variété de ressources est mise à la disposition des utilisateurs sur Internet. Ces ressources ont la particularité d'être hétérogènes, distribuées et possédant un volume en croissance continue. Ces ressources, appelées aussi "items", peuvent être tout type de document électronique regroupant des données accessibles sous différents formats (textuel ou multimédia). D'après le site http://www.internetlivestats.com/, l'évolution des sites Web dans le réseau Internet est ultra rapide. Les statistiques réalisées pour ce sujet entre l'année 2000 et 2015 sont illustrées dans la figure 1.1. Il est clair, que les utilisateurs ont tendance à avoir recourt à Internet pour rechercher des réponses à leurs problématiques quotidiennes. Or, devant cette surabondance de ressources, l'utilisateur devient incapable de gérer cette masse d'information et de repérer les informations qui correspondent au mieux à ses attentes (information pertinentes).

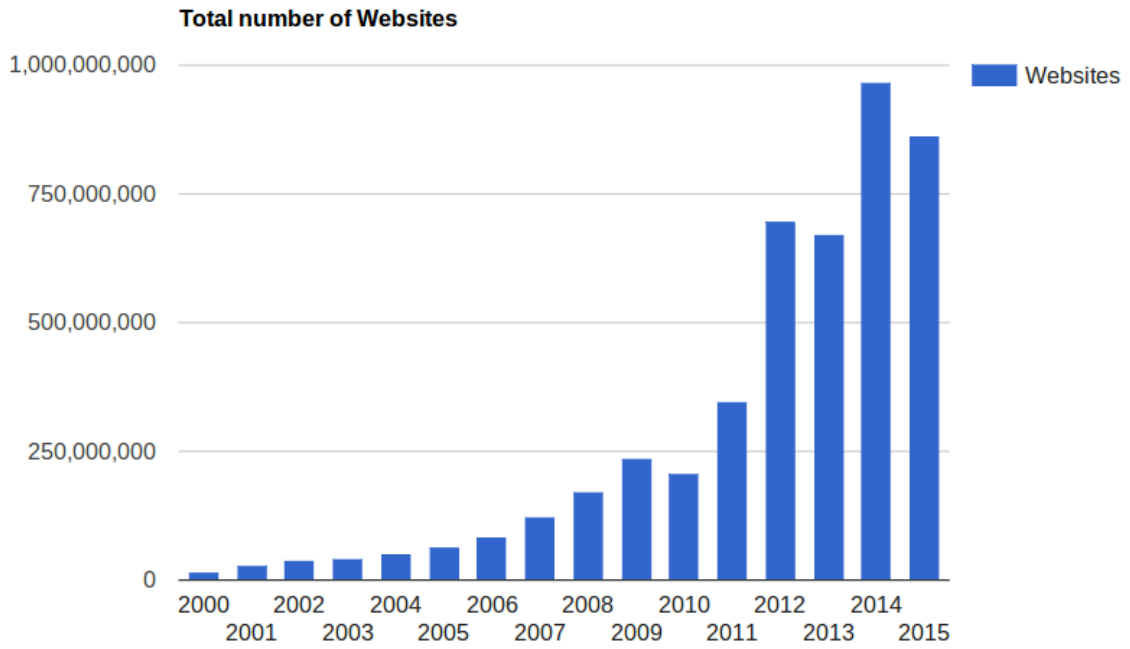


FIGURE 1.1 – Nombre de sites web entre l'an 2000 et 2015

Dans ce contexte, le besoin des outils qui facilitent l'accès à l'information s'avère crucial. Les moteurs de recherche et le domaine de la Recherche d'Information (RI) ont fait leur apparition pour pallier le problème d'accès aux informations présentes. Ces moteurs ont pour rôle de parcourir le Web afin d'indexer les ressources disponibles pour ensuite restituer celles qui correspondent à une requête donnée par l'utilisateur. Le moteur de recherche le plus populaire est *Google*, il a marqué dans la dernière décennie une évolution considérable en intégrant de nouvelles techniques avancées de requêtage (syntaxe et sémantique) et de réordonnancement des résultats.

En outre, les techniques de RI traditionnelles exploitent principalement le contenu des documents Web sans se soucier de la nature des documents ou des informations sociales qui entourent les documents. Peu importe les avis des utilisateurs sur un document donné, deux items proposés pour une requête donnée ayant un contenu qui satisfait cette requête auront la même importance bien que l'un soit mieux apprécié par les autres utilisateurs que l'autre. De plus, les requêtes attendues par les moteurs de recherche doivent être composées de mots clés. Certains utilisateurs préfèrent poser des requêtes bien détaillées en langage naturel. Avec les moteurs de recherche classiques cela n'est pas possible, on risque d'avoir des résultats qui s'éloignent du domaine ou sujet de recherche. Ces utilisateurs se rendent

généralement dans des forums pour demander des suggestions ou des recommandations. De cette manière, ils ne peuvent pas obtenir des réponses instantanées comme c'est le cas avec les moteurs de recherche. Il arrive qu'on souhaite chercher un type précis de document pour une requête donnée, comme par exemple, chercher tous les comptes rendus de lecture ou critiques d'un livre. Cette façon de rechercher n'est pas considérée dans les moteurs de recherche qui ne tiennent pas en considération le type des documents. Cela pour la seule cause que les documents ne sont pas typés, il n'y a pas un module de classification (en type/genre) derrière les moteurs de recherche. D'un autre coté, la RI ne tient pas en compte de l'utilisateur qui pose la requête, autrement dit deux utilisateurs qui posent la même requête auront la même réponse, or une requête posée par deux utilisateur peut avoir un sens différent pour chacun. En conséquence, les systèmes de recommandation sont apparus pour essayer de combler les failles de la RI.

1.1 Le projet INTER-TEXTES

Ma thèse s'inscrit dans le cadre du projet Inter-Textes (2013-2016) et du Programme d'Investissement d'Avenir (PIA) au sein du Centre pour l'édition électronique ouverte, Cléo pour le portail OpenEdition. Ce projet est l'un des axes du programme de R&D du Cléo en fouille de texte, il vise à valoriser et mettre en relation des ensembles de documents de nature scientifique par le biais de citations, d'allusion, de reprises, de références ou de liens hypertextes. Il s'agit de concevoir et développer des fonctionnalités d'agrégation de contenus scientifiques, de restructuration des documents, de validation des sources, de cross-linking des contenus et un système de recommandation.

Des analyses sur la mise en page des contenus, des contenus textuels (analyse sémantique) de leurs liens (références croisées entre les documents) et sur les usages (analyse des logs) compléteront dans le futur le processus de recherche.

Ce projet financé par le programme « Donds de la Société Numérique » géré par la Caisse des dépôts et consignations, fait intervenir quatre partenaires :

— **QWAM** [1] : PME spécialisée dans des solutions d'accès à l'information électronique, de la gestion de l'information scientifique et technique, de la gestion des flux de presse générale et économique et des applications d'agrégation, d'indexation et de recherche fédérées.

— **Demain Un Autre Jour** [2] : PME développant et commercialisant une plateforme pour la transformation de documents en XML, en particulier pour les secteurs de la presse, puis pour l'enrichissement de ces contenus.

— **Cléo** [3] : structure qui développe le portail OpenEdition, un ensemble de plateformes de ressources électroniques en sciences humaines et sociales : OpenEdition Books (les collections de livres), Revues.org (les revues), Hypothèses (les carnets de recherche), Calenda (les annonces d'événements).

— **LSIS** [4] : UMR fédérant près de 200 chercheurs, enseignants-chercheurs et doctorants autour de plusieurs domaines de l'informatique,de l'automatique et de l'image. Son équipe-projet DIMAG, en particulier, travaille sur la fouille, l'intégration et la fusion des données et ressources disponibles sur le Web.

1. www.qwamci.com/qwam-content-intelligence/
2. http://www.demainunautrejour.com/
3. http://cleo.openedition.org/
4. http://www.lsis.org/

1.2 Approches

Pour atteindre les objectifs fixés dans le projet Inter-Textes, nous avons divisé le travail en trois parties, chaque partie traite un domaine différent mais reste complémentaire. En premier, nous avons réalisé une étude sur la base de documents des plateformes Revues.org et Hypothèses afin d'extraire des caractéristiques qui permettent d'aider à faire une classification automatique en genre et plus particulièrement de détecter automatiquement les textes porteurs d'opinion (comptes rendus de lecture) dans le but de les exploiter pour le processus de recommandation.

La seconde partie concerne le domaine de la RI. l'objectif est d'utiliser les techniques de RI classique pour recommander automatiquement des livres d'Amazon à des utilisateurs qui ont posé des requêtes longues et complexes en langue naturelle. Ces requêtes sont composées de plusieurs éléments comme par exemple, un titre et un descriptif du besoin. En effet, nous avons fixé trois hypothèses, une qui concerne les modèles de recherche utilisés, la deuxième concerne la requête de l'utilisateur et la troisième repose sur le processus d'ordonnancement des résultats.

Dans la dernière partie, nous souhaitons explorer une nouvelle méthode de modélisation de document afin d'améliorer les performances du système de recommandation dans la partie précédente. L'enjeu est de trouver des liaisons entre les documents et ainsi de construire un graphe qui les regroupe pour l'exploiter par la suite dans le processus de recommandation. A ce niveau, notre hypothèse est que l'utilisation de ce type de structure permet de restituer plus de documents pertinents en comparaison des méthodes classiques. Dans cette partie, nous étudions deux collections de données différentes.

1.3 Contributions

Les contributions de cette thèse comprennent :

— Un détecteur automatique de comptes rendus de lecture BENKOUSSAS et FAATH [2014].

— Un modèle de recommandation pour des requêtes complexes qui intègre les informations sociales dans le processus de réordonnancement BENKOUSSAS et BELLOT [2013, 2014].

— Un modèle de recommandation qui exploite une structure de graphe basée sur des relations sociales BENKOUSSAS et BELLOT [2015a,b,c]; BENKOUSSAS et OLLAGNIER [2015].

— Un modèle de recommandation basé sur des méthodes de RI. Ce modèle combine les résultats de deux modèles différents de RI BENKOUSSAS et BELLOT [2015a,c].

1.4 Organisation du manuscrit

Ce manuscrit est organisé en trois parties. Dans la première partie, nous présentons le contexte général en décrivant l'origine du domaine de la RI et les systèmes de recommandation. Nous décrivons également le projet Inter-Textes dans lequel s'inscrit cette thèse, suivi d'une brève description des approches et des contributions. Cette partie contient deux chapitres "état de l'art ".

— Le chapitre *Recherche d'Information*, où nous introduisons un état de l'art du domaine de la Recherche d'Information (RI). Ce dernier se développe depuis plusieurs décennies et évolue en fonction des informations qui nous entourent (le type d'information

disponible et les requêtes des utilisateurs). La RI se construit sur une culture de la validation d'hypothèses par l'expérimentation où sont utilisées les notions de pertinence et mesures d'évaluation pour la compréhension du comportement d'un système de RI. Ce chapitre définit ces notions avec une brève description de quelques compagnes d'évaluation également.

— Le chapitre *Système de Recommandation*, où nous introduisons un état de l'art des systèmes de recommandation. Ces derniers sont de plus en plus développés dans plusieurs domaines d'application pour fournir des recommandations adaptées aux goûts, aux besoins ou aux moyens des utilisateurs afin de les aider à accéder à des ressources utiles ou intéressantes au sein d'un espace de données important. Ce chapitre définit les différentes techniques existantes pour la recommandation et particulièrement les techniques basées sur les graphes.

La deuxième partie est consacrée à la description de nos contributions. Elle contient les chapitres suivants :

— Le chapitre *Détection automatique des comptes rendus de lecture*, où nous nous intéressons à la détection de textes critiques porteurs d'opinions. Plus précisément, nous cherchons à recueillir automatiquement un ensemble de comptes rendus de lecture de livres au sein de plateformes Web afin de les exploiter ultérieurement dans le cadre d'une recherche "sociale"de livres. Nous montrons qu'une telle classification en genre peut bénéficier d'approches statistiques simples, au delà des sacs de mots.

— Le chapitre *Recommandation pour des requêtes complexes*, où nous présentons la méthodologie générale que nous avons adopté pour mener la tâche de recommandation de livres pour des requêtes d'utilisateurs complexes écrites en langage naturel. Nous présentons d'une part les données utilisées dans nos expérimentations et d'autre part les architectures de chacune des méthodes proposées et les résultats obtenus.

— Le chapitre *Recommandation sur des données structurelles (graphes)*, où nous présentons une nouvelle méthode de recommandation qui combine des approches de recherche d'information et des algorithmes de parcours de graphe. Nous décrivons la modélisation des données en structure de graphe ensuite nous introduisons l'architecture globale de notre méthode. Nous avons testé cette dernière sur deux collections de données de nature différente ; la première est une collection standard de la campagne d'évaluation INEX (au sein de la conférence CLEF la tâche Social Book Search) et la deuxième est une collection de documents scientifiques issue du portail OpenEdition.org plus précisément de la plateforme Revues.org.

La dernière partie de la thèse comprend la conclusion et les perspectives de recherche. Elle résume les principales contributions de la thèse et présente quelques orientations futures de nos travaux de recherche dans le cadre des systèmes de recommandation.

Chapitre 2

Recherche d'information

Résumé : Dans ce chapitre, nous introduisons un état de l'art du domaine de la Recherche d'Information (RI). Ce dernier se développe depuis plusieurs décennies et évolue en fonction des informations qui nous entourent (le type d'information disponible et les besoins des utilisateurs). La RI se construit sur une culture de la validation d'hypothèses par l'expérimentation où sont utilisées les notions de pertinence et des mesures d'évaluation standardisées. Ce chapitre définit ces notions avec une brève description de quelques campagnes d'évaluation.

Sommaire

2.1 Introduction

De plus en plus d'informations se génèrent chaque jour sur le Web soit plus de 29 000 Gigaoctets par seconde et plus de 915 000 000 000 de Gigaoctets de données publiés chaque année d'après le site *http://www.planetoscope.com/*. Des données sont envoyées et récoltées chaque seconde par près de 3 milliards d'internautes, soit 42% de la population mondiale connectée, un chiffre en constante augmentation. Plus de 2 milliards d'êtres humains utilisent le Web, soit 30% de la population mondiale [1]. Dans ce contexte où l'accès à internet est quasi-permanent, accéder rapidement et surtout efficacement à l'information est un défi majeur. Un des exemples concrets du besoin en recherche d'information est le nombre énorme de requêtes soumises au moteur de recherche Google au fil de ces dernières années comme illustré dans la figure 2.1.

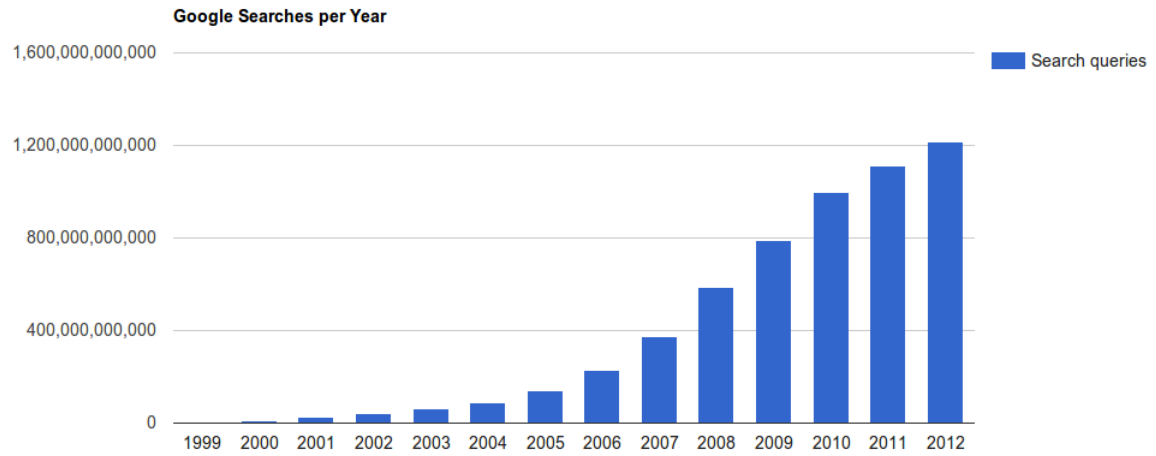


FIGURE 2.1 – Statistiques sur le nombre de requêtes soumises au moteur de recherche Google. http://www.internetlivestats.com/, le 2/03/2016.

La RI s'intéresse à l'acquisition, l'organisation, la recherche et la sélection de l'information. Le défi est de pouvoir trouver l'information qui correspond le mieux à l'attente de l'utilisateur. Idéalement, pour proposer une réponse pertinente à l'utilisateur, le système devrait pouvoir connaître le contexte de la recherche, son niveau de connaissance par rapport à son besoin ou encore ses connaissances dans des thématiques connexes. L'opérationnalisation de la RI est réalisée par des outils qu'on appelle Systèmes de Recherche d'Information (SRI). Un SRI prend en entrée une requête formulée par un utilisateur puis récupère des données au sein d'une collection préalablement indexée. Historiquement, la RI fait principalement référence à la recherche documentaire, autrement dit, les données que le SRI récupère sont des documents entiers contenant des informations jugées comme pertinentes par rapport à la requête de l'utilisateur.

L'évaluation d'un SRI consiste à mesurer ses performances vis-à-vis du besoin de l'utilisateur, cependant plusieurs méthodes d'évaluation adoptées en RI sont basées sur un modèle qui fournit une base d'évaluation comparative de l'efficacité de différents systèmes qui utilisent les mêmes ressources. Ces ressources sont essentiellement des collections de tests, des requêtes préalablement construites (ou crawlées), des algorithmes de sélection et de collecte, des jugements de pertinence et des métriques d'évaluation BOURAMOUL [2014].

Tout au long de ce chapitre, notre intérêt se porte ainsi sur les principes de la RI. La section 2.2 décrit ses concepts de base ainsi que les différents modèles. Dans la section 2.3,

1. http://www.planetoscope.com/Internet-/1523-informations-publiees-dans-le-monde-sur-le-net-en-gigaoctets-.html

nous donnons un aperçu sur des méthodes de reformulation des requêtes des utilisateurs. La section 2.4 est consacrée à l'évaluation des SRI et dans la section qui suit (2.5), nous présentons les collections de tests populaires et largement utilisées dans le domaine de la RI.

2.2 Concepts de base d'un système de recherche d'information

Un SRI a pour principale fonction de fournir aux utilisateurs les informations (qu'elles soient textuelles, visuelles ou sonores) qui répondent à leurs besoins informationnels. Un SRI intègre un ensemble de modèles pour la représentation des unités d'information (documents et requêtes) ainsi qu'un processus de recherche/décision qui permet de sélectionner l'information pertinente en réponse au besoin exprimé par l'utilisateur. Globalement, un SRI est composé de trois éléments :

1. Un modèle de document pour l'indexation ;
2. Un modèle de représentation de la requête ;
3. Une fonction de correspondance entre la requête et le document.

Cette composition constitue le processus de recherche d'information qui est illustré dans la figure 2.2.

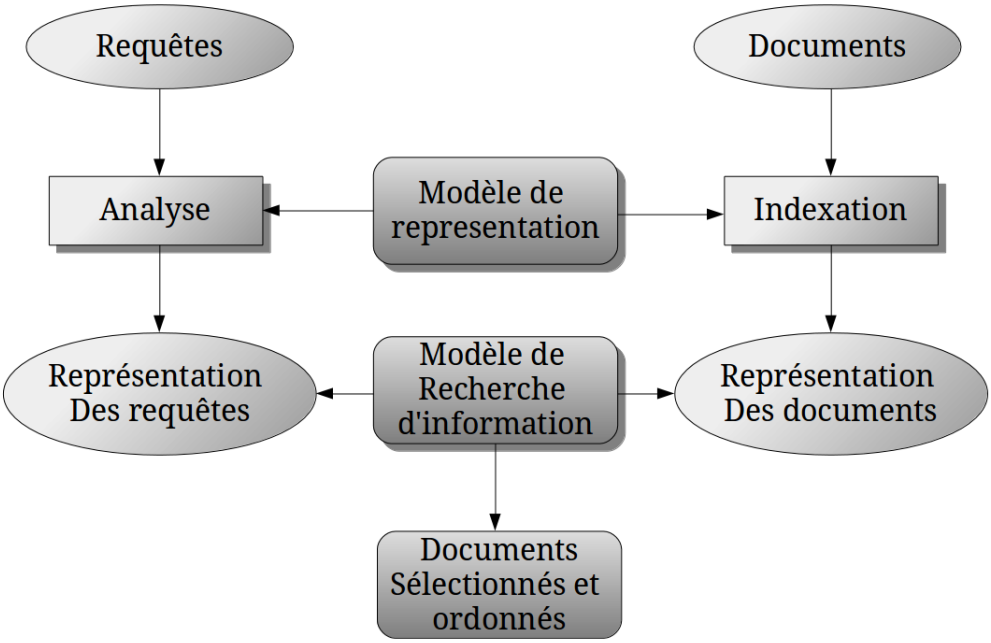


FIGURE 2.2 – Processus en U de recherche d'information

Les requêtes représentent le besoin en information exprimé par l'utilisateur. La littérature propose divers types de langages d'interrogation pour formuler une requête. Nous citons et détaillons les plus répandus dans la section 2.3 de ce chapitre. Dans ce qui suit, nous détaillons les autres éléments du processus.

2.2.1 Indexation des documents

Dans un processus de RI le coût de recherche doit être acceptable, il est donc nécessaire de procéder à une phase primordiale pour l'optimisation du temps d'exécution de ce proces-

sus. Cette phase s'appelle l'**indexation**, elle consiste à traduire une requête ou un document d'une représentation brute vers une représentation structurée selon un ensemble de règles et notations prédéfinies. Le résultat de l'indexation constitue le descripteur du document ou de requête, il est souvent représenté par une liste de termes ou groupe de termes significatifs pour l'unité textuelle correspondante. Les descripteurs des documents (mots, groupe de mots) forment le langage d'indexation. L'étape d'indexation influence en premier les performances d'un SRI, sa qualité dépend en partie des réponses du système.

L'indexation peut être manuelle, automatique ou semi-automatique.

— **Indexation manuelle** :
L'indexation manuelle demeure assez répandue dans plusieurs services commerciaux. Dans ce type d'indexation, les documents de la collection sont analysés un à un par un spécialiste du domaine ou par un expert documentaliste qui détermine manuellement, les mots clés qui lui semblent les plus significatifs pour représenter le document. Cette indexation se caractérise par sa profondeur, sa cohérence et sa qualité par contre elle est dépendente des connaissances des indexeurs et demande beaucoup de ressources en experts et temps pour la lecture des documents. Son application s'avère impossible pour une collection de documents volumineuse.

CLEVERDON [1967] a montré sur une collection de 1 400 documents et 221 requêtes, que l'indexation manuelle limitée à des termes simples choisis librement s'avère plus performante que l'indexation basée sur des termes et syntagmes extraits uniquement d'une liste de vocabulaire contrôlé (ces deux formes d'indexation étant manuelles). Le lien entre indexation manuelle et vocabulaire contrôlé n'est pas essentiel à une bonne performance.

— **Indexation automatique** :
Cette indexation est actuellement la plus utilisée car elle pallie la majorité des problèmes de l'indexation manuelle. L'indexation automatique est élaborée par un programme informatique qui détecte automatiquement les termes les plus représentatifs du contenu du document en analysant son texte mot à mot. Tous les mots outils qui ne jouent qu'un rôle syntaxique sont éliminés grâce à un anti-dictionnaire. Pour mettre en évidence les diverses contributions des mots extraits dans la représentation d'un document, un poids lui est attribué.

— **Indexation semi-automatique** :
Dans ce cadre, un premier processus automatique permet d'extraire les termes du document. Cependant le choix final reste au spécialiste du domaine ou au documentaliste pour établir les relations entre les mots clés et choisir les termes significatifs.

2.2.2 Appariement document-requête

Le processus d'appariement document-requête est le noyau d'un SRI. Il permet de calculer la proximité du vocabulaire entre documents et requêtes en associant à chaque document une valeur de pertinence par rapport à une requête. Les documents ayant une valeur positive sont sélectionnés. Cette valeur est calculée à partir d'une fonction de similarité notée RSV(Q,d) (Retrieval Status Value), où Q est une requête et d un document. Elle tient compte des poids des termes souvent déterminés en fonction d'analyses statiques et probabilistes.

On parle souvent de la *pertinence utilisateur* et la *pertinence système*. La première correspond au jugement de l'utilisateur sur la réponse retournée par le SRI en fonction de son besoin en information et la deuxième est la mesure d'évaluation de la similarité entre le document et la requête (Retrieval status value, RSV).

2.2.3 Les modèles de recherche d'information

Il existe un grand nombre de modèles de recherche d'information, et ces modèles diffèrent principalement par la façon dont les informations disponibles sont représentées, et par la façon d'interroger la base documentaire.

Les principaux modèles sont : les modèles **booléens**, les modèles **algébriques** ou appelés aussi **vectoriels**, les modèles **probabilistes**, les modèles de **langage** et les modèles **inférentiels bayésiens**.

2.2.3.1 Le modèle booléen

En ce qui concerne la représentation des documents, le modèle de recherche booléen peut être considéré comme le plus simple et le plus rapide à mettre en œuvre. Il a été le premier modèle utilisé en RI SALTON [1973]. L'interface d'interrogation de la plupart des moteurs de recherche (comme exemple : Google) est basée sur les principes de ce modèle. Il est composé d'une liste de termes (mots-clés) pouvant être combinés à des opérateurs logiques **et**, **ou** et **non**.

La formulation de la requête se base sur les trois opérateurs booléens :

— La conjonction **et** ($\wedge$), exige que les termes soient présents simultanément dans la description d'un document,

— La disjonction **ou** ($\vee$), exige qu'au moins un des termes soit présent dans la description des documents retournés,

— La négation **non** ($\neg$), utilisée pour écarter les documents qui contiennent un terme donné.

Une requête Q est donc composée de termes liés par les 3 opérateurs booléens présentés comme par exemple : $Q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$.

Le modèle booléen considère que les termes de l'index sont présents ou absents dans un document, en conséquence, les poids de ces termes sont binaires ($w_{ij} = \{0, 1\}$). Un document d est représenté comme une conjonction logique des termes non pondérées. Exemple : $d = t_1 \wedge t_2 \wedge t_3 ... \wedge t_n$.

La fonction de correspondance dans ce cas consiste à vérifier si le descripteur de chaque document d implique l'expression logique de la requête Q de l'utilisateur. Le résultat de cette fonction est binaire. Ainsi la similarité document requête est calculée par :

$$\mathrm{RSV}(Q, d) = \begin{cases} 1, \ si\ d\ appartient\ \text{à}\ l'ensemble\ décrit\ par\ Q \\ 0, \ sinon \end{cases} \tag{2.1}$$

Le modèle booléen présente quelques lacunes dans la présentation des résultats. La correspondance entre un document et une requête est binaire, en conséquence, le résultat de recherche est une liste de documents non ordonnée. Une autre étape est exigée pour trouver les documents qui intéressent les utilisateurs. Ceci s'avère difficile dans le cas où la liste des résultats est volumineuse. Une des autres lacunes du modèle booléen est la définition de l'importance des termes dans les documents ou requêtes qui ne sont pas pondérés.

Le modèle booléen standard n'est utilisé que dans très peu de systèmes de nos jours. Si on utilise un modèle booléen, c'est plutôt une extension de ce modèle. Les extensions proposées essaient justement de corriger les lacunes du modèle standard.

2.2.3.2 Le modèle algébrique (vectoriel)

Le modèle vectoriel standard est un modèle de recherche d'information très connu. Il intègre dans un espace vectoriel une représentation qui symbolise les documents ou les requêtes en fonction des termes d'indexation qui les composent. La forme d'implémentation la plus connue du modèle vectoriel est le système de recherche documentaire SMART SALTON [1968]; SALTON et MCGILL [1983].

Ce modèle préconise la représentation des requêtes utilisateurs et documents sous forme de vecteurs, dans l'espace engendré par les N termes d'indexation. Cet ensemble est défini par l'ensemble de termes que le système a rencontré durant l'indexation.

Soit l'espace vectoriel suivant $< t_1, t_2, t_3, ..., t_n >$, chaque document et requête sont respectivement représentés par un vecteur document et un vecteur requête :

— $d_j = (w_{1j}, w_{2j}, ..., w_{nj})$, $w_{ij} \in [0\ 1]$ est le poids du terme t_i dans le document d_j.

— $q = (w_{1q}, w_{2q}, ..., w_{nq})$, $w_{iq} \in [0\ 1]$ est poids du terme t_i dans la requête q.

La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Cela implique que la pertinence d'un document relativement à une requête est reliée à la mesure de similarité des vecteurs associés. Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs et se présente comme suit SALTON [1971] :

$$\mathrm{RSV}(q,d) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}}$$

Plus les vecteurs sont similaires, plus l'angle formé est petit, et plus le cosinus de cet angle est grand. À l'inverse du modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui permet de retrouver des documents qui ne reflètent la requête qu'approximativement. Les résultats peuvent donc être ordonnés par ordre de pertinence décroissante.

L'utilisation répandue du modèle vectoriel en recherche d'information est principalement due à l'uniformité de son modèle de représentation requête-document, l'ordre induit par la fonction de similitude ainsi que les possibilités aisées offertes pour ajuster les fonctions de pondération des termes des documents ainsi que des requêtes afin d'améliorer les résultats de la recherche.

Toutefois, le modèle vectoriel présente un inconvénient majeur qui est le fait qu'il suppose que les termes d'indexation forment une base. Or il existe énormément de relations sémantiques qui font qu'un terme pourra s'exprimer différemment en fonction du contexte. Par ailleurs il est très difficile voire impossible de traduire des relations sémantiques par des combinaisons linéaires de termes.

2.2.3.3 Le modèle probabiliste

MARON et KUHNS [1960] étaient à l'origine de l'apparition de modèle probabiliste. Il est fondé sur le calcul de la probabilité de pertinence d'un document pour une requête. La similarité entre un document et une requête est mesurée par le rapport entre la probabilité qu'un document d donné soit pertinent pour une requête Q, notée $p(d/\mathrm{Q})$, et la probabilité qu'il soit non pertinent et $p(\bar{d},\mathrm{Q})$. Ces probabilités sont estimées par les probabilités conditionnelles selon qu'un terme de la requête est présent, dans un document pertinent ou dans un document non pertinent. Les documents et les requêtes sont représentés par des vecteurs booléens dans un espace à n dimensions. Un exemple de représentation d'un document d_j et

une requête Q est comme présenté dans 2.2.3.2. Le score d'appariement entre le document d_j et la requête Q, noté RSV(Q, d_j) est donné par :

$$\text{RSV}(\text{Q}, d_j) = \frac{p(d/\text{Q})}{p(\bar{d}, \text{Q})} = \sum_{i=1}^{t} log \frac{p(1-q_i)}{q(1-p_i)}$$

Où :

— p_i : est la probabilité que le terme t_i est présent dans le document d sachant que celui ci est pertinent,

— q_i : est la probabilité que le terme t_i est présent dans le document d sachant que celui ci n'est pas pertinent,

— t : est le nombre total de termes dans la requête.

Le modèle probabiliste a donné lieu à de nombreuses extensions comme exemple, le système OKAPI ROBERTSON et WALKER [1999]; ROBERTSON et collab. [1994].

2.2.3.4 Les modèles de langue

L'hypothèse de base des modèles de recherche précédents consiste à dire qu'un document n'est pertinent que s'il ressemble à la requête. Autrement dit, dans ces modèles, on cherche à mesurer la similarité entre un document d et une requête q ou à estimer la probabilité que le document répond à la requête. L'hypothèse des modèles de langue est différente. Elle consiste à ordonner chaque document d de la collection C suivant leur capacité à générer la requête de l'utilisateur q. Ainsi, il s'agit d'estimer la probabilité de génération P(q/d). Pour simplifier, on suppose en général que les mots qui apparaissent dans la requête sont indépendants les uns des autres. Pour une requête $q = t_1 t_2 ... t_n$, cette probabilité de génération est estimée comme suit :

$$\text{P}(q/d) = \text{P}(t_1 t_2 ... t_n/d) = \prod_{t \in q} \text{P}(t/d)$$

Cette mesure, introduite par PONTE et CROFT [1998] a été reprise par BERGER et LAFFERTY [1999] pour proposer leur modèle.

2.2.3.5 Le modèle inférentiel bayésien

Dans un modèle inférentiel bayésien, on se base sur un graphe de dépendances, direct et acyclique CALLAN [1996]; TURTLE et CROFT [1990, 1991]. Dans ce graphe, les nœuds représentent des variables aléatoires propositionnelles et les arcs des relations causales entre les nœuds. Ainsi, si le nœud p représente une proposition qui cause ou implique la proposition représentée par le nœud q, on trace un arc allant de p vers q. Les nœuds sont pondérés par des valeurs de probabilités conditionnelles.

Dans le contexte de la recherche d'information et dans l'espace défini par les termes d'indexation, les nœuds représentent des concepts, des groupes de termes ou des documents entiers et les arcs représentent les dépendances entre termes et entre termes et documents. On définit :

— T variables aléatoires binaires $t_1 t_2 ... t_n$ associées aux termes d'indexation.

— D_j : variable aléatoire associée à un document.

— Q : variable aléatoire associée à une requête.

La mesure de pertinence de Q relativement au D_j en traitant les probabilités conditionnelles de Bayes se calcule comme suit :

$$RSV(Q, D_j) = 1 - P(Q \wedge D_j)$$

où :

$$P(Q \wedge D_j) = \sum_{i=1}^{T} P(Q|t_i) * (\prod_{t_i \notin D_j} P(\bar{t_i}|D_j)) * P(D_j)$$

Avec :

— $P(Q|t_i)$ est la probabilité que le terme t_i appartienne à un document pertinent de Q ;

— $P(t_i|D_j)$ est la probabilité que le terme t_i appartienne au document D_j sachant qu'il est pertinent ;

— $P(\bar{t_i}|D_j) = 1 - P(t_i|D_j)$

— $P(D_j)$ est la probabilités d'observer D_j

Les probabilités conditionnelles de chaque nœud sont calculées par propagation des liens de corrélation entre eux.

Le modèle inférentiel bayésien présente l'intérêt de considérer la dépendance entre termes mais engendre une complexité de calcul importante.

2.3 Reformulation de requête

La reformulation de requête est proposée dans le but d'adapter le SRI au mieux aux besoins des utilisateurs. C'est un processus qui permet de générer une requête plus adéquate à la recherche d'information dans l'environnement du SRI, que celle initialement formulée par l'utilisateur. De ce fait plusieurs techniques ont été proposées pour améliorer les performances des SRI. Les méthodes de reformulation aident à trouver plus de documents pertinents vis-à-vis d'une requête donnée et à exprimer la requête de l'utilisateur de manière à mieux répondre à son besoin.

Il existe plusieurs techniques de reformulation de requête. Nous présentons dans ce qui suit la reformulation par expansion automatique de la requête et par réinjection de pertinence (Relevance Feedback).

La figure 2.3 présente le processus d'un SRI employant des techniques de reformulation de la requête initiale de l'utilisateur. La reformulation peut se faire par expansion de la requête ou par réinjection de pertinence (Relevance Feedback).

2.3.1 Expansion automatique de requête

Le but de l'expansion de requête est d'élargir l'ensemble de documents retournés par le SRI. Dans ce cas, la requête peut être étendue en ajoutant des termes similaires à ceux présents dans la requête initiale. Ces termes sont issus de ressources linguistiques existantes, par exemple, des bases de données lexicales comme Wordnet MILLER [1995] qui est une ressource qui reste attirante pour l'expansion de requête, car, en tant qu'humains, nous avons l'impression que l'ajout de termes linguistiquement cohérents avec ceux de la requête donnera forcément de meilleurs résultats.

Wordnet a été utilisé en premier par VOORHEES [1994]. Cette dernière a conclu que le succès de sa méthode de reformulation qui utilise Wordnet était lié à deux éléments : la

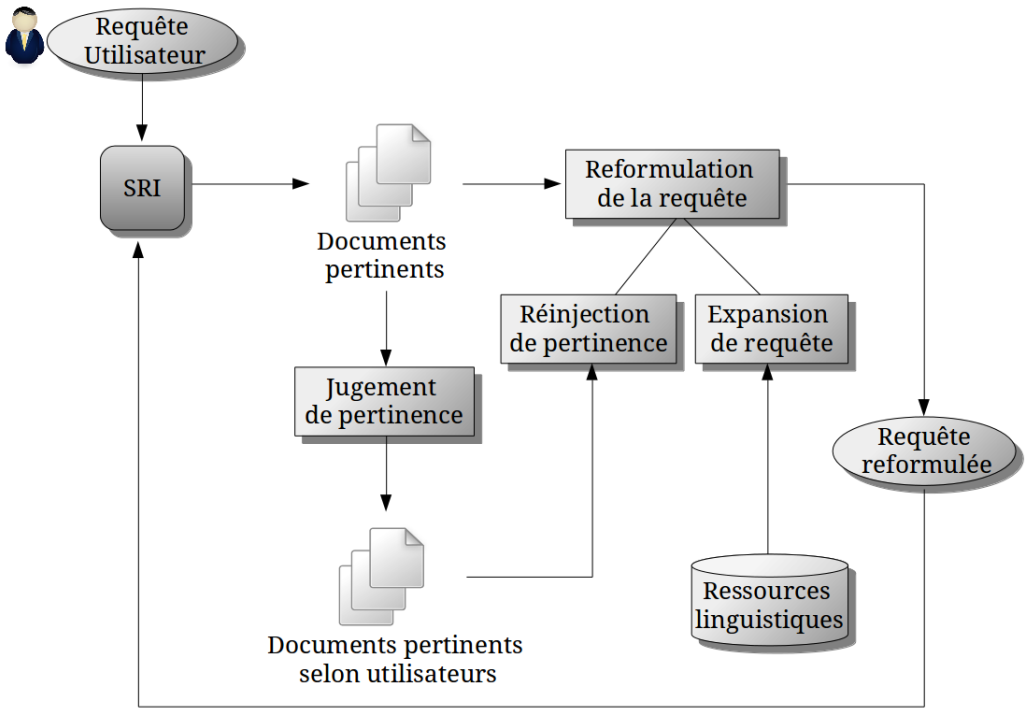


FIGURE 2.3 – Processus d'un SRI employant des techniques de reformulation de requête

qualité de la requête initiale et celle des synonymes trouvés. Le fait que la requête initiale exprime bien le besoin en information, les performances ne s'améliorent pas après l'ajout des synonymes. Par contre, l'ajout automatique et sans désambiguïsation des synonymes dégrade les performances de la recherche. Pour pallier ce problème, VOORHEES [1994] a choisi manuellement les synonymes dans Wordnet et a réussi à améliorer les requêtes difficiles. Ce choix reste coûteux en terme de temps et de ressources humaines si on a des bases de requêtes volumineuses.

NAVIGLI et VELARDI [2003] ont proposé une étude sur les meilleurs termes d'expansion à choisir après que la désambiguïsation (choix des synsets dans Wordnet) est faite. Cette étude montre que l'utilisation de la définition dans le glossaire de Wordnet est plus performante pour l'expansion de la requête que les synonymes et hyperonymes. Une autre étude réalisée par FANG [2008], a montré une amélioration significative des performances lors de l'expansion en utilisant les définitions de synsets et en pondérant les termes d'expansion provenant de Wordnet.

2.3.2 Reformulation par réinjection de pertinence

La reformulation de requête par réinjection de pertinence est souvent connue sous le nom de *Relevance Feedback* (BOUGHANEM et collab. [1999]; ROCCHIO, JR. [1971]; SALTON et BUCKLEY [1990]). Cette méthode permet une modification de la requête initiale, sur la base des jugements de pertinence de l'utilisateur sur les documents sélectionnés par le SRI. La Relevance Feedback est un processus qui comporte principalement trois étapes : l'échantillonnage, l'extraction des évidences et la réécriture de la requête.

— *Échantillonnage* : c'est la construction d'un échantillon de documents à partir des éléments jugés pertinents par l'utilisateur.

— *Extraction des évidences* : est l'étape la plus importante, elle consiste à extraire les termes qui serviront à l'enrichissement de la requête initiale. Plusieurs approches ont

été développées, la plus connue est celle de ROCCHIO, JR. [1971].

— *Réécriture de la requête* : consiste à construire une nouvelle requête en combinant la requête initiale avec les termes extraits dans l'étape précédente.

Ce processus de reformulation peut être renouvelé plusieurs fois pour le même besoin en information, on parle alors de réinjection de pertinence à itérations multiples. La nouvelle requête obtenue à chaque itération de réinjection, permet de corriger la direction de recherche dans le sens des documents pertinents.

2.4 Évaluation des systèmes de recherche d'information

L'évaluation d'un SRI constitue une étape importante lors de sa mise en œuvre. Elle est apparue avec les premiers prototypes KENT et collab. [1955] et est considérée comme un problème majeur par la communauté de la RI qui a investi beaucoup d'efforts pour essayer de le résoudre CLEVERDON et KEAN [1968]; HULL [1993]; MEHLITZ et collab. [2007]; VOORHEES [2002]. L'évaluation d'un SRI permet de paramétrer le modèle de recherche d'information utilisé, d'estimer l'impact de chacun de ses caractéristiques et enfin de fournir des éléments de comparaison entre modèles. Plusieurs quantités mesurables ont été proposées pour l'évaluation d'un SRI : le temps de réponse, la pertinence, la qualité et la présentation des résultats, etc.

2.4.1 Évaluation de la pertinence

En recherche d'information, un document peut être jugé par plusieurs utilisateurs ou dans des conditions différentes d'une façon pas forcement booléenne, les utilisateurs peuvent attribuer une note de pertinence aux documents comme le cas avec quelques sites commerciaux (Amazon, etc.). Cet aspect de jugement peut influencer également l'ordonnancement des documents retournés par le SRI de telle sorte qu'un utilisateur peut considérer un document plus pertinent qu'un autre, même si ce document est aussi considéré comme pertinent par le même utilisateur. Dans ce cas, l'évaluation idéale de la pertinence d'un document doit être effectuée par l'utilisateur qui a posé la requête durant la même session de recherche, ce qui n'est pas le cas dans la plupart des campagnes d'évaluation comme nous allons le présenter dans la section 2.5.

2.4.2 Mesures d'évaluation

Pour comparer les SRI entre eux, des mesures d'efficacité ont été introduites. Ces mesures sont des métriques objectives qui reposent sur la notion de pertinence, leur efficacité est donc liée à la qualité des jugements de pertinence. La plupart des métriques proposées en recherche d'information se basent sur deux notions historiques : la précision et le rappel CLEVERDON et collab. [1966].

Le rappel (R) mesure la capacité d'un SRI à sélectionner tous les documents pertinents. Il donne une indication sur le nombre de documents pertinents trouvés par rapport au nombre total de documents pertinents pour la requête.

La précision (P) mesure la capacité du système à rejeter tous les documents non pertinents. Elle donne une indication sur la proportion des documents pertinents renvoyés par

le SRI. Elle est d'une part un indicateur pour la qualité du résultat à la demande de la recherche, et d'autre part, elle sert à ne pas distribuer des documents non pertinents afin de ne pas saturer la capacité d'un système.

Sachant que D_r est l'ensemble de documents retournés par le SRI pour une requête donnée et D_p est l'ensemble de documents pertinents à cette même requête, les valeurs du rappel et de la précision sont entre 0 et 1 et se calculent comme suit :

$$R = \frac{|D_r \cap D_p|}{|D_p|}$$

$$P = \frac{|D_r \cap D_p|}{|D_r|}$$

Le rappel et la précision, sont les mesures les plus populaires en RI, ces deux mesures représentent deux aspects qui sont différents pour les documents retrouvés. La combinaison de ces deux mesures d'évaluation qui peut être la plus appropriée parmi les mesures qui ont été proposées est : la mesure *Harmonique* F SHAW et collab. [1996]. C'est une mesure unique qui combine le rappel et la précision, sa valeur est entre 0 et 1. La F-mesure se calcule comme suit :

$$F = \frac{2 * R * P}{R + P}$$

Pour avoir une évaluation de la performance du système sur toutes les requêtes et non pas sur une seule, on calcule une précision moyenne à chaque niveau de rappel appelée **MAP** (*Mean Average Precision*). Pour ce faire, il faut unifier les niveaux de rappel pour l'ensemble des requêtes. On retient généralement 11 points de rappel standards, de 0 à 1 à pas de 0,1. Les valeurs de précision non obtenues à partir des valeurs de rappel sont calculées comme suit, par interpolation linéaire.

Pour deux points de rappel, i et j, $i < j$, si la précision au point i est inférieure à celle au point j, on dit que la précision interpolée à i égale la précision à j. Formellement :

$$p_i^{'} = max(p_i, p_j), \forall i < j$$

où $p_i^{'}$ est la précision interpolée au point de rappel i, et p_i est la vraie précision au point de rappel i. Cette interpolation est encore discutable, mais présente un intérêt dans l'évaluation de SRI SALTON et FOX [1983].

Une autre mesure utilisée pour l'évaluation des SRI s'appelle NDCG (*Normalized Discounted Cumulative Gain*)Le NDCG pour la requête q au rang k, NDCG@$k(q)$, est calculé à partir du DCG@$k(q)$ (*Discounted Cumulative Gain*) défini de la façon suivante :

$$DCG@k(q) = \sum_{i=1}^{k} \frac{2^{(i)} - 1}{log_2(i+1)}$$

où $r^{(i)}$ correspond au degré de pertinence du document à la position i. Le DCG est une mesure qui présente l'inconvénient de ne pas être à valeur dans l'intervalle [0,1]. Le NDCG@$k(q)$ est alors normalisé par le terme Z_k, la valeur maximale du DCG@$k(q)$, $\forall k$ et se calcule comme suit :

$$NDCG@k(q) = \frac{1}{Z_k} DCG@k(q)$$

La valeur du NDCG@$k(q)$ est comprise dans l'intervalle [0,1]. Le NDCG@$k(q)$ d'un ensemble des requêtes du jeu de données est défini comme la moyenne des NDCG@k de chaque requête.

2.5 Campagnes d'évaluation

Les collections de tests sont utilisées pour juger de l'efficacité des SRI et ainsi faire évoluer leur performance. Différentes collections de test sont utilisées en recherche d'information. Parmi elles nous citons : la collection TREC (Text REtrieval Conference) et INEX (INitiative for Evaluation of XML Retrieval) qui est intégrée à CLEF (Cross Language Evaluation Forum) depuis 2014 et NTCIR.

2.5.1 TREC : Text REtrieval Conference

TREC est un projet international qui a été lancé en 1992 HARMAN [1992] par le **NSIT** (*National Institute of Standards and Technology*) aux États-Unis. Il est co-sponsorisé par le **NIST** et **DARPA/ITO** (*Defense Advanced Research Projects Agency - Information Technology Office*). Cette campagne offre une très large collection de documents de sources très variées : Financial Time, Résumés de publications USDOE, SAN JOSE Mercury news, etc., organisées sous différentes collections qui évoluent chaque année. Pour chaque session de TREC, un ensemble de documents et de requête est fourni. Les participants exploitent leurs propres systèmes de recherche sur les données et renvoient à **NIST** une liste ordonnée de documents. **NIST** évalue ensuite les résultats.

Différents éléments constituent le projet TREC. Parmi ces éléments, on a :

— **Tâches** : chaque année, des tâches sont définies dans le but d'évaluer des approches spécifiques en recherche d'information concernant le filtrage, le croisement de langues, la recherche dans de très large corpus (100 giga octet et plus), les modèles d'interactions, etc. Dans TREC 2015, la liste des tâches est la suivante :

1. *Clinical Decision Support Track*
2. *Contextual Suggestion Track*
3. *Dynamic Domain Track*
4. *Live QA Track*
5. *Microblog Track*
6. *Tasks Track*
7. *Temporal Summarization Track*
8. *Total Recall Track*

— **Les participants** : Dans la première édition de TREC 25 groupes ont participé en 1992, ce nombre n'a cessé d'augmenter chaque année pour atteindre 93 groupes en 2003.

— **Structure de la collection** : Les documents de la collection TREC sont sous format XML et Json. Chaque document est identifié par un numéro et décrit par un auteur, une date de production et un contenu textuel. Il y a aussi les requêtes qui sont également identifiées par un numéro. Chaque requête est décrite par un sujet générique, une description brève et une description étendue sur les caractéristiques des documents pertinents associés à la requête.

2.5.2 NTCIR : NII Testbeds and Community for Information access Research

Cette campagne entre dans le champ de l'accès à l'information. Plusieurs disciplines sont couvertes telles que la RI, le résumé automatique, l'extraction d'information et le QR (Question-Réponse). L'objectif de la campagne NTCIR est d'évaluer l'efficacité des systèmes développés dans ces disciplines. Son cadre de travail porte sur l'étude des langues asiatiques et, en particulier, le chinois classique, le chinois simplifié et le japonais. Le deuxième objectif de NTCIR est la création de corpus textuels multiligues à grande échelle et réutilisable pour l'expérimentation. Le premier NTCIR a eu lieu en 1999, chaque NTCIR est constitué d'un lot de tâches avec en particlier la tâche *Adhoc*. Pour l'année 2015/2016 (NTCIR-12), la liste des tâches est la suivante NORIKO et MAKOTO [2016] :

1. *Search Intent and Task Mining ("IMine-2")*
2. *Medical Natural Language Processing for Clinical Document ("MedNLPDoc")*
3. *Mobile Information Access ("MobileClick-2")*
4. *Spoken Query and Spoken Document Retrieval ("SpokenQuery&Doc-2")*
5. *Temporal Information Access ("Temporalia-2")*
6. *Mathematical Information Retrieval (MathIR)*
7. *Lifelog Task ("Lifelog")*
8. *QA Lab for Entrance Exam ("QALab-2")*
9. *Short Text Conversation Task ("STC")*

2.5.3 CLEF : Conference and Labs of the Evaluation Forum

Cette compagne offre une infrastructure pour diverses tâches comme : l'évaluation des systèmes mutilingues et multimodaux ; possibilité de réglage de paramètres ; utilisation et accès aux données non structurées, semi-structurées ou très structurées ainsi que les données sémantiquement enrichies ; création des collections de test réutilisables pour les analyses comparatives ; comparaison des approches et échange des idées afin de partager les connaissances PETERS [2001].

Différentes tâches apparaissent au fil du temps dans le cadre de plusieurs initiatives CLEF, telles que :

1. Dans le domaine médical, la piste *CLEF eHealth* qui vise à développer des méthodes et des ressources dans un cadre multilingue afin d'améliorer la compréhension des textes médicaux. Dans ce cadre, on retrouve deux tâches : *Information Extraction from Clinical Data* et *User-centred Health Information Retrieval*.
2. Dans le domaine de traitement d'image, la piste *ImageCLEF* qui a pour objectif l'évaluation automatique des annotations et d'indexation des images. On retrouve les tâches suivantes : *Image Annoation*, *Medical Classification*, *Medical Clustering*, *Liver CT Annoation*
3. La piste *Question Answering* qui vise à répondre par des réponses précises à des questions en langage naturel. Les tâches associées sont : *Question Answering over Linked Data*, *Entrance Exams*, *Large-Scale Biomedical Semantic Indexing*, *Biomedical Question Answering*

2.5.4 INEX : INitiative for Evaluation of XML Retrieval

La campagne INEX (*INitiative for Evaluation of XML Retrieval*) est lancée en 2002 et a été intégrée à CLEF (Conference and Labs of the Evaluation Forum) Labs [2] en 2014. Elle vise à permettre une évaluation et une comparaison aussi rigoureuse que possible des SRI dans les collections XML orientées documents.

Ainsi INEX fournit un ensemble de documents, un ensemble de requêtes et des jugements de pertinence, c'est-à-dire les estimations humaines des éléments pertinents concernant chaque requête. Les tâches proposées par INEX sont :

1. *Social Book Search (2011-2014)*
2. *Linked Data (2012-2013)*
3. *Data Centric (2011)*
4. *Tweet Contextualization (2011-2014)*
5. *Snippet Retrieval (2011-2013)*
6. *Relevance Feedback (2011-2012)*

Dans le cadre de cette thèse, nous nous intéressons à la tâche *Social Book Search (SBS)* [3]. La collection lors des éditions de 2013 KOOLEN et collab. [2013], 2014 HALL et collab. [2014] et 2015 KOOLEN et collab. [2015a], regroupait 2,8 millions de descriptions de livres issues d'Amazon [4] et enrichis avec des contenus à partir du réseau social de lecteurs LibraryThing [5]. Chaque description de livre est identifiée par un ISBN est écrite dans un fichier XML. Un extrait d'une description de livre est fourni dans les figures 2.4 et 2.5.

Dans un document de la collection SBS, des méta-données concernant la publication et le contenu (ISBN, auteur, titre, date de publication, l'éditeur, les dimensions du livre, description du livre par Amazon, etc.) sont d'abord fournies, puis viennent des méta-données sociales comme les commentaires des utilisateurs, leurs notes, leurs dates de publication, les votes sur l'utilité des commentaires, les produits similaires fournis par Amazon, etc.

Un autre élément majeur dans la tâche SBS est l'ensemble des requêtes appelées *topics*. Un topic est la représentation la plus fidèle possible d'un besoin d'information. Ils sont crawlés à partir des forums de discussion de LibraryThing. L'ensemble des topics est renouvelé chaque année par les organisateurs de la tâche SBS. Chaque topic contient plusieurs méta-donnée (titre, le groupe d'appartenance de l'utilisateur qui a posé la requête, le contenu en langage naturel du besoin et pour l'année 2015 le catalogue de livres de l'utilisateur) comme illustré dans la figure 2.6.

2. http://clef2015.clef-initiative.eu/CLEF2015/lab_overview.php
3. http://social-book-search.humanities.uva.nl/
4. http://www.amazon.com/
5. http://www.librarything.com/

FIGURE 2.4 – Extrait d'un document XML de la collection INEX SBS. (Voir la suite de l'extrait dans la figure 2.5)

```
<book>
  <isbn>003014213X</isbn>
  <title>Invitation to Critical Thinking</title>
  <ean>9780030142130</ean>
  <binding>Paperback</binding>
  <label>Harcourt School</label>
  <listprice>\$36.91</listprice>
  <manufacturer>Harcourt School</manufacturer>
  <publisher>Harcourt School</publisher>
  <readinglevel/>
  <releasedate/>
  <publicationdate>1990-01</publicationdate>
  <studio>Harcourt School</studio>
  <edition>2nd</edition>
  <dewey>160</dewey>
  <numberofpages>423</numberofpages>
  <dimensions>
    <height>100</height>
    <width>775</width>
    <length>925</length>
    <weight>145</weight>
  </dimensions>
  <reviews>
    <review>
      <authorid>A1HFQ2FKCXYME1</authorid>
      <date>2008-05-03</date>
      <summary>Very dry and difficult to comprehend.</summary>
      <content>
        Unless this book is an absolute must for a particular course,
            I'd stay away. The writing is good but almost seems to be
            written by authors who want to impress you with their
          intelligence rather than convey the concept of critical
          thinking. I'm sure there are better books on the market
          covering this topic. My opinion isn't a lone one. Even my
          professor didn't like the textbook but, due to red tape,
          had no choice to use it. I learned more from her lectures
          and handouts than anything else. And it was no surprise
          that most of my classmates agreed with me.
      </content>
      <rating>1</rating>
      <totalvotes>1</totalvotes>
      <helpfulvotes>1</helpfulvotes>
    </review>
  </reviews>
  ...
```

FIGURE 2.5 – Extrait d'un document XML de la collection INEX SBS. (Suite de la figure 2.4)

```
...
  <editorialreviews>
    <editorialreview>
      <source>Product Description</source>
      <content>
        As a primary text for critical thinking and practical
           reasoning courses, Rudinow and Barry s Third Edition
           offers practical coverage appropriate for the single
           semester course. Comprehensive discussions include new
           material on the topics of mass media and deductive
           validity, as well as argument forms, Venn diagrams, truth
           tables, inductive reasoning, informal fallacies, and
           problem solving.
      </content>
    </editorialreview>
  </editorialreviews>
  <images/>
  <creators>
    <creator>
      <name>Joel Rudinow</name>
      <role>Author</role>
    </creator>
  </creators>
  <subjects>
    <subject>Sami (European people) - Social conditions</subject>
  </subjects>
  <tags>
    <tag count="1">Reference</tag>
    <tag count="1">nc</tag>
    <tag count="1">critical thinking</tag>
    <tag count="1">untagged</tag>
    <tag count="1">logic</tag>
  </tags>
  <similarproducts>
    <similarproduct>0312436289</similarproduct>
    <similarproduct>0393310728</similarproduct>
    <similarproduct>0495095540</similarproduct>
    <similarproduct>0737729287</similarproduct>
    <similarproduct>0826498949</similarproduct>
  </similarproducts>
  <browseNodes>
     <browseNode id="53">Nonfiction</browseNode>
     <browseNode id="1000">Subjects</browseNode>
     <browseNode id="11019">Philosophy</browseNode>
     <browseNode id="11043">General</browseNode>
     <browseNode id="11053">Logic \& Language</browseNode>
     <browseNode id="283155">Books</browseNode>
</book>
```

FIGURE 2.6 – Exemple de topic.

```
<topic id="1116">
   <title>Which LISP?</title>
   <mediated_query>introduction book to Lisp</mediated_query>
   <group>Purely Programmers</group>
   <narrative>
     It'll be time for me to shake things up and learn a new
        language soon. I had started on Erlang a while back and
        getting back to it might be fun. But I'm starting to lean
        toward Lisp--probably Common Lisp rather than Scheme.
        Anyone care to recommend a good first Lisp book? Would I be
         crazy to hope that there's one out there with an emphasis
        on using Lisp in a web development and/or system
        administration context? Not that I'm unhappy with PHP and
        Perl, but the best way for me to find the time to learn a
        new language is to use it for my work...
   </narrative>
</topic>
```

2.6 Conclusion

Dans ce chapitre, nous nous sommes essentiellement intéressés à l'étude des systèmes de recherche d'information traditionnels, et les modèles utilisés pour les construire. Chacun de ces modèles ou stratégies contribue à la résolution des problèmes inhérents à la recherche d'information. Nous avons présenté des modèles qui traitent de la représentation des documents et des requêtes et d'autres traitent de la modification ou la reformulation de la requête.

La finalité de chaque système de recherche d'information est de satisfaire les besoins des utilisateurs. ces derniers sont préoccupés par un seul problème : celui de pouvoir récupérer tous les documents dont chaque utilisateur a besoin d'une façon rapide et efficace.

Nous avons présenté les techniques d'évaluation d'un SRI ainsi que les principales mesures d'évaluation utilisées dans la littérature suivies d'une brève présentation des campagnes d'évaluation très populaires dans le domaine de la RI.

Dans le chapitre suivant, nous présentons un autre scénario d'accès au contenu sans requêtes explicites : *les systèmes de recommandation*.

Chapitre 3

Recommandation automatique de lectures

Résumé : Dans ce chapitre, nous introduisons un état de l'art des systèmes de recommandation. Ces derniers sont de plus en plus développés dans plusieurs domaines d'application pour fournir des recommandations adaptées aux goûts, aux besoins ou aux moyens des utilisateurs afin de les aider à accéder à des ressources utiles ou intéressantes au sein d'un espace de données important. Ce chapitre définit les différentes techniques existantes pour la recommandation et particulièrement les techniques basées sur les graphes.

Sommaire

3.1 Introduction

Ces dernières années, de nombreuses entreprises et des sites Web utilisent des systèmes permettant l'analyse des préférences de leurs utilisateurs et leurs besoins dans le but d'améliorer la qualité de leurs services et produits. Cette alternative permet d'améliorer les performances des systèmes et ainsi de mieux répondre aux attentes des utilisateurs. C'est ce qu'on appelle la *recommandation*. En raison de l'explosion de la quantité de données diffusées sur Internet ces dernières années, rechercher et trouver des produits, des services ou des contenus pertinents est devenu une tâche difficile pour l'utilisateur qui est souvent perdu face à une telle masse d'informations. Ceci explique l'intérêt croissant porté aux systèmes de recommandation (SR) qui consistent à proposer aux utilisateurs des ressources qui peuvent correspondre à différents types de données tels que des films FISK [1997]; SAID et collab. [2010]; VLACHOS et SVONAVA [2013], des livres MOONEY et ROY [1999], des restaurants, des news KUO et collab. [2015]; MUI et collab. [2001]; PRONOZA et collab. [2014], de la musique BAUMANN et HUMMEL [2005]; BU et collab. [2010]; CELMA et collab. [2005], des recettes de cuisines JILLY FERNY [2010]; FREYNE et BERKOVSKY [2010]; TENG et collab. [2011], des articles scientifiques BOGERS et VAN DEN BOSCH [2008]; KERN et collab. [2014], etc.

Le développement des systèmes de recommandation s'est initié à partir d'une observation assez simple : les individus s'appuient souvent sur les recommandations des autres pour la prise de décisions quotidiennes MAHMOOD et RICCI [2009]. Par exemple, il est commun de s'appuyer sur ce que nos semblables recommandent lors du choix d'un livre à lire ; les employeurs comptent sur les lettres de recommandation pour leurs décisions de recrutement ; et pour la sélection de films à regarder, les individus tendent à lire et se fier aux critiques de films qui apparaissent dans les médias qu'ils ont l'habitude de suivre et leurs journaux, revues et sites habituels PICOT-CLÉMENTE [2011].

La problématique des systèmes de recommandation peut se résumer dans la question suivante : comment guider l'utilisateur dans son exploration de la masse de données disponibles afin qu'il trouve des recommandations pertinentes ?

Le processus de recommandation va guider l'utilisateur en cherchant à prédire l'avis qu'il donnerait à chaque élément et lui recommande ceux qui ont obtenu le meilleur avis prédit. C'est une forme particulière de filtrage visant à présenter des éléments susceptibles d'intéresser l'utilisateur suivant des facteurs qui diffèrent selon le type de recommandation.

Il est possible de classer les systèmes de recommandation de différentes manières. La plus classique évoquée dans la littérature est une classification selon trois approches : les systèmes de recommandation basés sur le contenu, le filtrage collaboratif et les systèmes de recommandation hybrides qui combinent les techniques des deux autres approches RICCI et collab. [2011b]. Il existe d'autres types de recommandation comme la recommandation basée sur les données géographiques, la recommandation basée sur la connaissance et la recommandation basée sur l'utilité, qui ont été proposées par BURKE [2007].

Nous décrivons dans la suite de ce chapitre, les trois principales classes de recommandation : basée sur le contenu dans la section 3.2.1, basée sur les utilisateurs (filtrage collaboratif) dans la section 3.2.2 et hybride dans la section 3.2.3. Nous présentons un l'état de l'art de la recommandation sur des contenus liés dans la section 3.3, suivi des méthodes d'évaluation de ces systèmes (section 3.4).

3.2 Les approches de recommandation

Les systèmes de recommandation sont définis comme étant "*des outils logiciels et des techniques qui suggèrent aux usagers des éléments utiles* "RICCI et collab. [2011a]. On emploie le terme général "item" pour dénoter les éléments de recommandation qui peuvent être de natures très différentes : documents textuels, images, vidéos, lieux, produits commerciaux, etc.

3.2.1 Systèmes de recommandation à base d'items

Les systèmes de recommandation à base d'items appelés aussi, systèmes de recommandation basés sur le contenu ou filtrage thématique. L'objectif principal de tels systèmes est de cibler des objets pertinents issus d'un large espace de sources possibles d'une façon personnalisée pour les utilisateurs. Son principe consiste à recommander des items similaires à ceux auxquels l'utilisateur s'est intéressé. Un profil est ainsi construit à partir de son historique pour représenter ses préférences ou ses intérêts. Le système met en relation ce profil avec les attributs des items afin de recommander de nouveaux items intéressants.

Les systèmes de recommandation basés sur le contenu prend place à l'intersection de différents domaines. En Recherche d'Information, l'utilisateur exprime un besoin ponctuel en donnant une requête (habituellement une liste de mots-clefs). Dans les systèmes de filtrage d'information (FI), le besoin est représenté par le profil de l'utilisateur. Les items à recommander peuvent être très différents, en fonction du nombre et du type des attributs utilisés pour les décrire. La tâche de recommandation peut être exprimée comme un problème d'apprentissage qui exploite l'historique des utilisateurs. Souvent, il est préférable que le système apprenne le profil de l'utilisateur plutôt que d'imposer à celui-ci de le fournir comme le cas avec Amazon. Cela implique généralement l'application de techniques d'apprentissage automatique ALPAYDIN [2004]. Leur but est d'apprendre à catégoriser de nouvelles informations en se basant sur les informations historiques, et qui ont été libellées implicitement ou explicitement comme intéressant ou non par l'utilisateur. Avec ces libellés, les méthodes d'apprentissage automatique sont capables de générer un modèle prédictif qui, étant donné un nouvel item, va aider à décider du degré d'intérêt que peut porter l'utilisateur pour l'item PICOT-CLÉMENTE [2011].

Dans les systèmes de recommandation basés sur le contenu, les items sont représentés sur un vecteur descripteur $X = (x_1, x_2, ..., x_n)$ de n composantes. Chaque composante représente un attribut (appelé aussi caractéristique ou propriété) et peut contenir des valeurs binaires, numériques ou encore nominales en fonction du domaine d'application. Par exemple, dans la recommandation de films, les attributs peuvent être le genre, le réalisateur, l'année de production, etc. Le vecteur d'attributs fait office de profil. L'objectif du moteur de recommandation est d'évaluer les similarités après la construction des profils. La notion de contenu ne se rapporte pas uniquement au contenu des ressources, mais également aux attributs descriptifs des utilisateurs. Pour recommander des ressources en se basant sur le contenu, deux éléments doivent être constitués : les profils de ressource et les profils d'utilisateur.

D'une manière générale, le profil d'un objet donné correspond à un ensemble de caractéristiques permettant son identification ou sa représentation.

— *Le profil de ressource ou d'item* : une ressource est généralement un document. Le profil correspond à une description réduite, dans la plupart des cas à une liste de mots-clés pondérés décrivant le contenu du document ADOMAVICIUS et TUZHILIN [2005]; PAZZANI et BILLSUS [2007]. Cette liste de mots subit généralement des traitements

linguistiques comme la racinisation (stemming), des corrections, des suppressions de mots outils, etc. Le profil de ressource peut également être construit de plusieurs façons. Tout d'abord en se basant sur des données concrètes comme le genre d'un film ou le type d'un document scientifique, on peut alors effectuer une analyse d'item et extraire des méta-données descriptives. Nous citons par exemple, les travaux de LAINÉ-CRUZEL [1999] qui permettent de définir des propriétés liées à l'ensemble d'un document (professions de l'auteur, type de document, etc.) ainsi que celles relatives à des parties de documents (type d'unité documentaire, forme discursive, style, etc.). C'est aussi le cas avec le site Pandora [1] qui analyse les morceaux musicaux sur environ 400 critères formant le profil de chaque morceau. Des similarités sont ensuite calculées entre les morceaux écoutés et appréciés par l'utilisateur et les morceaux qui n'ont pas encore été écoutés. Une autre méthode de construction des profils d'items consiste à se baser sur des informations apportées par les utilisateurs, généralement sous forme de textes. Ces informations peuvent être par exemple des données structurées, comme avec les systèmes de tags GODOY et AMANDI [2008], mais également des données non-structurées comme des textes descriptifs (synopsis), des critiques journalistiques ou encore des commentaires des utilisateurs. Ces textes peuvent être une source d'extraction des caractéristiques.

— *Le profil utilisateur* : Il s'agit de la description des caractéristiques de l'utilisateur issues de son centre d'intérêt et ses préférences (items appréciés ou pas). Dans ce cas, il s'agit d'une extraction automatique du profil et le système de recommandation cherche à trouver des items qui s'approchent le plus des items appréciés ainsi que d'ignorer tout item qui se rapproche des items que l'utilisateur n'a pas appréciés. Le profil utilisateur peut contenir aussi des descripteurs qui peuvent correspondre à d'autres traces laissées lors de navigation sur le Web : items consultés SUGIYAMA et collab. [2004], notés ou directement déduits des réponses aux questionnaires LI et KIM [2004]. Dans ce cas, l'objectif du moteur de recommandation est de trouver des items ayant le plus de descripteurs en commun avec ceux présents dans le profil de l'utilisateur (comme par exemple, pour les vêtements : la couleur, le type, etc., pour les films : le genre, la langue, l'année de production, etc.). Il existe d'autres méthodes de construction de profil utilisateur qui se démarquent davantage de la recherche d'information. Dans ce cadre, les recommandations sont calculées selon la probabilité qu'un utilisateur donné apprécie un item, ce qui revient à un problème de classification des items en plusieurs classes selon les degrés d'appréciation (binaire dans le cas de : « aime » et « n'aime pas », multi-classes dans le cas d'une notation, par exemple de 1 à 5).

Des mesures de similarités sont calculées après la construction des profils d'items et d'utilisateurs. Ces similarités sont calculées dans le but de comparer et de rechercher les items qui peuvent correspondre le plus au profil de l'utilisateur.

La recommandation basée sur le contenu est généralement appropriée dans le cadre de ressources de type textuel. Dans ce cas, des items peuvent être classés ensemble suite à leur similarité du point de vue de leurs attributs mais ils ont une qualité et une pertinence totalement différentes. Ces systèmes reposent sur un calcul de similarité donc, ils ne recommandent que des items similaires à ceux qu'un utilisateur donné a apprécié. Cela empêche l'apparition d'autres items qui, probablement, peuvent intéresser ce même utilisateur. C'est ce qui représente les limites des systèmes basés sur le contenu. Notons également, l'une des plus connues des limitations de tels systèmes est le problème de *démarrage à froid* où, un

1. Pandora Music Recommender System créé par Music Genome Project, www.pandora.com/

nouveau utilisateur ne pourra pas bénéficier de recommandations personnalisées puisque aucun profil n'est encore construit : il doit tout d'abord, soit faire quelques consultations ou fournir quelques appréciations soit remplir manuellement une fiche d'informations (formulaire) pour décrire son centre d'intérêt.

Une des façons de répondre à cette problématique est d'utiliser les commentaires. Dans LEVI et collab. [2012], les auteurs ont conçu un système de recommandation d'hôtels qui résout le problème de démarrage à froid. Leur système utilise les textes des commentaires des utilisateurs comme données principales. Ils ont défini des groupes de contextes en se basant sur les commentaires extraits des sites TripAdvisor.com et Venere.com. L'algorithme analyse et exploite les commentaires écrits et ceux jugés utiles par un nouvel utilisateur pour l'affilier au groupe de contexte correspondant.

3.2.2 Systèmes de recommandation à base de profils d'utilisateurs

On appelle aussi les systèmes de recommandation à base d'utilisateur, *systèmes de filtrage collaboratif*. Le principe de tels systèmes est à la base de la recommandation, les méthodes de filtrage par le contenu présentées dans la section précédente, sont plutôt liées aux systèmes de recherche d'informations dits personnalisés. Les systèmes de filtrage collaboratif sont des systèmes qui se basent sur les opinions et évaluations d'un groupe de personnes afin d'aider un individu particulier. L'idée est ici non plus de s'intéresser spécifiquement au nouvel item qui serait susceptible de plaire à l'utilisateur mais de regarder quels items ont apprécié les utilisateurs ayant des profils proches de l'utilisateur courant. Ce type de système utilise uniquement les informations contenues dans la matrice d'usages comme donnée d'entrée. Cette matrice est construite à partir des comportements des utilisateurs ou de leurs avis sur les items qu'ils connaissent déjà. De ce fait, on appelle systèmes à filtrage collaboratif *passif*, les systèmes de recommandation basés sur l'analyse de comportements des utilisateurs (les achats effectués, les pages visitées, etc.) et filtrage collaboratif *actif*, les systèmes basés sur des données déclarées par les utilisateurs (comme les notes).

Les approches de filtrage collaboratif ont l'avantage de se dispenser de la connaissance des items que l'on recommande. Autrement dit, il n'est pas nécessaire d'analyser le contenu des items. Le but est de calculer la proximité entre les utilisateurs. La supposition fondamentale des SFC est la suivante : si les utilisateurs X et Y notent n items similairement ou ont des comportements semblables comme par exemple : les mêmes achats, consultations, écoutes, etc., par conséquent ils vont noter ou réagir sur d'autres items de la même façon GOLDBERG et collab. [2001].

La matrice d'usage se compose des lignes qui correspondent aux utilisateurs et des colonnes qui correspondent aux différents items. Chaque cellule de la matrice correspond à une note fournie par l'utilisateur ou déduite à partir de son comportement. Le but des systèmes de filtrage collaboratif est de prédire la valeur de ces cellules qui représentent des items non-notés et probablement inconnus par les utilisateurs afin de recommander les items les mieux notés.

Les notes ou évaluations peuvent être des indications explicites, par exemple Amazon offre la possibilité de donner une note entre 1 et 5, ou des indications implicites comme les achats ou le nombre de clics. Par exemple, on peut convertir la liste des utilisateurs et les livres qu'ils aiment ou n'aiment pas vers une matrice d'usages (voir la table 3.1). Dans cette matrice, par exemple Michel est l'utilisateur actif à qui nous souhaitons recommander un(des) livre(s). Cependant nous devons prédire les valeurs manquantes pour lesquelles Michel n'a pas mentionné de préférences.

TABLEAU 3.1 – Un exemple de matrice d'usages

(a)

Adam :	☺	Saga « Harry Potter », Le Seigneur des Anneaux
	☹	Paris et Londres en 1789
Marie :	☺	Paris et Londres en 1789, L'Alchimiste
	☹	Le Seigneur des Anneaux
Michel :	☺	L'Alchimiste
	☹	Saga « Harry Potter »
Sara :	☺	L'Alchimiste
	☹	Le Seigneur des Anneaux

(b)

	Saga « Harry Potter »	Le Seigneur des Anneaux	Paris et Londres en 1789	L'Alchimiste
Adam	☺	☺	☹	
Marie		☹	☺	☺
Michel	☹			☺
Sara		☹		☺

Afin de remplir les cases vides de la matrice des usages, deux principales approches sont utilisées dans la littérature : *les approches basées sur la mémoire* (ou approches basées sur les plus proches voisins) et *les approches basées sur les modèles*. Il existe aussi des travaux où les auteurs ont hybridé ces deux approches LAI et collab. [2012].

3.2.2.1 Les approches basées sur la mémoire (*les plus proches voisins*)

Les SFC qui utilisent le voisinage se fondent sur l'avis de personnes partageant les mêmes idées et les mêmes intérêts pour donner une évaluation sur un item donné. Ces approches consistent à utiliser des algorithmes qui estiment les similarités entre les lignes (trouver les utilisateurs ayant le même comportement) ou entre colonnes (trouver les items qui ont été appréciés par le même public) de la matrice d'usages.

Soit Pred(u_l, i_j) la valeur de la case à prédire (la note), qui correspond à l'utilisateur u_l et à l'item i_j. Il faut choisir la mesure de similarité entre les lignes ou les colonnes, définir le nombre de *voisins* à prendre pour référence et finalement choisir la méthode de combinaison d'avis pour inférer une nouvelle évaluation.

— **Calcul des similarités** : Le choix de la mesure de similarité à utiliser dépend généralement de la nature de la matrice d'usages. Si cette dernière contient des données binaires comme dans l'exemple présenté dans la table 3.1 (« aime » et « n'aime pas »), la *distance de Jaccard* est l'une des mesures qui peuvent être utilisées (équation 3.1, S_a et S_b représentent l'ensemble des items notés par les utilisateurs u_a et u_b respectivement). Elle mesure le recouvrement entre les attributs de deux vecteurs de la matrice (verticaux ou horizontaux) sans tenir en compte des différences de notes entre ces deux vecteurs POIRIER [2011].

$$Sim_{Jaccard}(u_a, u_b) = \frac{|S_a \cap S_b|}{S_a \cup S_b} \tag{3.1}$$

Les cases de la matrice d'usages peuvent contenir des notes d'utilisateurs. Dans ce cas, les deux mesures les plus utilisées sont la *similarité Cosinus* et la *similarité de*

Pearson. La *similarité Cosinus* (équation 3.2).Elle est mesurée entre les deux vecteurs de notes des utilisateurs ($\overrightarrow{u_a}$, $\overrightarrow{u_b}$) et représente l'angle entre ces deux vecteurs.

$$cos(\overrightarrow{u_a}, \overrightarrow{u_b}) = \frac{\sum_{\{i \in S_a \cap S_b\}} n_{ai} * n_{bi}}{\sqrt{\sum_{\{i \in S_a\}} n_{ai}^2 \sum_{\{i \in S_b\}} n_{bi}^2}} \tag{3.2}$$

Où, n_{ai} et n_{bi} sont les notes attribuées à l'item i par les utilisateurs u_a et u_b respectivement.

Quant à la *similarité de Pearson* (équation 3.3) c'est une méthode qui calcule la corrélation statistique entre deux ensembles de notes d'utilisateurs pour déterminer leur similarité.

$$Sim_{Pearson}(a,b) = \frac{\sum_{\{i \in S_a \cap S_b\}} (n_{ai} - \overline{n_a}) * (n_{bi} - \overline{n_b})}{\sqrt{\sum_{\{i \in S_a \cap S_b\}} (n_{ai} - \overline{n_a})^2 \sum_{\{i \in S_a \cap S_b\}} (n_{bi} - \overline{n_b})^2}} \tag{3.3}$$

Où, $\overline{n_a}$ et $\overline{n_b}$ représentent les moyennes des notes dans les vecteurs a et b respectivement.

De ce fait, ces mesures permettent de construire deux types de matrices de similarité, *Utilisateur-Utilisateur* et *Item-Item* selon qu'on souhaite travailler sur les lignes ou les colonnes. Une fois la matrice des similarités construite, elle est utilisée dans la prédiction des notes (le calcul des recommandations).

— **Calcul des recommandations** :

— *L'approche basée sur les utilisateurs* où, on cherche les utilisateurs ayant les mêmes usages que l'utilisateur à qui l'on souhaite recommander des items. Les recommandations sont alors établies en fonction des notes d'utilisateurs similaires. On peut prédire la note Pred(u_l, i_j) de l'utilisateur u_l pour l'item i_j par la moyenne des notes de ses k voisins, mais le problème de cette méthode est qu'elle ne prend pas en compte les valeurs de similarité des utilisateurs voisins. Pour pallier ce problème, une solution très utilisée consiste à calculer la moyenne pondérée des évaluations des utilisateurs pour fournir une prédiction d'utilité Cornuéjols et Miclet [2011]) comme l'indique l'équation 3.4.

$$Pred(u_l, i_j) = \bar{X}_l + \frac{\sum_{v=1}^{k} sim(u_l, u_v)(X_v^j - \bar{X}_v)}{\sum_{v=1}^{k} |sim(u_l, u_v)|} \tag{3.4}$$

Où, $\bar{X}_l$ et $\bar{X}_v$ sont respectivement, les moyennes des notes attribuées par les utilisateurs u_l et u_v à tous les items.

— *L'approche basée sur les items* qui consiste à utiliser une mesure de similarité entre items pour déterminer les k items les plus similaires à l'item i_j pour lequel on cherche à calculer sa note $Pred(u_l, i_j)$ Linden et collab. [2003]; Sarwar et collab. [2001]. Amazon Pu et Faltings [2013] a mis en avant cette approche avec un système construisant une matrice de relations entre les items en se basant sur les achats des utilisateurs. Pour la prédiction des recommandations dans cette approche, on calcule la moyenne des notes de chaque utilisateur comme présenté dans l'équation 3.5

$$Pred(u_l, i_j) = \bar{X}^i + \frac{\sum_{v=1}^{k} sim(i_j, i_v)(X_l^v - \bar{X}^v)}{\sum_{v=1}^{k} |sim(i_j, i_v)|} \tag{3.5}$$

Où, $\bar{X}^i$ et $\bar{X}^v$ sont respectivement, les moyennes des notes reçues pour les items i_j et i_v par tous les utilisateurs.

3.2.2.2 Les approches basées sur un modèle d'apprentissage

Dans ces approches, les méthodes de l'apprentissage automatique sont utilisées, nous citons comme exemple, les modèles bayésiens CHIEN et GEORGE [1999]; MIYAHARA et PAZZANI [2000], les méthodes de partitionnement CHEE et collab. [2001]; CONNOR et HERLOCKER [2001]; SARWAR et collab. [2002]; UNGAR et FOSTER [1998] et les méthodes basées sur la régression VUCETIC et OBRADOVIC [2005]. Elles répondent à la problématique de la complexité de l'approche basée sur la mémoire en utilisant des modèles d'utilisateur, de ressource ou de communauté. En revanche elles ont un coût de conception et de fonctionnement plus important dû à l'étape de construction des modèles SU et KHOSHGOFTAAR [2009].

Les systèmes de filtrage collaboratif basés sur des modèles tentent de fournir des résultats plus précis que les systèmes basés sur l'approche mémoire (voisinage). Cependant, la grande partie des travaux de recherche et des systèmes commerciaux (par exemple, Amazon [2] LINDEN et collab. [2003], TiVo [3] ALI et VAN STAM [2004] et Netflix [4] BARBIERI et collab. [2010] sont basés sur le voisinage. Actuellement, il existe beaucoup plus de systèmes de recommandation basés sur le voisinage, car ils sont considérés comme plus faciles et intuitifs à manipuler HAMZAOUI [2014].

Les systèmes de recommandation qui se basent sur les utilisateurs (filtrage collaboratif) présentent quelques inconvénients. Nous en citons quelques uns dans ce qui suit :

— **Manque de données** : Dans les applications de recommandation où l'on a des notations explicites, le pourcentage moyen de ressources notées par les utilisateurs est très bas. Dans BONNIN [2010], l'auteur a donné l'exemple de la base de données MovieLens [5] qui contient 100 000 notes pour 1 642 films par 943 utilisateur, ce qui fait 6,3% de notes fournies. Dans un tel cadre, la similarité entre deux utilisateurs ne peut être calculée que s'ils ont noté des ressources communes en raison du manque d'informations entre les utilisateurs.

— **Démarrage à froid** : On note trois cas de démarrage à froid : un système qui débute où la matrice d'usages est vide, un nouvel utilisateur qui veut avoir des recommandations VOLINSKY [2009] et un nouvel item qui n'a pas encore d'avis pour le comparer à d'autres items SARWAR et collab. [2000].

Pour répondre à ces problématiques, les systèmes de recommandation hybrides sont apparus. Nous les présentons dans la section suivante.

3.2.3 Systèmes de recommandation hybrides

Les différentes limites présentées précédemment des deux autres types de système de recommandation ont conduit à proposer des systèmes *hybrides*. Ces derniers combinent plusieurs approches de recommandation BURKE [2002]; GUNAWARDANA et MEEK [2009]

2. www.amazon.com
3. www.tivo.com
4. www.netflix.com
5. http://grouplens.org/datasets/

pour prédire une note. La problématique majeure des systèmes hybrides réside dans la combinaison des différentes approches, nous citons quelques méthodes de combinaison qui ont été résumées dans BURKE [2007] comme suit :

— **Weighted** : consiste à interpoler les scores des différentes recommandations ;

— **Switching** : déterminer la technique de recommandation la plus appropriée au cas par cas ;

— **Mixed** : mélanger les recommandations issues des différentes techniques dans une seule liste finale ;

— **Feature Combination** : combiner les attributs provenant de techniques différentes ;

— **Meat-level** : utiliser une première technique pour construire un modèle qui sera utilisé par la deuxième technique.

L'hybridation peut être appliquée au niveau des notes ou bien au niveau de la fonction de prédiction. Plusieurs travaux ont montré l'efficacité des systèmes de recommandation hybrides comme le système proposé par BELL et collab. [2007]. Ils mélangent pas moins de 107 modèles qui sont en réalité des variantes très proches les unes des autres de 5 modèles de base. Ce système a obtenu le meilleur résultat lors du concours Netflix.

LI et KIM [2003] se sont également intéressé à la problématique d'hybridation en proposant une approche fondée sur la construction de classes. L'avantage de leur approche est qu'elle résout le problème de démarrage à froid précédemment évoqué. Le principe de l'approche se décompose de plusieurs étapes en commençant par une phase de partitionnement afin de grouper les documents proposés aux utilisateurs ensuite les auteurs calculent la distance entre les différents groupes précédemment construits mais également entre les documents et les groupes (mesures utilisées : cosinus améliorée et Pearson). À la fin, les recommandations sont proposées à l'utilisateur sur la base des items proches de son voisinage.

3.3 Recommandation sur des contenus liés (structurés en graphe)

De nombreuses applications reposent sur l'utilisation de graphe d'items (contenus liés) ou d'utilisateurs (communauté) afin de proposer des recommandations les plus pertinentes possible. Dans cette section, nous présentons une vue générale sur les graphes et leurs applications sur les systèmes de recommandation.

La notion de graphe est devenue aujourd'hui majeure dans de nombreux domaines. Nous présentons dans ce qui suit un état de l'art sur l'utilisation des structures de graphe dans le cas de la recommandation.

3.3.1 Recommandation basée sur les graphes

De nombreuses applications Web traitent des données possédant une forte structure de communautés, d'où la modélisation en graphes est la plus adéquate afin de proposer aux utilisateurs des contenus personnalisés. L'exemple le plus représentatif de ces applications est Facebook qui recommande aux utilisateurs de nouveaux "amis" en se basant sur leurs distances avec les autres utilisateurs.

Plusieurs algorithmes de recommandation s'appuient sur cette notion de distance pour représenter l'influence que les utilisateurs ou les items exercent les uns sur les autres. Par exemple le filtrage collaboratif par *Horting* AGGARWAL et collab. [1999] est une approche

basée sur un graphe de relations de similarité (représentées par des arcs) entre les utilisateurs (les nœuds du graphe). La notion d'influence qui correspond aux arcs se décline sous forme de deux contraintes : la contrainte de *Horting* impose de ne considérer que les utilisateurs ayant un grand nombre d'évaluations communes ; et la contrainte de prédictibilité ajoute à la notion de *Horting* une information sur le degré de ressemblance entre deux utilisateurs en se basant sur la distance de *Manhattan* CRAW [2010]. Le parcours du graphe permet de filtrer les utilisateurs proches et ceux ayant une expérience importante. La notion de prédictibilité est plus contraignante que le concept de proximité car le système a besoin d'un échantillon suffisamment important de ressources communément mesurées.

SCHWARTZ et WOOD [1993] ont décrit l'utilisation de la théorie des graphes telle que les cliques, les composants connexes et les algorithmes de parcours DIESTEL [2005]. Leur étude se focalise sur la modélisation des intérêts partagés par les utilisateurs dans le Web en utilisant les emails comme moyen de liaison de telle sorte si un utilisateur envoie un mail à un autre utilisateur pour un objectif précis, une liaison se crée entre ces deux utilisateurs. L'analyse de liens a été beaucoup utilisée, elle s'adapte selon le domaines d'application comme par exemple dans le domaine médical, KASTRIN et collab. [2014] utilisent le réseau sémantique MEDLINE pour lier des segments d'informations scientifiques.

Plusieurs approches se sont basées sur les graphes dans le but d'améliorer les recommandations. Par exemple, BENCHETTARA et collab. [2010] ont modélisé les données en un graphe bipartie intégrant l'ensemble des utilisateurs et items dans deux couches liées par des relations qui représentent les transactions des utilisateurs. Les auteurs ont considéré la prédiction des liens comme un problème d'apprentissage automatique et ils ont montré que la nature bipartie du graphe peut améliorer les modèles de prévision. Cette constatation est obtenue en projetant le graphe bipartie sur un graphe unimodal. De nouvelles variantes de mesures ont été introduites pour calculer la probabilité de connexion de deux nœuds donnés.

D'autres travaux ont utilisé la *marche aléatoire* (*Random Walk*) pour proposer des recommandations aux utilisateurs. Nous citons YIN et collab. [2010], qui ont proposé un système de recommandation qui utilise les attributs et les propriétés structurelles des items pour recommander des liens potentiels dans des réseaux sociaux. Ils ont également étudié différentes méthodes pour le calcul des poids des relations dans un réseau social et ont expérimenté sur deux ensemble de données réelles : DBLP [6] (Digital Bibliography Project) et IMDB [7] (The Internet Movie Database).

SAWANT [2013] a développé un système de recommandation personnalisé sur la collection « Yelp Dataset Challenge [8] » en utilisant le même algorithme de filtrage collaboratif basé sur le réseau d'inférence proposé par ZHOU et collab. [2007] et SHANG et collab. [2008]. Les données de Yelp ont été représentées par un graphe bipartie pondéré où les liens entre les utilisateurs et les ressources (des commerces ou des entreprises) sont pondérés par les votes des utilisateurs. L'auteur a considéré le problème de recommandation comme une projection dans le graphe. Plus spécifiquement, un processus d'allocation des ressources du réseau pour produire des mesures de similarité entre chaque paire d'utilisateurs et chaque paire d'entreprises/commerces qui sont ensuite utilisées pour prédire les votes et proposer des recommandations. A travers les expérimentations, l'auteur a montré que l'utilisation du graphe et la pondération des liens améliorent les performances de prédiction.

6. http://dblp.uni-trier.de/
7. http://www.imdb.com/
8. https://www.yelp.com/dataset_challenge

Nous avons introduit précédemment (section 3.2.2.2) un problème majeur de la recommandation personnalisée qui est le démarrage à froid. Plusieurs travaux ont eu recours à la modélisation des données en graphe pour remédier ce problème. CHEKKAI et collab. [2011] ont introduit le *Problème des Nœuds Critiques* dit aussi CNP (*Critical Node Problem*) pour pallier ce problème de démarrage à froid d'un nouvel utilisateur. L'objectif du CNP est de trouver un ensemble de k-nœuds dans un graphe, dont leur suppression donne une fragmentation maximale du graphe ASHWIN et collab. [2007]; SHIVASHANKAR et collab. [2012]. L'ensemble des k-nœuds critiques peut représenter dans un système de recommandation collaboratif l'ensemble des usagers permettant de déléguer efficacement leur communauté et faire adapter un nouveau nœud représentant un nouvel utilisateur tout en le mettant en contact avec des nœuds lui semblant importants.

3.3.1.1 Représentation en graphe

La plupart des systèmes de recommandation basés sur les graphes font face à deux étapes avant de procéder à la recommandation qui sont : (i) la modélisation en graphe (que représentent les nœuds ?) et (ii) les structures/opérations qui sont exploitées/menées dans le graphe. Un des exemples les plus connus de système qui exploite à la fois des liens et le contenu textuel des documents est le moteur de recherche Google. Ce dernier n'est pas un système de recommandation personnalisé puisque l'utilisateur doit taper manuellement son besoin en information sous forme de requêtes. Le facteur majeur de ré-ordonnancement des pages recommandées pour une requête donnée est basé sur les relations d'hyperlien entre les pages BRIN et PAGE [1998]. Il s'agit de la méthode de ré-ordonnancement appelée *PageRank*. Cette méthode détermine l'ordre des pages en associant à chaque page p sa valeur de *PageRank*. Plus précisément, le PageRank d'une page p est la probabilité de visiter p dans une marche aléatoire dans le Web. Le *PageRank* de p est défini comme suit :

$$\mathrm{PR}(p) = (1-d) + d(\mathrm{PR}(\mathrm{T}_1)/\mathrm{C}(\mathrm{T}_1) + ... + \mathrm{PR}(\mathrm{T}_n)/\mathrm{C}(\mathrm{T}_n))$$

Où, $\mathrm{T}_1 ... \mathrm{T}_n$ sont les pages qui citent p, d est le coefficient d'amortissement (0.85) et $\mathrm{C}(\mathrm{T}_i)$ est défini comme le nombre de liens émis par la page p.

3.3.2 Systèmes de recommandation à base de réseaux de citations

Les bibliothèques numériques ont pour objectif d'aider les utilisateurs à acquérir des documents et des connaissances d'une manière rapide et efficace. Des techniques capables de trouver les documents appropriés en fonction des demandes réelles sont nécessaires. Certains chercheurs ont proposé différents systèmes de recommandation et des moteurs de recherche pour des bibliothèques numériques BOLLACKER et collab. [2000]; CHEN et collab. [2006]; HUANG et collab. [2002a]; RATPRASARTPORN et collab. [2009]. Nous avons déjà parlé des systèmes de recommandation commerciaux comme Amazon qui adoptent la technique de filtrage collaboratif combinant à la fois l'historique des achats, les évaluations des clients, leurs commentaires et les caractéristiques des produits pour proposer les recommandations. Cependant, ces techniques de recommandation ne sont pas appropriées pour accomplir la tâche de recommandation dans une bibliothèque numérique qui n'intègre pas les votes des utilisateurs et le concept de profilage.

Dû au nombre impressionnant de publications scientifiques, il est impossible d'étudier manuellement tous les documents dans un domaine donné. Les systèmes de recommandation

d'articles scientifiques se sont développés pour identifier les informations ou les connaissances pertinentes et qui répondent aux exigences des utilisateurs. Par exemple, lorsqu'on commence à travailler sur un nouveau problème de recherche, les chercheurs souhaitent généralement, comprendre rapidement l'état de l'art correspondant à ce problème ce qui revient à trouver les documents les plus pertinents.

3.3.2.1 Analayse des relations de citations

De nombreux travaux se sont focalisés sur la tâche des réseaux de citations, notamment à la classification des types de citations HUANG et QIU [2010]; NANBA et OKUMURA [1999]; TANG et collab. [2009], l'importance et l'influence entre les différentes citations HUANG et QIU [2010]; TANG et collab. [2009]. TANG et collab. [2009] ont extrait le contexte des citations (les 50 mots qui précèdent et qui suivent la citation) pour classer les relations de citations en plusieurs catégories ("drill down", "similar"et "other") en utilisant des caractéristiques linguistiques et les similarités des topics entre les articles citant et cité. Ils ont étudié la force d'influence (i.e., "strong", "middle"et "weak") des relations des citations entre elles en supposant que si deux articles qui se citent décrivent un contenu similaire, l'article cité peut avoir une forte influence sur l'article citant. Cependant, considérer la similitude du contenu seule pour évaluer le niveau d'influence peut biaiser les résultats car il existe des articles d'une forte influence mais peuvent beaucoup varier dans le contenu.

HUANG et QIU [2010] ont proposé un réseau de liens sémantiques de citation (*Citation Semantic Link Network, C-SLN*) pour décrire l'information sémantique dans le réseau de citations. Ils ont intégré plusieurs méthodes de traitement automatique du langage naturel pour la construction du C-SLN et calculé l'importance des références. Ils ont supposé qu'une référence apparaissant à plusieurs fois dans la partie principale d'un article, doit avoir une grande importance. Cependant, l'extraction des références avec leurs occurrences et positions peut être une tâche fastidieuse.

3.3.2.2 Les relations entre les documents scientifiques dans un graphe de citations

Pour utiliser les informations liées aux citations, généralement, il y a deux façons différentes basées sur (i) le voisinage (citation directe) et (ii) la structure globale du graphe de citations i.e. nombre de chemins entre les nœuds et la longueur des chemins LIANG et collab. [2011]. Il existe deux méthodes qui utilisent les références et citations directes sont :

— *Le couplage bibliographique*, appelé également *co-couplage*, proposé par KESSLER [1963]. dans cette méthode, la similarité entre deux articles est basée sur leur nombre de co-références. Il a supposé que si deux articles ont des références communes, ils ont probablement un même sujet. Cette méthode est considérée comme un outil d'aide à la recherche de documents de telle sorte pour un article a_i, on peut retrouver tous les articles qui partagent des co-références avec a_i. Cependant, le co-couplage a des limites, deux articles peuvent citer la même référence alors qu'ils n'ont pas de sujet commun.

— *La co-citation*, proposé par MARSHAKOVA [1973] et SMALL [1973]. Dans cette méthode la similarité entre deux articles est basée sur le nombre de fois où ils sont co-cités. Deux articles sont co-cités s'ils apparaissent ensemble dans la bibliographie d'un autre article. Contrairement à la méthode du co-couplage, la similarité entre deux articles n'est pas fixée parce qu'avec le temps, ils peuvent recevoir de plus en plus de co-citations. Cependant, cette méthode ne peut pas être appliquée immédiatement après

que deux articles sont récemment publiés. De plus, comme avec la méthode du co-couplage, le fait que deux articles sont co-cités ne signifie pas qu'ils partagent le même sujet.

3.3.2.3 La recommandation des documents scientifiques

LIANG et collab. [2011] ont développé un système de recommandation qui propose des articles scientifiques pertinents par rapport à un article donné par l'utilisateur. Leur système se base sur un réseau de citations et procède en deux étapes. En premier, ils définissent une métrique appelée *Local Relation Strength* qui mesure la dépendance entre l'article cité et celui qui le cite. Ensuite, un modèle appelé *Global Relation Strength* est proposé pour capturer la valeur de pertinence entre deux articles dans le graphe de citations qui incorpore la distance dans le graphe et le pois des liens.

Le filtrage collaboratif est très populaire dans le domaine de la recommandation. MCNEE et collab. [2002] ont proposé deux algorithmes « Utilisateur-Item CF » et « Item-Item CF » en intégrant le filtrage collaboratif dans le domaine de recherche des documents scientifiques. Ils ont adopté plusieurs façons pour modéliser un graphe de citations en une matrice de votes (Utilisateur-Item). En particulier, il existe deux manières pour une telle modélisation LIANG et collab. [2011] :

— Un document représentera un utilisateur dans la matrice et une citation représentera un item et chaque document votera pour les documents présents dans ses références (les documents cités).

— Les utilisateurs et les items dans la matrice seront les documents et la note entre deux documents sera la valeur d'une mesure de co-citation (similarité lexicale, nombre de références communes, etc.).

3.4 Évaluation des systèmes de recommandation

Pour s'assurer de la capacité des systèmes de recommandation à satisfaire les besoins des utilisateurs, une étape d'évaluation est nécessaire. En revanche, il existe plusieurs mesures d'évaluation et le choix d'une de ces mesures dépend fortement du type de données à traiter et des intérêts des utilisateurs HERLOCKER et collab. [2004]. Pour l'évaluation des systèmes de recommandation, on utilise souvent des mesures d'évaluation comme le rappel et la précision.

Dans la littérature, l'évaluation des performances des systèmes de recommandation se limite généralement au calcul de précision HERLOCKER et collab. [2004].

La mesure la plus utilisée dans le domaine des systèmes de recommandation est la moyenne des carrés entre les notes prédites et les notes réelles ou *Root Mean Squared Error* (RMSE), sélectionnée notamment pour le Challenge *Netflix* BELL et KOREN [2007]; BENNETT et LANNING [2007]. Ce challenge a été proposé en 2006 par la société *Netflix*, dans le but d'améliorer les performances du moteur de recommandation utilisé sur son site. Ils ont mis à disposition de tous les candidats, une grosse quantité de données d'usages (480 000 clients, les informations de plus de 17 000 films et plus de 100 million de votes) BENNETT et LANNING [2007].

Dans le cadre de cette thèse, pour les systèmes de recommandation développés, nous avons utilisé les mesures d'évaluation qui sont le rappel, la précision et le nDCG@10. Nous avons déjà défini dans le chapitre précédent ces mesures (voir section 2.4.2). Dans cette section, nous donnons une brève définition des mesures RMSE et MAE.

— **Root Mean Squared Error (RMSE)** : La RMSE permet de mesurer l'erreur faite entre la note prédite et la note réelle donnée par l'utilisateur. Sa formule est la suivante :

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i}(p_{ui} - n_{ui})^2}{n}}$$

Où, n_{ui} représente la note réelle donnée par l'utilisateur u sur l'item i, p_{ui} représente la note prédite par le système de recommandation et n le nombre total de notes prédites.

La RMSE étant une mesure d'erreur, l'objectif est d'avoir une valeur la plus basse possible. Dans la première année du challenge *Netflix* (2006), la valeur la plus basse obtenue par le système *Cinematch* [9] était de 0,9525. Trois ans plus tard, Koren a réussi de baisser cette valeur de 10,09%, soit 0,8553 de RMSE avec son système *Belkor's pragmatic chaos* KOREN [2009] ce qu'il lui a permis de remporter le Challenge. Ce score a été obtenu en combinant plus de 250 méthodes de prédiction, certaines d'entre elles étant paramétrées. Cette faible amélioration du score RMSE donne une idée des limites probables de la recommandation automatique et montre aussi qu'il n'est pas facile d'améliorer les performances des systèmes de recommandation.

— **Mean Absolute Error (MAE)** : La MAE est également une mesure d'erreur qui permet de calculer la moyenne des erreurs absolues entre les notes prédites et les vraies notes. Une erreur absolue mesure l'imprécision entre une note prédite et la note réelle correspondante par la formule suivante :

$$e_{ui} = |p_{ui} - n_{ui}|$$

et la MAE se calcule comme suit :

$$\text{MAE} = \frac{\sum_{ui} e_{ui}}{n}$$

Où n représente le nombre total de notes prédites.

3.5 Conclusion

Les approches des systèmes de recommandation sont particulièrement variées, et se classent de différentes manières en fonction de plusieurs critères. Dans ce chapitre, nous avons présenté ces approches en les séparant en deux classes : les approches de recommandation classiques qui regroupent les approches basées sur le contenu, sur les profils utilisateurs (filtrage collaboratif) et l'approche hybride qui combine les deux précédentes approches. Dans la deuxième classe, nous avons défini les systèmes de recommandation basés sur la structure de graphe et plus précisément les systèmes de recommandation des documents scientifiques dans des bibliothèques numériques puisque nous nous focalisons sur ce type de systèmes dans la deuxième partie de ce manuscrit.

Dans la suite de la thèse, nous nous intéressons principalement aux systèmes de recommandation basés sur les graphes qui répondent à une stratégie basée sur les liens et les forces des votes des utilisateurs. Deux domaines d'application différents sont utilisés pour tester les

9. http://beelan515-yahoo-com.wikidot.com/netflix-cinematch , version du site : 09/05/2016

méthodes proposées : la recommandation des livres d'Amazon (la compagne d'évaluation INEX Social Book Search/CLEF) et la plateforme Revues.org d'OpenEdition.

Dans le prochain chapitre, nous présentons une étude faite sur les comptes rendus de lecture dans le domaine des sciences humaines et sociales et les méthodes utilisées pour les détecter. Ce type de document est porteur d'opinion et représente l'avis des lecteurs experts sur des ouvrages scientifiques ce qui peut être un facteur puissant pour la recommandation des documents scientifiques qui ne possèdent pas de notes (votes) contrairement à ce qu'on trouve dans des sites comme Amazon.

Chapitre 4

Détection automatique des comptes rendus de lecture

Résumé : Dans ce chapitre, nous nous intéressons à la détection de textes critiques porteurs d'opinions. Plus précisément, nous cherchons à recueillir automatiquement un ensemble de comptes rendus de lecture de livres au sein de plateformes Web afin de les exploiter ultérieurement dans le cadre d'une recherche "sociale" de livres. Nous montrons qu'une telle classification en genres bénéficie d'approches statistiques simples, au delà des sacs de mots.

Sommaire

4.1 Introduction

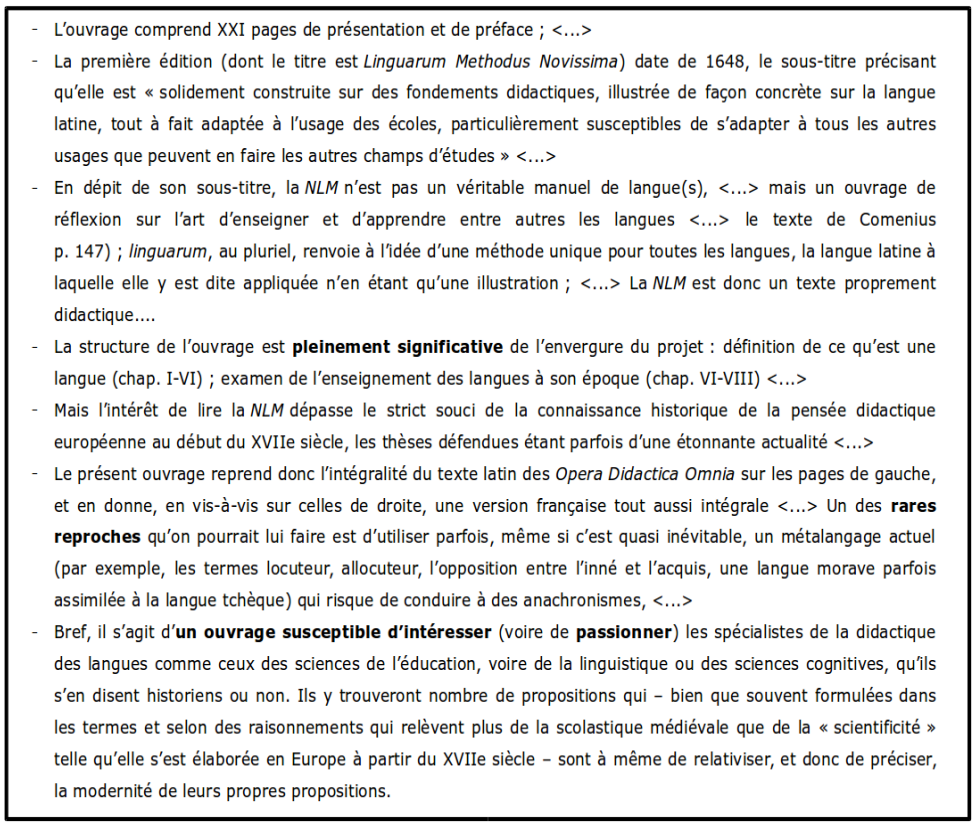
La recommandation automatique des livres est devenue une tâche très importante avec la croissance des ressources électroniques dans le Web. Elle nécessite le traitement du langage naturel et la recherche d'information. La recommandation peut être élaborée par la combinaison de l'analyse de contenu et le filtrage collaboratif. Ce dernier consiste à exploiter le comportement de l'utilisateur (navigation, clics, achats, etc.) ou à analyser les commentaires et les critiques sur les réseaux sociaux tels que LibraryThink [1] ou des librairies en ligne tel que Amazon Book [2], comme il a été proposé pour la tâche "Social Book Search "dans la compagne d'évaluation INEX KAZAI et collab. [2010]. Contrairement aux études antérieures sur la recommandation des livres, nous traitons spécialement les contenus scientifiques dans le domaine des sciences humaines et sociales. Nous nous intéressons à la collecte automatique d'un corpus de critiques appelés ``comptes rendus de lecture ´´. Ce type de critiques, contrairement à ceux que l'on trouve dans les réseaux sociaux et dans les forums, sont écrits par des experts dans des revues scientifiques. Les comptes rendus de lecture expriment deux points de vue complémentaires sur un livre donné : une analyse approfondie d'un coté et l'opinion de l'auteur sur un/plusieurs aspect(s) de ce livre. La figure 4.1 illustre un exemple d'un compte rendu de lecture de l'ouvrage intitulé *Novissima Linguarum Methodus / La toute nouvelle méthode des langues* de l'auteur *Jan Amos Comenius*. Dans ce compte rendu, l'auteur décrit les différentes parties qui composent le livre critiqué en exprimant son opinion avec des mots qui reflètent un avis positif (mots en gras dans la figure). Ce type de mot est difficilement détectable automatiquement, notamment avec des outils d'analyse d'opinion. De ce fait, notre contribution consiste à exploiter des méthodes de classification en genre pour détecter automatiquement ce type de document. La principale problématique est de trouver une méthode de représentation avec les caractéristiques adéquates pour procéder ensuite à la classification.

Ces dernières années, beaucoup de chercheurs (voir chapitre 3) ont exploité les commentaires des utilisateurs dans plusieurs sources (Amazon, Facebook, Twitter, etc.) pour l'analyse d'opinion mais à notre connaissance, notre contribution s'avère la première à détecter et collecter automatiquement des critiques longues scientifiques et à les exploiter dans un besoin de recommandation de livres.

Recommander des livres dans le domaine des sciences humaines et sociales repose sur plusieurs facteurs. Les avis des lecteurs sur des ouvrages ou des articles ont une grande importance dans la communauté scientifique. Dans notre contribution, nous nous sommes intéressés aux comptes rendus de lecture qui peuvent améliorer les résultats de recommandation. Un moteur de recherche classique ne propose pas de rechercher des comptes rendus de lecture d'un livre donné autrement qu'avec une requête composée du titre du livre et des mots clés tels que, *compte rendu* ou *review*. De plus, les moteurs de recherche n'intègrent pas un système de filtrage des genres de documents, une classification des résultats de la recherche doit être effectuée manuellement et sa réalisation est donc coûteuse en terme de temps. En effet, chaque document (ou une partie) doit être manuellement lu pour attribuer une catégorie adaptée (genre tel qu'un chapitre d'ouvrage, note de lecture, annonce, etc.). Il est donc important de passer par un processus de classification automatique pour filtrer les documents.

Nous présentons dans ce chapitre une approche d'identification des comptes rendus de

1. http ://www.librarything.com
2. https://www.amazon.com/books-used-books-textbooks/b?ie=UTF8&node=283155



- L'ouvrage comprend XXI pages de présentation et de préface ; <...>
- La première édition (dont le titre est *Linguarum Methodus Novissima*) date de 1648, le sous-titre précisant qu'elle est « solidement construite sur des fondements didactiques, illustrée de façon concrète sur la langue latine, tout à fait adaptée à l'usage des écoles, particulièrement susceptibles de s'adapter à tous les autres usages que peuvent en faire les autres champs d'études » <...>
- En dépit de son sous-titre, la *NLM* n'est pas un véritable manuel de langue(s), <...> mais un ouvrage de réflexion sur l'art d'enseigner et d'apprendre entre autres les langues <...> le texte de Comenius p. 147) ; *linguarum*, au pluriel, renvoie à l'idée d'une méthode unique pour toutes les langues, la langue latine à laquelle elle y est dite appliquée n'en étant qu'une illustration ; <...> La *NLM* est donc un texte proprement didactique....
- La structure de l'ouvrage est **pleinement significative** de l'envergure du projet : définition de ce qu'est une langue (chap. I-VI) ; examen de l'enseignement des langues à son époque (chap. VI-VIII) <...>
- Mais l'intérêt de lire la *NLM* dépasse le strict souci de la connaissance historique de la pensée didactique européenne au début du XVIIe siècle, les thèses défendues étant parfois d'une étonnante actualité <...>
- Le présent ouvrage reprend donc l'intégralité du texte latin des *Opera Didactica Omnia* sur les pages de gauche, et en donne, en vis-à-vis sur celles de droite, une version française tout aussi intégrale <...> Un des **rares reproches** qu'on pourrait lui faire est d'utiliser parfois, même si c'est quasi inévitable, un métalangage actuel (par exemple, les termes locuteur, allocuteur, l'opposition entre l'inné et l'acquis, une langue morave parfois assimilée à la langue tchèque) qui risque de conduire à des anachronismes, <...>
- Bref, il s'agit d'**un ouvrage susceptible d'intéresser** (voire de **passionner**) les spécialistes de la didactique des langues comme ceux des sciences de l'éducation, voire de la linguistique ou des sciences cognitives, qu'ils s'en disent historiens ou non. Ils y trouveront nombre de propositions qui – bien que souvent formulées dans les termes et selon des raisonnements qui relèvent plus de la scolastique médiévale que de la « scientificité » telle qu'elle s'est élaborée en Europe à partir du XVIIe siècle – sont à même de relativiser, et donc de préciser, la modernité de leurs propres propositions.

FIGURE 4.1 – Extrait d'un compte rendu de lecture (publié dans la revue *Documents pour l'histoire du français langue étrangère ou seconde*, URL :http://dhfles.revues.org/1131)

lecture pour construire un corpus de critiques pour une future utilisation dans le processus de recommandation. Nous montrons ici que les méthodes généralement utilisées pour une classification thématique, donnent des résultats très satisfaisants pour une classification en genre. Nous supposons avoir deux genres : Compte Rendu de lecture, noté $Review$ et le reste, noté $\overline{Review}$ qui désigne tout document qui n'est pas un compte rendu de lecture.

Après avoir fait le point sur l'état de l'art sur les techniques de classification en genre en section 5.2, nous décrivons, dans la section 5.3, le portail OpenEdition qui est la source des livres et des comptes rendus de lecture considérés dans cette étude. Dans la section 5.4, nous définissons le corpus manuellement construit à partir de la plateforme revues.org du portail OpenEdition. Nous donnerons les schémas d'indexation utilisés pour le processus de classification dans la section 5.5. Nous présentons ensuite, les expérimentations faites pour la détection et la construction automatique d'un corpus de comptes rendus de lecture.

4.2 État de l'art

La classification automatique de texte est une tâche standard dans plusieurs domaines notamment les systèmes de recherche d'information et le traitement automatique des langues. La classification la plus connue est la classification par thématique JOACHIMS [1998]; SEBASTIANI et RICERCHE [2002], mais d'autres types de classification ont été développés comme pour les sentiments BO PANG [2002], la détection des auteurs STAMATATOS et collab. [2000]; DE VEL et collab. [2001] et aussi la personnalité de l'auteur OBERLANDER et NOWSON [2006].

Le genre est une autre caractéristique d'un texte. Il a été démontré par BIBER [1988] et d'autres auteurs après lui, que le genre du texte affecte ses propriétés formelles [3]. Toutefois, il est possible d'utiliser des indices (lexicaux, syntaxiques ou encore structurels) à partir du texte comme des caractéristiques pour prédire son genre qui peut par la suite aider dans une application de recherche d'information FINN et KUSHMERICK [2006]; FREUND et collab. [2006]; KARLGREN et CUTTING [1994]; KESSLER et collab. [1997]. Par exemple, une recherche dans le web sur le thème "oiseau"peut retourner une entrée vers une encyclopédie, une fiche d'information biologique, un reportage sur l'élevage des oiseaux, un blog posté sur les différents types d'oiseaux ou d'autres documents de natures différentes. L'utilisateur peut rejeter plusieurs d'entre eux seulement à cause de leur genre qui ne correspond pas au type ou à la qualité de l'information recherchée.

Les premiers travaux sur la classification automatique en genre ont été réalisés par Karlgren et Cutting KARLGREN et CUTTING [1994] où ils ont utilisé une combinaison d'indices structurels (ex. le nombre de noms), d'indices lexicaux (ex. le nombre de "it ") et d'autres indices sur les termes fréquents dans les phrases du texte. Par contre KESSLER et collab. [1997] ont évité toute caractéristique structurelle puisque, en général, cela nécessite un processus d'annotation ou d'analyse préalable. Ils ont utilisé les indices de ponctuation (ex. le nombre de ".", de ",", etc.) et d'autres indices qui sont spécifiques aux textes en anglais. Ces approches sont généralement liées au vocabulaire du domaine, en conséquence leur adaptation pour d'autres domaines émergents demande une phase de réapprentissage sur un corpus conséquent.

Certaines approches proposent des propriétés indépendantes de la langue telles que le nombre de caractères et le nombre de phrases par document, la moyenne du nombre de caractères et de mots par phrase, etc. Ces approches s'avèrent efficaces pour la classification de genres journalistiques PETRENZ et WEBBER [2011]. Plusieurs travaux combinent propriétés stylométriques (longueur de phrases, signes de ponctuation), catégories lexicales et mots-clés pour la classification des genres littéraires. D'HONDT [2014] a identifié dans les documents des indices stylistiques, syntaxiques, et les éléments du texte appartenant à un lexique d'opinions. Ces indices ont ensuite été utilisés pour construire un modèle par apprentissage au moyen d'un perceptron, en limitant le nombre de prédictions à 1 à 5 catégories par documents traité. LECLUZE et LEJEUNE [2014] ont considéré que le style littéraire de nouvelles détermine la catégorie littéraire d'appartenance. Les auteurs ont également pris en compte les éléments du texte appartenant à divers champs lexicaux, constatant un bénéfice dans la classification.

Pour mettre en œuvre des méthodes de classification il faut faire un choix d'un mode de représentation des documents SEBASTIANI et RICERCHE [2002]. Ensuite, il est nécessaire de choisir un algorithme de classification. Dans ce qui suit, nous présentons quelques schémas de représentation de documents que nous avons utilisé dans nos expérimentations et également quelques modèles d'apprentissage.

4.2.1 Processus de classification

Le processus de classification regroupe deux étapes. L'étape de représentation des données et la classification de documents.

3. Une propriété formelle est un matériau linguistique utilisé dans un texte (les pronoms interrogatifs, mode de verbe, etc.)

4.2.1.1 Représentation vectorielle des documents textuels

L'exploitation et le traitement automatique des documents, en particulier pour les tâches de classification, nécessitent une première étape consistant à les représenter. Pour cela, la méthode la plus courante consiste à projeter les données textuelles (par exemple les mots) dans un espace vectoriel. De nombreux travaux de classification en genres utilisent une telle approche. Citons par exemple VINOT et collab. [2003] qui représentent les données à l'aide d'un modèle vectoriel pour une tâche de détection de documents à contenus racistes. MEMMI [2000] et POULIQUEN [2002] utilisent quant à eux un modèle vectoriel pour calculer des scores de similarité entre différents documents. Les travaux de PISETTA et collab. [2006] emploient une technique d'analyse linguistique (étudier les termes très fréquents, regrouper ceux sémantiquement proches à l'aide de WordNet [4]) pour extraire un ensemble de termes candidats qui permettront de construire des concepts relatifs au corpus étudié. Ensuite, ils utilisent un modèle vectoriel pour représenter les documents par un vecteur de concepts.

Le modèle vectoriel le plus couramment utilisé dans la littérature pour la tâche de classification PISETTA et collab. [2006] est celui de SALTON et collab. [1975]. Le modèle de Salton consiste à représenter un corpus par une matrice telle que les lignes soient relatives aux descripteurs et les colonnes aux documents. Une cellule d'une telle matrice comptabilise la fréquence d'apparition d'un descripteur dans un document. Ainsi, la matrice formée peut être utilisée pour effectuer diverses tâches automatiques de fouille de textes.

Dans ce cas, les vecteurs caractéristiques adoptés contiennent les poids des différents mots qui apparaissent au moins une fois dans le corpus. Ainsi chaque document sera présenté comme suit :

$$\overrightarrow{d_j} = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{ij} \\ \vdots \\ a_{mj} \end{pmatrix}$$

Où a_{aij} représente le poids de terme m_i dans le document d_j. Ce poids permet de refléter l'importance du terme m_i dans le document traité d_j et se calcule de diverses manières. Dans ce qui suit (section 4.2.1.2), nous décrivons brièvement la méthode de calcul de poids TF-IDF des vecteurs descripteurs que nous avons utilisée dans nos expérimentations ainsi que quelques algorithmes de sélection de caractéristiques.

4.2.1.2 Phase de classification

Le principe d'une classification automatique de textes est d'utiliser un modèle afin de classer un document dans une catégorie pertinente. On distingue deux types de modèles de classification : ceux nécessitant une phase d'apprentissage et ceux sans phase d'apprentissage. Parmi les modèles utilisant un apprentissage, on distingue l'apprentissage supervisé et non supervisé. Nous effectuons dans nos travaux uniquement de la classification avec apprentissage supervisé.

La notion d'**apprentissage** introduit le fait d'apprendre un ensemble de relations entre les critères caractérisant l'élément à classer et sa classe cible. Les algorithmes de classification

4. https ://wordnet.princeton.edu/

avec apprentissage ont recours à un ensemble d'exemples afin d'apprendre ces relations. La notion de **supervisé** signifie que les exemples sont étiquetés (la classe est connue).

Il existe de nombreuses méthodes de classification avec apprentissage supervisé. Nous présentons les algorithmes suivants qui sont les plus utilisés dans la littérature :

4.2.1.2.1 Naïf Bayes Le classifieur Bayésien Naïf est un classifieur probabiliste simple appliquant le théorème de BAYES [1763]. Considérons $v_j = (v_{j1}, ..., v_{jk}, ..., v_{jd})$ un vecteur de variables aléatoire représentant un document d_j et C un ensemble de classes. En s'appuyant sur le théorème de Bayes, la probabilité que ce dernier appartienne à la classe $c_i \in$ C est définie par la formule suivante :

$$\mathrm{P}(c_i|v_j) = \frac{p(v_j|c_j).p(c_i)}{p(v_j)} \tag{4.1}$$

La variable aléatoire v_{jk} du vecteur v_j représente l'occurrence de l'unité linguistique k retenue pour la classification dans le document d_j. La classe c_k d'appartenance de la représentation vectorielle v_j d'un document d_j est définie comme suit :

$$c_k = arg\ max \mathrm{P}(c_i \in \mathrm{C}) \prod_{i=1}^{n} p(w_i|\mathrm{C}) \tag{4.2}$$

où, w_i est un mot dans le document d_i, n représente le nombre total de mots dans d_i et C est l'ensemble de classes d'appartenance. Le classifieur Bayésien Naïf affecte au document d_j la classe ayant obtenue la probabilité d'appartenance la plus élevée. La probabilité "a priori" $p(c_i)$ est définie de la façon suivante :

$$\mathrm{P}(c_i) = \frac{nombre\ de\ documents\ \in c_i}{nombre\ total\ de\ documents} \tag{4.3}$$

En faisant l'hypothèse que les v_j sont indépendantes, la probabilité conditionnelle $\mathrm{P}(v_j|c_i)$ est définie ainsi :

$$\mathrm{P}(v_j|c_i) = \prod_{k} \mathrm{P}(v_{jk}|c_i) \tag{4.4}$$

Un des avantages du classifieur Bayésien Naïf réside dans sa capacité d'estimation des paramètres avec peu de données d'apprentissage CHARTON et collab. [2008].

4.2.1.2.2 Support Vector Machines (SVM) Les Machines à Vecteurs de Supports sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression VAPNIK [1995]. La première idée clé est la notion de marge maximale, i.e. la distance entre la frontière de séparation et les échantillons les plus

proches est maximisée BURGES [1998]; ESSEGHIR et collab. [2010]; GUYON et collab. [2002]; VAPNIK [1995]. L'autre idée maîtresse des SVM est de transformer, grâce à une fonction noyau, l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe un séparateur linéaire.

Formellement, considérons l'ensemble d'apprentissage suivant composé de n exemples et p attributs :

$$D = (x_i, c_i) | x_i \in R^p, c_i \in -1, +1, i = 1, ..., n \tag{4.5}$$

Où, c_i est soit +1 ou -1, indiquant la classe à laquelle le vecteur réel p-dimensionnel x_i appartient.

N'importe quel hyperplan qui peut diviser les points ayant $c_i = +1$, à partir de ceux ayant $c_i = -1$, peut être écrit comme un ensemble de points qui vérifient la condition suivante :

$$w.x - b = 0 \tag{4.6}$$

Où, w est normal à l'hyperplan, $|b|/||w||$ est la distance perpendiculaire de l'origine à l'hyperplan. $||w||$ est la norme euclidienne de w et b est une constante.

La recherche de la marge optimale permettant de déterminer les paramètres w et b de l'hyperplan conduit à un problème d'optimisation quadratique qui consiste (dans le cadre général) à minimiser :

$$||w||^2 + C \sum \varepsilon_i | y_i (w.\phi(x_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0 \tag{4.7}$$

Où C est un paramètre de compromis entre la marge et les erreurs [5], ε_i est une variable ressort associée à l'observation x_i, et ϕ est une transformation. Le problème peut être résolu par la méthode Lagrangienne d'optimisation quadratique avec contraintes (formulation duale) pour maximiser la marge VAPNIK [1995].

$$\sum \alpha_i (1/2 \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j) | 0 \leq \alpha_i \leq C, \sum \alpha_i y_i = 0) \tag{4.8}$$

Où, α_i est le multiplicateur Lagrangien associé aux vecteurs x_i. Si la valeur de α_i est non-nulle alors x_i est un vecteur de support et $K(x_i, x_j) = \phi(x_i).\phi(x_j)$ est le noyau de transformation. Le noyau d'un SVM est une fonction symétrique définie-positive qui permet de projeter les données dans un espace transformé de grande dimension dans lequel s'opère plus facilement la séparation des classes.

La décision est obtenue selon (le signe de) la fonction :

$$f(x) = sign[\alpha_i y_i k(x_i, x) + b] \tag{4.9}$$

5. Le choix de C est critique si les données sont bruitées.

4.2.2 Représentation des documents

Il est souvent nécessaire de représenter les données textuelles sous une forme exploitable par les algorithmes d'analyse et de classification.

La pondération TF-IDF (Term Frequency - Inverse Document Frequency) permet de mesurer l'importance d'un mot en fonction de sa fréquence dans le document (TF) pondérée par la fréquence d'apparition du mot dans tout le corpus (IDF)SALTON et BUCKLEY [1988]. Elle permet de donner un poids plus important aux mots discriminants d'un document. Ainsi, un mot apparaissant dans tous les documents du corpus aura un poids faible.

Le poids d'un mot m_i dans un document d_j est calculé ainsi :

$$\mathrm{T}f.idf(m_i,d_j) = \mathrm{TF}(m_i,d_j).\mathrm{IDF}(m_i) \tag{4.10}$$

Où $\mathrm{TF}(m_i,d_j)$ est le nombre d'occurrences de m_i dans d_j, $\mathrm{IDF}(m_i) = log(\frac{|\mathrm{D}|}{|d_k;m_k \in d_k|})$ avec |D| le nombre de documents dans le corpus.

Il existe d'autres variantes du TF-IDF, qui n'ont pas été utilisées dans nos travaux mais sont données par NOBATA et collab. [2003] :

$$\mathrm{T}f.idf(m_i,d_j) = \frac{\mathrm{TF}(m_i,d_j)-1}{\mathrm{TF}(m_i,d_j)}\mathrm{IDF}(m_i) \tag{4.11}$$

$$\mathrm{T}f.idf(m_i,d_j) = \frac{\mathrm{TF}(m_i,d_j)}{\mathrm{TF}(m_i,d_j)+1}\mathrm{IDF}(m_i) \tag{4.12}$$

À la différence de la formule 4.10 qui utilise la fréquence des termes à l'état brut (mesure utilisée dans notre étude), les deux autres formules (4.11) et (4.12) sont utilisées pour normaliser la fréquence.

La particularité du TF-IDF est qu'elle donne un poids faible aux mots présents dans l'ensemble des documents car ils ne sont pas discriminants comme le cas des ``stop words''. Un tel traitement peut dont se révéler particulièrement pertinent pour le processus de classification qui n'utilisent pas de prétraitements (comme la suppression des ``stop words'') LAROUM et collab. [2010].

4.2.3 Sélection de caractéristiques

La sélection de caractéristiques est un enjeu crucial pour le processus de classification dans de nombreux domaines BRADLEY et collab. [1998]; DASH et LIU [1997]; JOHN et collab. [1994]. Il y a deux objectifs lors de l'optimisation des procédures de classification, atteignant la plus haute précision et la sélection d'un ensemble le plus petit et discriminant

possible de caractéristiques. Comme défini dans LIU et YU [2005], la sélection des caractéristiques est un processus qui sélectionne un sous-ensemble à partir de l'ensemble des caractéristiques de départ. Les algorithmes de sélection de caractéristiques suivent 4 étapes principales (illustrées dans la figure 4.2, qui sont : la génération du sous-ensemble, évaluation du sous-ensemble, arrêt d'évaluation et finalement la validation des résultats DASH et LIU [1997]; LIU et YU [2005]. L'étape principale est la génération du sous-ensemble de caractéristiques qui est un processus avec deux problèmes basiques :

1. Selon le point de recherche de départ, on distingue trois principales approches : (i) *La Sélection Forward* qui démarre avec aucune caractéristique et les ajoute au fur et à mesure une à une. A chaque étape, on ajoute une caractéristique qui minimise l'erreur jusqu'à ne plus pouvoir diminuer significativement le taux d'erreur ; (ii) *La Sélection Backward* qui démarre avec l'ensemble des caractéristiques et élimine celles qui augmentent le taux d'erreur d'une étape à l'autre et s'arrête lors d'une augmentation significative du taux d'erreur ; (iii) *La Sélection Stepwise* qui est une modification de la méthode Forward : les caractéristiques sont ajoutées ou éliminées jusqu'à obtention d'un ensemble optimal.
2. Selon la stratégie de recherche, on distingue la recherche complète, la recherche séquentielle et la recherche aléatoire. La recherche complète permet d'obtenir un résultat optimal selon le critère d'évaluation utilisé. La recherche séquentielle qui élimine progressivement les attributs qui ont l'influence la plus petite sur la fonction de coût du processus de classification. Cette recherche renonce à l'optimal et risque de perdre des sous-ensembles optimaux. La recherche aléatoire commence par un sous-ensemble aléatoirement choisi et procède de deux façons différentes.

Durant l'étape de génération du sous-ensemble des caractéristiques, chaque sous-ensemble candidat est évalué et comparé avec celui qui a donné les meilleurs résultats précédemment selon le critère d'évaluation choisi. Si le nouveau sous-ensemble améliore les résultats, il remplace le meilleur sous-ensemble précédent. Le processus de génération de sous-ensemble et d'évaluation se répète jusqu'à atteindre le seuil du critère défini au départ.

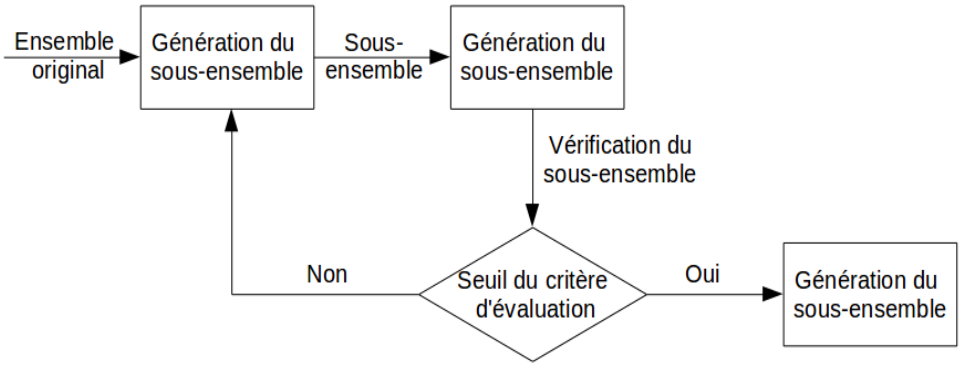


FIGURE 4.2 – Le processus de sélection de caractéristiques avec validation

Les algorithmes de sélection sont répartis en trois groupes principaux :

1. Les *filter methods* ESSEGHIR et collab. [2010]; LIU et YU [2005] ou méthodes filtres opèrent directement sur le jeu de données et fournissent une pondération, un classement ou un ensemble de variables en sortie. Ces méthodes ont l'avantage d'être rapides et indépendantes du modèle de classification, mais au prix de résultats inférieurs ;
2. Les *wrapper methods* ESSEGHIR et collab. [2010]; LIU et YU [2005] ou méthodes enveloppes effectuent une recherche dans l'espace des sous-ensembles de variables,

guidées par le résultat du modèle, par exemple les performances en validation croisée sur les données d'apprentissage. Elles ont souvent de meilleurs résultats que les méthodes de filtrage, mais au prix d'un temps de calcul plus important ;

3. Les *embedded methods* LAL et collab. [2006]; LIU et YU [2005] ou méthodes embarquées utilisent l'information interne du modèle de classification (par exemple, le vecteur de poids dans le cas des SVM – Support Vector Machine), ces méthodes sont donc proches des méthodes enveloppes, du fait qu'elles combinent le processus d'exploration avec un algorithme d'apprentissage sans étape de validation, pour maximiser la qualité de l'ajustement et minimiser le nombre d'attributs.

Nous présentons dans ce qui suit les deux méthodes de sélection utilisées dans nos expérimentations pour la détection automatique des comptes rendus de lecture.

4.2.3.1 RFE-SVM

Recursive Feature Elimination-Support Vector Machines (RFE-SVM) GUYON et collab. [2002] est un algorithme de sélection d'attributs pour la classification supervisée. Cet algorithme a été utilisé en bioinformatique pour l'analyse du niveau d'expression de gènes et dans l'analyse de données de transcriptome RAMASWAMY et collab. [2001]. Ceci a montré que RFE-SVM GUYON et collab. [2002] sélectionne de bons sous-ensembles d'attributs. Cependant, son utilisation a montré quelques limites comme le fait qu'il n'intègre pas de retours en arrière, et donc dans chaque récursion, l'attribut ou le sous-ensemble d'attributs (caractéristiques) éliminé ne peut plus jamais revenir dans le sous-ensemble sélectionné, ce qui aura pour effet de biaiser la recherche.

L'algorithme RFE-SVM fait partie des algorithmes de type *Embedded*. En effet, il intègre le filtrage dans le processus d'apprentissage SVM dans le but d'évaluer chaque sous-ensemble grâce à un classifieur SVM mais aussi pour avoir des informations sur la contribution de chaque attribut sur la construction de l'*hyperplan séparateur*.

La méthode repose sur le fait que chaque élément w_j du vecteur de poids w sur chaque variable j est une combinaison linéaire des observations et que la plupart des λ_i sont nuls, exceptés pour les observations support. Par conséquent, la mesure w peut être directement reliée à l'importance des variables dans le modèle SVM.

En effet, la mesure w^2 est une mesure de pouvoir prédictif. Ainsi, les variables j de plus petit poids w_j^2 seront progressivement éliminées dans la procédure RFE. Ces scores sont ici simplement les composantes du vecteur de poids w, définissant l'hyperplan optimal obtenu, qui est une combinaison linéaire des n vecteurs supports x_i du problème, faisant intervenir les multiplicateurs de Lagrange λ_i VAPNIK [1995].

$$w_{(j)} = \sum_{\text{SV}s} y_i \lambda_i x_{i(j)} \tag{4.13}$$

Où, $x_{i(j)}$ est la valeur du $j^{ème}$ attribut du $i^{ème}$ vecteur et y_i est la classe d'appartenance.

L'idée est que les attributs, qui correspondent à des directions de l'espace selon lesquelles le vecteur w admet une faible énergie, ne sont pas aussi utiles au problème que les autres attributs (puisqu'ils contribuent faiblement à la définition de l'hyperplan optimal). Donc à chaque récursion de l'algorithme RFE-SVM, l'attribut possédant le score le plus faible est éliminé. Le processus est arrêté lorsque le critère d'arrêt est atteint.

La procédure itérative de l'algorithme RFE-SVM s'écrit alors comme suit :

1. Apprentissage du SVM et obtention des poids $w_{(j)}$ de chaque attribut ;
2. Ordonnancement des attributs par leurs valeurs w^2 ;
3. Élimination de l'attribut ayant la plus petite valeur de w^2, re-apprentissage du SVM avec l'ensemble réduit d'attributs, obtention des nouveaux poids de chaque attribut ;
4. Si le nombre d'attributs de l'ensemble réduit est supérieur au nombre fixé, refaire (2), sinon ; arrêter ;
5. Retourner l'ensemble d'attributs obtenu.

L'algorithme SVM-RFE a été conçu pour résoudre les problèmes de la classification binaire, dans DUAN et RAJAPAKSE [2004]; LI et collab. [2004], les auteurs ont proposé une version pour les cas d'une classification multiple.

4.2.3.2 Sélection des caractéristiques avec le Z score et le Z score normalisé

Avec les méthodes de sélection de caractéristiques on vise à diminuer le coût de calcul en réduisant la dimension de l'espace des caractéristiques et garder les plus pertinentes pour le processus de classification. Un autre avantage des méthodes de sélection des caractéristiques est la possibilité de réduire le phénomène de surapprentissage (over-fitting). Ce dernier se produit quant le modèle entraîné donne de bons résultats une fois évalué avec le corpus d'apprentissage mais diminue considérablement les résultats avec le corpus de test non annoté. La méthode de sélection la plus simple et la plus couramment utilisée est l'élimination des mots vides et l'utilisation d'une fonction de stemming. Cependant, il existe des méthodes de calcul de score pour les mots dans le but de les classer et donc réduire l'espace des caractéristiques. KUMMER [2012]; SAVOY et ZUBAREYVA [2010] proposent de définir un poids à chaque terme suivant la méthode de Muller CHARLE [1992] dans le but de classer des phrases dans deux catégories : avec opinion (AO) ou sans opinion (SO) en choisissant les termes ou caractéristiques qui sont uniques ou les plus représentatifs de ces catégories. En sachant que :

— S correspond au sous-ensemble du corpus de départ C ;
— a représente le nombre d'occurrences du mot t_i dans l'ensemble de document S ;
— b représente le nombre des termes du même mot t_i dans le reste du corpus (dénoté par C−) ;
— c représente le nombre d'occurrences des termes qui ne correspondent pas au mot t_i dans l'ensemble S ;
— d représente le nombre d'occurrences des termes qui ne correspondent pas au mot t_i dans l'ensemble C− ;
— $a+b$ représente donc le nombre total d'occurrences du terme t_i dans le corpus de départ C ;
— $a+c$ indique le nombre total d'occurrences des termes dans l'ensemble S ;
— Le corpus de départ C consiste en l'union des deux sous-ensembles S et C− ($C = S \cup C-$) ;
— Le corpus C contient n termes ($n = a+b+c+d$).

Les auteurs supposent que la distribution du nombre de termes du mot t_i suit une loi binomiale avec les paramètres $Pr(t_i)$ et n'. Le paramètre $Pr(t_i)$ représente la probabilité d'occurrence du mot t_i dans un corpus C. Cette probabilité peut être estimée par : ($a+$

$b)/n$, si ce tirage aléatoire est répété $n' = a + c$ fois, elle donne une estimation du nombre d'occurrences du mot inclus dans le sous-ensemble S par $Pr(t_i).n'$. D'un autre côté a donne aussi le nombre d'observations du mot t_i dans S. Une grande différence entre a et la valeur produite par $Pr(t_i).n'$ est clairement une indication que la présence de a occurrences du terme t_i correspond à une caractéristique discriminante du sous-ensemble S comparé au sous-ensemble C−.

Dans le but d'obtenir une règle claire, les auteurs proposent de calculer le Z score attaché à chaque caractéristique t_i. Si la moyenne d'une distribution Binomiale est $Pr(t_i).n'$, sa variance est alors $n'.Pr(t_i).(1 - Pr(t_i))$. Ces deux éléments sont utilisés pour calculer le score standard comme décrit dans l'équation 4.14.

$$Z\ score(t_i) = \frac{a - (n'.Pr(t_i))}{\sqrt{n'.Pr(t_i).(1 - Pr(t_i))}} \tag{4.14}$$

Si on réécrit l'équation 4.14 en utilisant les variables a, b, c, d et n décrites précédemment, après simplification, on aura la formule suivante :

$$Z\ score(t_i) = \frac{a.d - c.b}{\sqrt{(a + c).(a + b).(c + d)}} \tag{4.15}$$

Les auteurs considèrent que les termes ayant un Z score entre l'intervalle $[-2, +2]$ appartiennent à un vocabulaire commun qui représente les caractéristiques discriminants pour la classification du corpus de référence MOAT NTCIR-6 (corpus en anglais) SEKI et collab. [2007]. Ce seuil a été choisi empiriquement. Un mot ayant un Z-score > 2 est considéré comme sur-utilisé, tandis qu'un mot ayant un Z score < -2 est considéré comme sous-utilisé. Afin de déterminer si une phrase contient un marqueur d'opinion, le Z-score de chaque mot de la phrase est récupéré. Comme règle d'agrégation, les auteurs calculent la somme des scores supérieurs à 2 (notée $sumPos$) et la somme des scores inférieurs à -2 (notée $sumNeg$). Si la $sumPos > |sumNeg|$, la phrase est placée dans la catégorie avec opinion (AO), sinon dans celle des phrases sans opinion (SO).

Le problème rencontré avec le Z score est le fait d'assigner un poids élevé aux mots qui apparaissent peu de fois dans le corpus. Dans le but de donner de l'importance au nombre d'apparitions du mot dans les deux catégories d'un corpus, KUMMER [2012] a employé une normalisation du Z score (décrite dans l'équation 4.17 en utilisant la mesure de normalisation ϕ introduite dans l'équation 4.16.

$$\phi = \frac{a - b}{a + b} \tag{4.16}$$

La normalisation ϕ est prise en compte pour calculer le Z score normalisé comme suit :

$$Z^{\phi}(w_i|C_j) = \begin{cases} Z(w_i|C_j).(1+|\phi(w_i|C_j|), si\ Z>0\ et\ \phi>0, \\ \quad ou\ si\ Z \leq 0\ et\ \phi \leq 0 \\ Z(w_i|C_j).(1-|\phi(w_i|C_j|), si\ Z>0\ et\ \phi \leq 0, \\ \quad ou\ si\ Z \leq 0\ et\ \phi>0 \end{cases} \tag{4.17}$$

ZUBARYEVA et SAVOY [2010] ont montré que l'utilisation du Z score normalisé dans leur modèle de représentation des phrases porteuses ou non d'opinion a permis d'apporter une qualité de réponses significativement supérieure aux autres approches plus complexes (Naïve Bayes et SVM). Il s'agit alors naturellement d'un indice pouvant être efficace pour la classification en genres.

4.2.4 Limites des schémas de représentation en sac de mots

La majorité des approches de classification axées sur les occurrences ou co-occurrences des termes ne tiennent pas compte de l'ordre dans lequel elles apparaissent, ni de leurs rôles syntaxique ou sémantique dans la phrase, le paragraphe ou le texte. Plusieurs chercheurs ont remédié à ces lacunes grâce à un pré-traitement des documents en exploitant la structure des documents pour les représenter (par exemple l'arborescence XML des documents dans les travaux de FÜRNKRANZ [1999]. Cependant, on cite quelques limitations majeures de la méthode de représentation en sac-de-mots :

— La présence de beaucoup de mots parasites due à la taille du vocabulaire construit à partir de l'ensemble des documents.

— En raison de la grande dimension de l'espace vectoriel, la construction des descripteurs de documents, le processus d'apprentissage et de classification peuvent prendre un temps de calcul très élevé.

— L'ambiguïté de certains mots, comme dans le cas d'une analyse d'opinion, le mot "*low*" est positive dans la phrase "*low price*" mais il est négatif dans "*low quality*".

Dans JAILLET et collab. [2003], les auteurs traitent de classification automatique supervisée de documents (corpus d'actualités). Bien que leur méthode de représentation se base sur le formalisme vectoriel, elle reste fondamentalement différente de la représentation en sac de mots SALTON et MCGILL [1986] car l'ensemble des termes est projeté sur un ensemble fini de concepts extrait d'un thesaurus. Leur méthode donne de bonnes performances et cela est dû au fait que la représentation de documents possède une information supplémentaire.

Dans BERNOTAS et collab. [2007], les auteurs ont utilisé une méthode de représentation de documents basée sur les tags qui décrivent leurs contenus et qui ont été attribués par les auteurs des documents. L'avantage principal de leur méthode est la possibilité d'assigner un document à plusieurs catégories. Chaque document est représenté par un vecteur de tags où chaque tag a un poids calculé par rapport à la collection des documents. Chaque document peut avoir un nombre différent de tags par rapport aux autres, cependant une normalisation additionnelle des vecteurs descriptifs des documents a été faite. Les auteurs ont montré dans leurs résultats que cette méthode agit négativement si la collection des documents est de petite taille car elle ne contient pas beaucoup de tags mais permet l'obtention de bonnes performances de classification si la collection est volumineuse.

Dans KHABBAZ et collab. [2012], les auteurs traitent de la classification des documents XML en considérant des caractéristiques tant structurelles qu'à base de contenu des documents. Leur approche mène à la construction d'un ensemble de vecteurs de caractéristiques

informatives qui représentent des aspects de contenu et structurels extraits en utilisant un algorithme de fouille dans l'arborescence XML des documents (tree-mining) combiné à un filtre de gain d'information pour rechercher les sous-structures XML les plus informatives. Les auteurs démontrent dans les résultats de leurs expériences l'efficacité de la combinaison des caractéristiques structurelles avec celles de contenu dans le processus d'apprentissage en comparaison avec les modèles de représentation basés seulement sur le contenu ou sur la structure des documents.

Nous avons présenté dans cette section un état de l'art qui regroupe la classification automatique et les différentes méthodes que nous allons utiliser pour la détection automatique des comptes rendus de lecture. Dans les sections suivantes, nous décrivons les données de test ainsi que les différentes expérimentations et résultats.

4.3 Description des plateformes d'OpenEdition

Le portail OpenEdition [6] (figure 4.3) contient quatre plateformes : Revues.org [7], Hypothèses [8], Calenda [9] et OpenEdition Books [10]. Ces plateformes sont dédiées aux ressources électroniques dans le domaine des sciences humaines et sociales (livres, revues, blogs de recherche, événements scientifiques, etc.). Créée en 1999, Revues.org est une plateforme de revues en sciences humaines et sociales. Elle accueille aujourd'hui 400 publications en ligne, soit plus de 100 000 articles, dont 95 % sont accessibles en texte intégral. Les sites de Revues.org reçoivent chaque mois une moyenne de 2,8 millions de visites. Fondée en 2009, Hypothèses contient aujourd'hui plus de 1 400 blogs scientifiques de recherche alimentés par une communauté de carnetiers de différentes nationalités. Tout le contenu de Hypothèses est en libre accès. Plusieurs types de carnets (recherche, séminaire, etc.) en différentes langues (français, allemand, anglais, etc.) sont présents dans Hypothèses. Calenda, une plateforme créée en 2000 pour proposer un panorama des sciences humaines et sociales. C'est un calendrier en libre accès qui informe étudiants, enseignants et chercheurs de l'actualité de la recherche. Ce calendrier est alimenté par les suggestions volontaires de ses utilisateurs. Finalement, OpenEdition Books, la nouvelle plateforme distribue des livres numériques. Contenant actuellement en 2016 plus de 2 500 livres, elle vise à construire une bibliothèque internationale numérique tout en encourageant les éditeurs à adopter le libre accès (Open Access) à long terme. Cette plateforme favorise tous les domaines culturels, à travers toutes les périodes historiques dans plusieurs langues.

Dans le portail OpenEdition, on trouve quelques revues et blogs consacrés uniquement aux critiques de livres. Dans la plateforme Revues.org, les documents sont déjà pré-classés en plusieurs catégories (articles, compte rendu, éditoriaux, etc. Voir tableau 4.2). Par contre, les blogs scientifiques ne possèdent pas de métadonnées, donc la recherche d'un type de document précis (par exemple : compte rendu de lecture) s'avère une tâche très difficile à établir manuellement vu la quantité de documents.

6. http ://www.openedition.org
7. http ://www.revues.org/
8. http ://hypotheses.org/
9. http ://calenda.org/
10. http ://books.openedition.org/

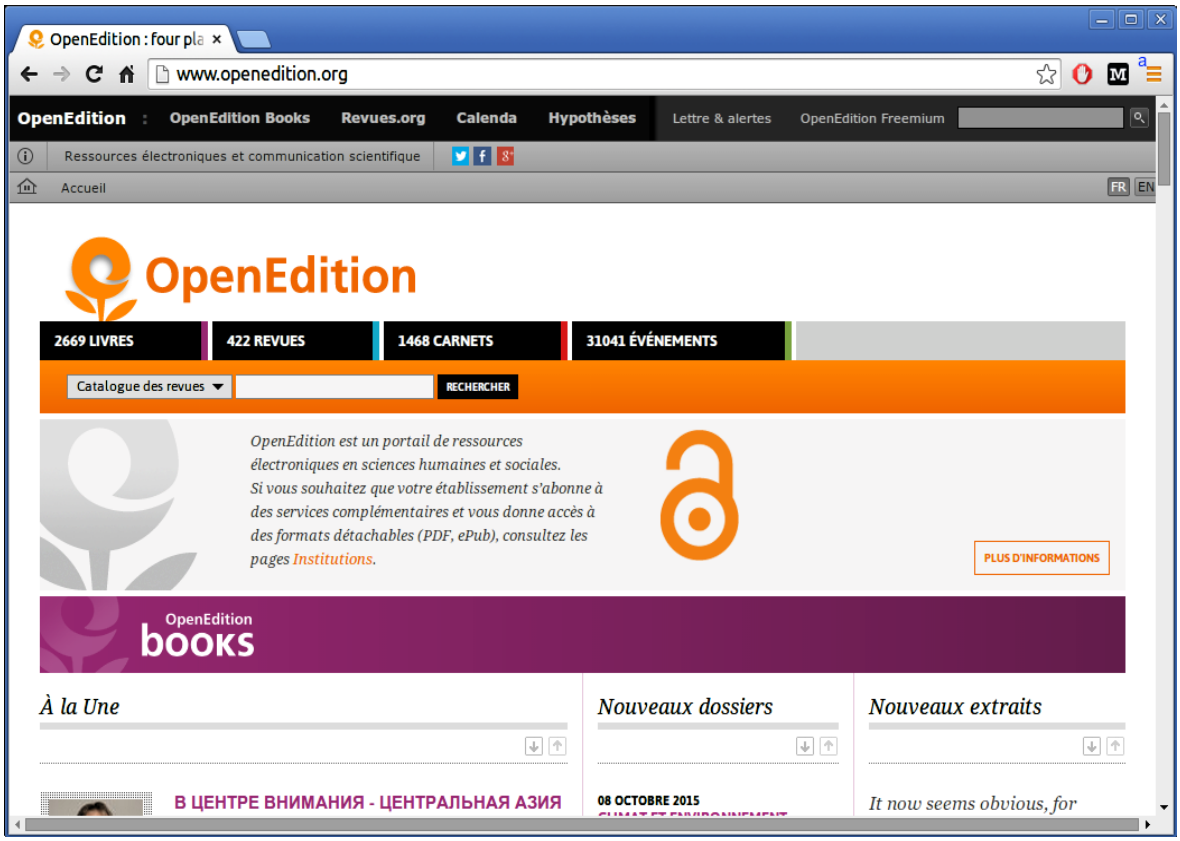


FIGURE 4.3 – La page d'accueil du portail OpenEdition. (capturée le 24/02/2016)

4.4 Corpus Revues.org

Dans cette section, nous décrivons les corpus d'apprentissage et d'évaluation que nous avons collectés à partir de la plateforme Revues.org. Contrairement aux autres collections de critiques/commentaires comme celles des films ou des restaurants que l'on trouve dans des sites Web, les comptes rendus de lectures sont beaucoup plus complexes à traiter car très souvent, ils sont eux-mêmes de longs articles scientifiques. S'il a été montré que les critiques de films (contenu et quantité) sont influencés par les ventes de films DUAN et collab. [2008], critiquer des livres scientifiques est l'une des activités des chercheurs. Par conséquent, les critiques sont une ressource très précieuse pour les travaux d'érudition dans un contexte de bibliothèque numérique. La figure 4.4 contient un extrait d'un compte rendu de lecture du livre *Wirtschaftsgeographie* des auteurs *Boris Braun* et *Christian Schulz*, publié dans la revue *articulo*. On remarque dans cet extrait de compte rendu des passages où l'auteur exprime son opinion et recommande la lecture de l'ouvrage.

On trouve dans le portail d'OpenEdition, beaucoup de textes qui semblent être des comptes rendus qui n'en sont pas. Cela rend le processus d'identification beaucoup plus difficile même pour un humain. Comme illustré dans la figure 6.4, un exemple de document qui ressemble à un compte rendu mais qui n'en est pas un, il s'agit d'un résumé du contenu de l'ouvrage. Sa structure est très similaire à un compte rendu de livre mais le texte ne contient aucune opinion ou phrase subjective. C'est l'une des raisons qui complique la tâche de construction automatique d'un corpus de critiques.

Full size image
Credits : © UTB GmbH

To come straight to the point, the new textbook *Wirtschaftsgeographie* [Economic Geography] by Boris Braun and Christian Schulz is a highly recommendable read. In pursuit of its goal to outline "*a very dynamic and multifarious sub-domain of geography*" [author's translation] (p. 249), this book is mainly directed at current and prospective students of the field of economic geography (p. 6). It keeps both these promises in quite an appealing manner. Alongside an insightful introduction to the myriad of theoretical structures applied in economic geography, its strength lies mainly in its engaging and stylistically coherent review as well as in the systematic cross-linkage of major topics. Valuable didactic tricks – such as info boxes giving a quick overview at the beginning of each chapter, additional text boxes that provide examples to underline the theoretical structures discussed, highlighted keywords in the body of the text, a rich stock of illustrations and charts, as well as review questions and recommendations for further reading – make this book a helpful companion for students. Topics are successfully cross-linked through clearly arranged and colour coded references of lexical precision as well as by repeatedly addressing, relating, and embedding major theoretical approaches from a variety of perception angles. The book embraces the exacting standards of both the authors, who explain them as follows: "*Currently relevant basic assumptions, theories, and models of economic geography, their differences as well as their manifold linkages shall be presented more vividly than in comparable textbooks*" [author's translation] (p. 6).

FIGURE 4.4 – Extrait d'un compte rendu de lecture (publié dans la revue *articulo*, URL : http://articulo.revues.org/2194

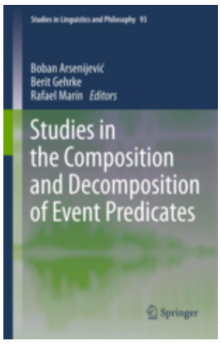
4.4.0.1 Corpus d'apprentissage

Pour ce travail de classification supervisée, nous avons utilisé un corpus d'apprentissage construit à partir de Revues.org. Nous considérons deux classes que nous appelons $Review$ et $\overline{Review}$. La classe $Review$ regroupe les comptes rendus de lecture. Dans la classe $\overline{Review}$, on trouve plusieurs types de documents comme par exemple : article, éditorial, chronique, chapitre d'ouvrage, etc. Ce corpus est équilibré, il contient 2000 documents en langue française (langue majoritaire dans Revues.org) sélectionnés aléatoirement dans chacune des deux classes et peut être utilisé pour de futures études comme l'analyse d'opinion et la recommandation BENKOUSSAS et collab. [2014]. Dans le tableau 4.1, nous présentons les statistiques réalisées sur le corpus d'apprentissage après suppression de toutes les balises XML-TEI [11]. Nous constatons que la taille en nombre moyen de mots des textes porteurs d'opinion représente $\simeq 32\%$ de la taille des textes ne portant aucune opinion. Ces résultats peuvent être une des caractéristiques de la classe $Review$.

Nous avons effectué une deuxième analyse sur les documents de la plateforme Revues.org (tableau 4.2). Nous remarquons que la langue française est majoritaire parmi les différentes langues de la plateforme (anglais, portugais, allemand, etc...).

11. Text Encoding Initiative (initiative pour l'encodage du texte) est une communauté académique internationale dans le champ des humanités numériques visant à définir des recommandations pour l'encodage de documents textuels. http://www.tei-c.org/index.xml

Nouvelle publication en linguistique : *Studies in the Composition and Decomposition of Event Predicates*



by **Boban Arsenijević**; **Berit Gehrke**; **Rafael Marín**, (eds.), Springer Verlag (Studies in Linguistics and Philosophy, 93), 2013, V, 79 p., 9 illus.

ISBN 978-94-007-5982-4

This book :

- Includes discussions of the processing aspects of the semantic ingredients of event predicates
- The role of scalar structures is shown to be more complex than traditionally assumed (one scale, simple mappings)
- Places the semantic entailments of event predicates and their more fine-grained components as the center of its study

This detailed, perceptive addition to the linguistics literature analyzes the semantic components of event predicates, exploring their fine-grained elements as well as their agency in linguistic processing. The papers go beyond pure semantics to consider their varying influences of event predicates on argument structure, aspect, scalarity, and event structure.

The volume shows how advances in the linguistic theory of event predicates, which have spawned Davidsonian and neo-Davidsonian notions of event arguments, in addition to 'event structure' frameworks and mereological models for the eventuality domain, have sidelined research on specific sets of entailments that support a typology of event predicates. Addressing this imbalance in the literature, the work also presents evidence indicating a more complex role for scalar structures than currently assumed. It will enrich the work of semanticists, psycholinguists, and syntacticians with a decompositional approach to verb phrase structure.

FIGURE 4.5 – Exemple de document qui ressemble à un compte rendu mais qui n'en est pas un

TABLEAU 4.1 – Statistiques sur les documents du corpus d'entraînement équilibré (classes $Review$ et $\overline{Review}$)

Nombre :	$Review$	$\overline{Review}$
de documents	2 000	2 000
minimum de mots	44	17
maximum de mots	5 226	31 563
moyen de mots	1 176	3 651

TABLEAU 4.2 – Statistiques sur la langue des documents de Revues.org

Classe	**Nombre de documents (type)**	**Documents en fr**	**Documents dans d'autres langues**	**% de documents en fr**
$Review$	31 263 (compte rendu)	29 020	2 243	92.82%
$\overline{Review}$	71 521	57 829	13 692	80.85%
	* 61 162 articles	50 868	10 293	
	* 3 456 éditoriaux	2 960	496	
	* 2 693 actualités	2 494	199	
	* 1 388 chroniques	1 308	80	
	* 2 822 autres ...	199	2 624	

4.4.0.2 Corpus de test

Désirant fonder nos conclusions sur une base plus solide, nous avons évalué les méthodes de classification sur deux corpus de test typés manuellement par des experts d'OpenEdition et créés à partir de Revues.org. Ces corpus sont pré-classés manuellement en deux classes (R$eview$ et $\overline{\text{R}eview}$). Le premier est peu volumineux, sa classe R$eview$ contient 200 documents issus des revues *Balkanologie* (son orientation est pluridisciplinaire et son objectif est de contribuer à une meilleure compréhension du monde balkanique contemporain) et *Clio* (destinée à celles et ceux qui mènent des recherches en histoire des femmes et du genre). Quant à la classe $\overline{\text{R}eview}$, elle contient 100 documents issus des revues *Archeosciences*, *Ashp*, *Asiecentrale*, *Asp* et *Assr*. Le deuxième corpus de test est très volumineux, il représente toute la plateforme Revues.org. Il est écrit dans plusieurs langues (majoritairement le français mais aussi anglais, portugais, arabe, allemand, etc.). Dans le tableau 4.3, nous détaillons sa composition.

TABLEAU 4.3 – Composition du deuxième corpus d'évaluation utilisé (Revues.org)

Classe	Nombre de documents
R$eview$	31 263
$\overline{\text{R}eview}$	71 521 ⟶ 61 162 articles ⟶ 3 456 editoriaux ⟶ 2 693 actualités ⟶ 1 388 chroniques ⟶ 2 822 autres ...

Notons que tous les documents du corpus d'apprentissage et d'évaluation sont en format XML-TEI. Pour évaluer le rôle de la structure des textes dans le processus de classification, nous avons effectué plusieurs étapes de représentation de documents pour ensuite procéder à une étape de suppression de toutes les balises XML-TEI avant le processus d'entraînement des modèles d'apprentissage pour ne se focaliser que sur le contenu (texte) et ignorer la structuration.

4.5 Schémas d'indexation

Trois techniques de prétraitement sont proposées dans cette section dont deux d'entre elles se basent sur les mots présents dans le textes.On suppose que les auteurs des comptes rendus scientifiques utilisent les mêmes mots pour critiquer un ouvrage quelle que soit la thématique. La question suivante se pose : *Peut-on dire que, dans notre cas, le mot est la meilleure unité pour la représentation et la classification du texte en genre (*R$eview$ *ou* $\overline{\text{R}eview}$*) ?*

Pour mieux répondre à cette question, nous expérimentons plusieurs méthodes d'indexation basée sur les mots et nous proposons une nouvelle méthode d'indexation qui exploite l'organisation des contenus des textes. Il s'agit de d'utiliser la distribution des entités nommées dans le texte comme indice supplémentaire pour la classification.

4.5.1 Pondération des mots par fréquence

Dans cette approche, nous considérons l'utilisation de la pondération *tf*idf* pour décrire l'importance relative d'un mot pour une classe de document particulière (la classe $Review$ dans notre cas). Chaque valeur *tf*idf*, pour chaque mot, est utilisée comme un poids indicateur de pertinence par rapport aux deux classes ($Review$ et $\overline{Review}$). La métrique *tf*idf* est constituée de deux composantes : term frequency (*tf*) et inverse document frequency (*idf*). Ces deux composantes sont multipliées lors du calcul du *tf*idf*. On peut choisir de garder seulement les mots dont la valeur de *tf*idf* dépasse un certain seuil ou limiter le nombre de mots à un pourcentage relatif au nombre total des mots. Nous avons sélectionné les 5 000 premiers mots ($\simeq 5\%$) triés par ordre décroissant de *tf*idf*.

4.5.2 Réduction de l'espace vectoriel avec le Z-score normalisé

Avec les méthodes de sélection de caractéristiques nous visons à réduire la dimension de l'espace des caractéristiques en identifiant les plus pertinentes. La méthode de sélection la plus couramment utilisée et l'élimination des mots outils et l'utilisation d'une fonction de stemming.

Pour déterminer les caractéristiques qui aident à distinguer les $Review$ des $\overline{Review}$, nous avons utilisé une méthode de sélection basée sur le z-score normalisé (décrit dans la section 4.2.3.2). Afin de mesurer la distance entre les textes pour identifier leurs genre, le corpus est subdivisé en deux classes ($Review$ et $\overline{Review}$) comme il est décrit précédemment. La classe $Review$ (notée C_0) et la classe $\overline{Review}$ (notée C_1). Pour un mot w, nous calculons son nombre d'occurrences dans la classe C_0 et C_1 (notés respectivement tf_{C0} et tf_{C1}). La taille de l'ensemble C_0 s'élève à n_{C0} tant dis que le volume du corpus est $n = n_{C0} + n_{C1}$.

Pour définir le pouvoir discriminant d'un mot, on fait l'hypothèse que sa distribution suit une loi binomiale de paramètres $P(w)$ et n_{C1}. $P(w)$ décrit dans l'équation 4.18 représente la probabilité de l'occurrence du mot w dans le corpus.

$$P(w) = \frac{tf_{C0} + tf_{C1}}{n_{C0} + n_{C1}} \tag{4.18}$$

Nous avons calculé le z-score pour chaque mot w selon l'équation 4.19.

$$Z - score(w_i) = \frac{tf_{C0} - n_{C0}.P[w]}{\sqrt{n_{C0}.P[w].(1 - P[w])}} \tag{4.19}$$

Nous avons employé l'approche normalisée du z-score (décrite dans l'équation 4.17 dans la section 4.2.3.2) en utilisant la mesure de normalisation ϕ que nous avons présentée dans l'équation 4.16.

Après le calcul du z-score normalisé, nous avons sélectionné tous les mots qui apparaissent plus de 5 fois dans chaque classe et nous obtenons un ensemble réduit de caractéristiques (réduction de l'espace vectoriel de près de 50%). Le but est de concevoir une méthode capable de sélectionner les mots les plus discriminants par rapport à la classe $Review$ et nous supposons que les mots moins fréquents dans le corpus ne répondent pas à notre besoin en classification par genre car ils présentent pas un pouvoir discriminant. Dans le tableau 4.4, les 30 premiers mots, avec les Z-scores normalisés les plus élevés, sont présentés. Nous n'avons pas appliqué un stemmer sur les données avant le calcul des scores.

TABLEAU 4.4 – Distribution des 30 premiers mots ayants des z-score les plus élevés (à partir du corpus d'apprentissage équilibré

#	mots	Z^{ϕ} score	#	mots	Z^{ϕ} score
1	doivent	112,21	16	cadre	27,41
2	d'éléments	70,46	17	connotations	27,39
3	représente	58,60	18	fédérer	27,37
4	davantage	52,26	19	national	27,12
5	jimmu	42,82	20	chrétiens	26,99
6	see	40,37	21	règles	26,75
7	soutenu	36,27	22	situation	26,43
8	effectue	34,53	23	penetrating	25,99
9	arif	33,84	24	émerger	25,30
10	attaché	33,57	25	yung	25,12
11	ayant	31,97	26	véricourt	23,90
12	much	31,90	27	utile	23,80
13	groupe	31,57	28	transferred	23,75
14	contexte	29,76	29	désacralisé	22,79
15	nécessaire	27,42	30	seventh	21,79

Nous pouvons constater l'existence de beaucoup de mots généralement utilisés pour décrire et critiquer un ouvrage comme par exemple (représente, effectue, nécessaire, contexte, utile, etc.). Ces mots vont nous servir de base à la construction des descripteurs des documents. Nous avons construit un descripteur binaire pour chaque document ou chaque élément indique la présence ou l'absence des mots préalablement sélectionnés ayant un nombre d'occurrence supérieur à 5 fois (choisie empiriquement) et possédant un Z-score normalisé supérieur à un seuil égal à 3,6 (ce seuil est choisi après étude de la liste ordonnée des mots). Ce dernier est fixé pour réduire l'espace des caractéristiques en considérant que les mots sont jugés discriminants par leurs valeurs de Z-score normalisé.

4.5.3 Distribution des entités nommées

Nous nous intéressons dans cette section à une représentation de texte qui exploite principalement les résultats de la détection des entités nommées dans le texte. On suppose que les entités nommées vont permettre l'identification de l'organisation du contenu textuel ce qui apporte des informations importantes au processus de détection automatique des comptes rendus de lecture.

En faveur du développement de la tâche d'extraction d'information que la tâche de reconnaissance des entités nommées (EN) est apparue. Cette tâche a gagné en maturité et s'est précisée grâce à la série des conférences MUC (Message Understanding Conferences) CHINCHOR et collab. [1999] suivi de la conférence CoNLL (Computational Natural Language Learning) qui a inclus une tâche de détection des entités nommées pour plusieurs langues SANG et MEULDER [2003]. Ensuite, le concept des entités nommées a été repris dans le cadre des campagnes d'évaluation du projet européen QUAERO GALIBERT et collab. [2012].

4.5.3.1 La tâche de reconnaissance d'entités nommées

L'objectif de la tâche de reconnaissance d'entités nommées est d'extraire et de typer des éléments informationnels d'un texte. Ces éléments sont des unités lexicales particulières, faisant référence à des noms de personnes, des noms d'organisations ou des localisations. D'autres types d'entités nommées peuvent être considérés comme les dates, les unités monétaires ou les quantités de tout genre. La tâche de reconnaissance d'entités nommées se décline souvent en deux sous traitements : (1) identifier ces unités dans un texte, (2) les catégoriser en fonction des types prédéfinis de classes

L'élaboration de la tâche de reconnaissance d'entités nommées commence par définir quelle est l'unité que nous voulons extraire et comment l'organiser. Déterminer les classes d'entités à prendre en compte revient à trouver une classification sémantique des unités lexicales dans le texte.

Dans notre contribution, nous nous sommes intéressés à trois types d'entités nommées : *Personne (Person)*, *Lieu (Location)* et *Date*. Nous supposons que ces trois unités fournissent l'information utile pour détecter les comptes rendus de lecture.

L'annotateur automatique d'entités nommées que nous avons utilisé est l'outil TagEN. Ce dernier est basé sur des transducteur à états finis, il a été développé par Jean-François Berroyer et Thierry Poibeau au Laboratoire d'Informatique de Paris-Nord (LIPN) POIBEAU [2003]. Cet annotateur est fondé sur des lexiques et des grammaires codées sous forme d'automates à nombre fini d'états. Même si les techniques d'apprentissage fonctionnent bien pour la reconnaissance des entités, les développeurs de TagEN ont choisi d'employer des automates dans la mesure où ceux-ci peuvent garantir un bon taux de reconnaissance et une bonne lisibilité des ressources.

L'annotateur TagEN utilise l'environnement Unitex [12]. Il s'agit d'un logiciel libre adaptable et intégrable à une chaîne de traitements. Il comporte un ensemble de programmes permettant de manipuler des transducteurs.

Les ressources utilisées par TagEN sont de deux types : d'une part des dictionnaires, de l'autre part des grammaires. En plus des dictionnaires fournis par Unitex, il est nécessaire de développer des dictionnaires spécifiques, encodant les informations pertinentes pour l'analyse des entités nommées. Il s'agit de listes de noms et de prénoms, de noms de lieux, etc. Ces listes doivent contenir le maximum d'information pour assurer au mieux la couverture (plus de détails dans POIBEAU [2003]), sachant qu'aujourd'hui, il est possible de trouver sur le Web des listes de plusieurs centaines de milliers d'entrées désignant des entités [13]. Les grammaires de reconnaissance sont codées sous forme de transducteurs récursifs POIBEAU [2003].

Nous avons évalué le système TagEN sur un corpus annoté manuellement extrait de la plateforme Revues.org OLLAGNIER et collab. [2014]. Un exemple de l'une des références est présenté dans ce qui suit :

Référence non annotée :
Lenclos, Jean-Philippe & Lenclos, Dominique. 2003. *Couleurs du monde : géographie de la couleur*. Paris : Ed. Le Moniteur. 288 pages.

12. http ://www-igm.univ-mlv.fr/ unitex/
13. http://www.lattice.cnrs.fr/IMG/pdf/traitement-contenu-textuel.pdf

Référence annotée :

```
<bibl>
 <author>
   <surname>Lenclos</surname>,
   <forename>Jean-Philippe</forename>
 </author> and
 <author>
   <surname>Lenclos</surname>,
   <forename>Dominique</forename>
 </author>. <date>2003</date>.
 <hi rend="italic">
   <title level="m">
     Couleurs du monde : geographie de la couleur
   </title>
 </hi>.
 <pubPlace>Paris</pubPlace> :
 <publisher>Ed. Le Moniteur</publisher>.
 <extent>288 pages</extent>.
</bibl>
```

Ce corpus contient des références bibliographiques avec plusieurs types d'entité nommée. Les entités qui nous intéressent sont *Person* (602 entités), *Date* (712 entités) et *Location* (316 entités).

TABLEAU 4.5 – Résultats d'évaluation de l'annotateur TagEN sur un ensemble de références bibliographiques issues de Revues.org

L'entité nommée	Rappel	Précision
Personne	78,89%	32,78%
Date	88,66%	74,01%
Location	50,80%	54,10%

Les résultats obtenus présentés dans le tableau 4.5 montrent des performances assez satisfaisantes en terme de rappel pour un annotateur qui utilisent des ressources (dictionnaires, corpus d'apprentissage) autres que celles utilisées pour le test. En effet, TagEN repose sur des modèle issus du corpus Quaero Oral (voir exemple dans la figure 4.6) et donc très différent des références bibliographiques de Revues.org (voir exemple figure 4.7) . Néanmoins, les résultats restent suffisants pour les entités nommées qui nous intéressent.

Certaines annotations attribuées par TagEN sont erronées parce qu'il s'agit des termes spécifiques au domaine du corpus Revues.org comme par exemple le mot "livre"qui est reconnu comme une monnaie et aussi "Université de Cambridge"où TagEN annote seulement "Cambridge"comme *Location* au lieu d'annoter l'ensemble comme *Organisation*.

La tâche d'annotation des entités nommées effectuée pour le corpus d'entraînement et de test de Revues.org nous ont permis d'avoir une vue globale de leurs répartitions dans les textes. Nous détaillons dans la section suivante la méthode de transformation des documents annotés par TagEN en un ensemble de vecteurs d'attributs que nous utilisons pour la classification.

```
1   que la môme Marion .
2   j' ai grandi dans une ferme de Mooresville dans l' Indiana , ma mère est morte
    quand j' avais trois ans , mon père m' a roué de coups parce qu' il ne connaissait
    pas de meilleure façon de m' élever .
3   j' aime bien le base-ball , le cinéma , les beaux vêtements , les voitures rapides
    , le whisky ,
4   et toi ?
5   le base-ball ,
6   le cinéma , les beaux vêtements , le whisky et toi .
7   la vie , c' est un truc hypersimple en fait .
8   &ouais
9   c' est simplissime
10  même
11  .
12  le coup de foudre de la semaine pour terminer .
13  euh
14  il s' est déroulé entre Claire Chazal et Jim Carréey
15  sur le plateau du 20 Heures .
16  dans un instant la météo de Catherine Laborde
17  et
18  puis ce sera encore du cinéma ,
19  bye ! au revoir !
20   demain vous retrouverez Jean-Pierre Pernaut à
21   13 h
```

FIGURE 4.6 – Extrait d'un document du corpus Quaero

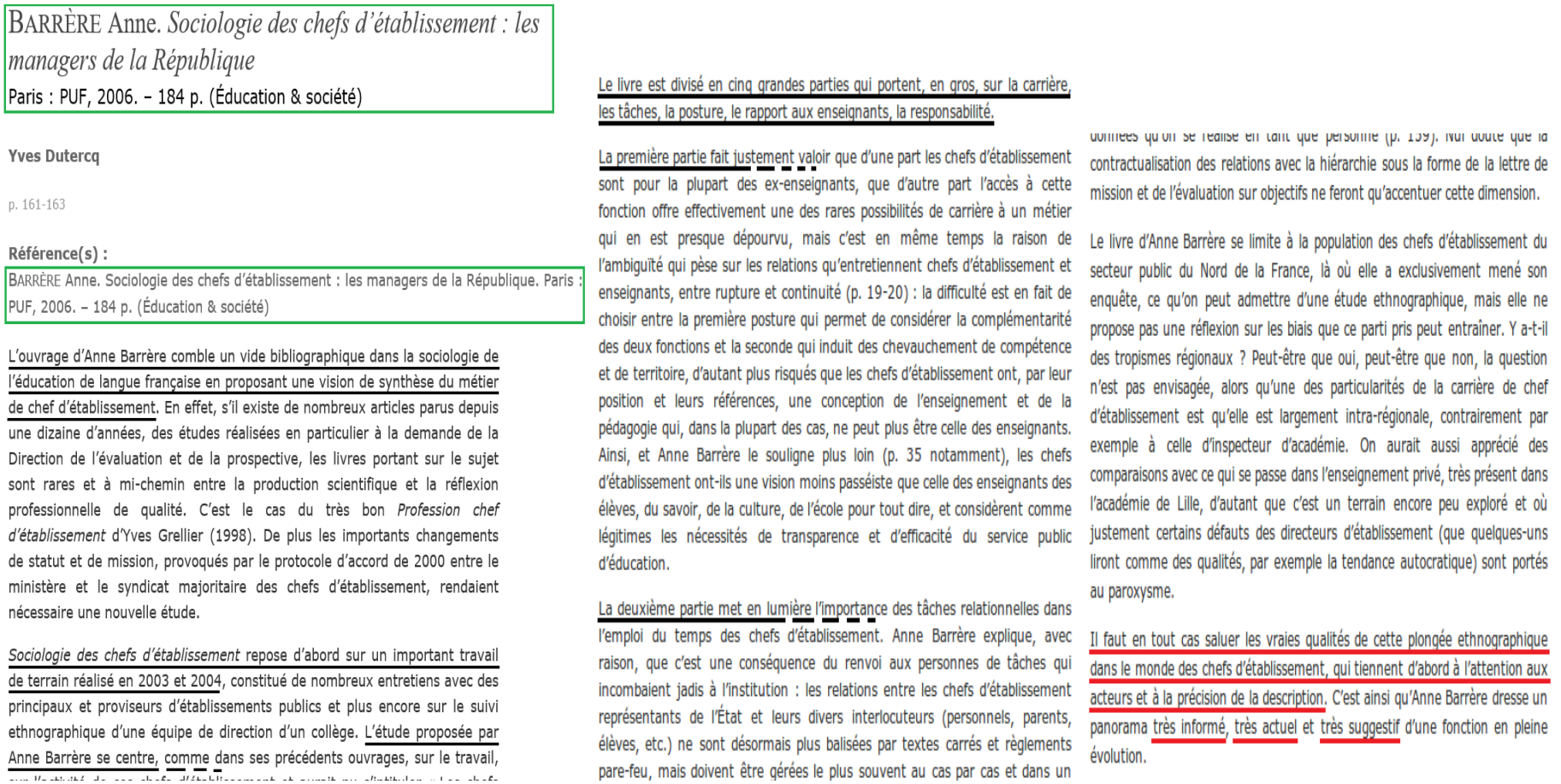
```
−<p>
  Les
  <hi rend="italic">Principia </hi>
  qui viennent de paraître
  <hi rend="italic">, </hi>
  pourrait-on penser, viennent alors former la synthèse de cette forte et complexe réflexion philosophique
  dispersée dans les livres antérieurs. Certes, ils forment cette synthèse, qui était attendue, de la rhétorique
  intégrée à la philosophie du questionnement, à la problématologie formulée par le philosophe bruxellois,
  mais les
  <hi rend="italic">Principia Rhetorica </hi>
  sont autre chose encore, de plus ambitieux et de plus englobant. À cet égard, un peu curieusement, le
  sous-titre de ce livre est trompeur : « une théorie générale de l'argumentation » ? Non, c'est trop restreint
  et trop « modeste ». L'ambition de Meyer, ambition qui structure tout le livre, est de prendre à bras le corps
  et d'inscrire dans un ensemble philosophiquement cohérent les pratiques discursives, dans toute leur
  diversité, qui permettent « la négociation de la distance entre des individus à propos d'une question
  donnée » – pratiques qui vont de l'argumentation apodictique, de la coopération en vue d'accord, à la
  <hi rend="italic">disputatio, </hi>
  à l'éristique des « conflits », à la figuralité poétique, à l'expression énigmatique ou hermétique. Le livre
  part de cette définition et aboutit à un vaste questionnement typologique, développé à grands traits,
  « Comment négocie-t-on la distance entre individus ? » (227 et ss.)
  <hi rend="sup"> </hi>
</p>
```

FIGURE 4.7 – Extrait d'un document du corpus Revues.org

4.5.3.2 Représentation vectorielle basée sur la distribution des entités nommées

La représentation de documents que nous décrivons consiste à exploiter une analyse linguistique et statistique pour déterminer les caractéristiques communes entre les comptes rendus et les autres types de documents. La figure 4.8 présente un exemple de compte rendu publié dans la plateforme Revues.org.

Dans la majorité des comptes rendus de lecture, les auteurs choisissent comme titre le titre du livre avec le nom de son(ses) auteur(s) avec en plus l'année de publication de l'ouvrage ce qui constitue une partie de la référence bibliographique. Ils commencent généralement par présenter le livre et détailler le domaine ou le sujet traité en rajoutant, parfois, des citations (des noms d'auteur, des titres, etc.) vers d'autres ouvrages du même sujet. Ensuite ils décrivent chacune des parties/chapitres de telle sorte que le lecteur ait un aperçu sur les idées développées par l'auteur de l'ouvrage parfois critiquées par l'auteur du compte rendu. L'exemple illustré dans la figure 4.8 montre cette structuration avec les soulignements en noir. À la fin de la plupart des comptes rendus les auteurs expriment leurs opinions sur les



BARRÈRE Anne. *Sociologie des chefs d'établissement : les managers de la République*
Paris : PUF, 2006. – 184 p. (Éducation & société)

Yves Dutercq

p. 161-163

Référence(s) :

BARRÈRE Anne. Sociologie des chefs d'établissement : les managers de la République. Paris : PUF, 2006. – 184 p. (Éducation & société)

L'ouvrage d'Anne Barrère comble un vide bibliographique dans la sociologie de l'éducation de langue française en proposant une vision de synthèse du métier de chef d'établissement. En effet, s'il existe de nombreux articles parus depuis une dizaine d'années, des études réalisées en particulier à la demande de la Direction de l'évaluation et de la prospective, les livres portant sur le sujet sont rares et à mi-chemin entre la production scientifique et la réflexion professionnelle de qualité. C'est le cas du très bon *Profession chef d'établissement* d'Yves Grellier (1998). De plus les importants changements de statut et de mission, provoqués par le protocole d'accord de 2000 entre le ministère et le syndicat majoritaire des chefs d'établissement, rendaient nécessaire une nouvelle étude.

Sociologie des chefs d'établissement repose d'abord sur un important travail de terrain réalisé en 2003 et 2004, constitué de nombreux entretiens avec des principaux et proviseurs d'établissements publics et plus encore sur le suivi ethnographique d'une équipe de direction d'un collège. L'étude proposée par Anne Barrère se centre, comme dans ses précédents ouvrages, sur le travail,

Le livre est divisé en cinq grandes parties qui portent, en gros, sur la carrière, les tâches, la posture, le rapport aux enseignants, la responsabilité.

La première partie fait justement valoir que d'une part les chefs d'établissement sont pour la plupart des ex-enseignants, que d'autre part l'accès à cette fonction offre effectivement une des rares possibilités de carrière à un métier qui en est presque dépourvu, mais c'est en même temps la raison de l'ambiguïté qui pèse sur les relations qu'entretiennent chefs d'établissement et enseignants, entre rupture et continuité (p. 19-20) : la difficulté est en fait de choisir entre la première posture qui permet de considérer la complémentarité des deux fonctions et la seconde qui induit des chevauchement de compétence et de territoire, d'autant plus risqués que les chefs d'établissement ont, par leur position et leurs références, une conception de l'enseignement et de la pédagogie qui, dans la plupart des cas, ne peut plus être celle des enseignants. Ainsi, et Anne Barrère le souligne plus loin (p. 35 notamment), les chefs d'établissement ont-ils une vision moins passéiste que celle des enseignants des élèves, du savoir, de la culture, de l'école pour tout dire, et considèrent comme légitimes les nécessités de transparence et d'efficacité du service public d'éducation.

La deuxième partie met en lumière l'importance des tâches relationnelles dans l'emploi du temps des chefs d'établissement. Anne Barrère explique, avec raison, que c'est une conséquence du renvoi aux personnes de tâches qui incombaient jadis à l'institution : les relations entre les chefs d'établissement représentants de l'État et leurs divers interlocuteurs (personnels, parents, élèves, etc.) ne sont désormais plus balisées par textes carrés et règlements pare-feu, mais doivent être gérées le plus souvent au cas par cas et dans un

données qu'on se réalise en tant que personne (p. 159). Nul doute que la contractualisation des relations avec la hiérarchie sous la forme de la lettre de mission et de l'évaluation sur objectifs ne feront qu'accentuer cette dimension.

Le livre d'Anne Barrère se limite à la population des chefs d'établissement du secteur public du Nord de la France, là où elle a exclusivement mené son enquête, ce qu'on peut admettre d'une étude ethnographique, mais elle ne propose pas une réflexion sur les biais que ce parti pris peut entraîner. Y a-t-il des tropismes régionaux ? Peut-être que oui, peut-être que non, la question n'est pas envisagée, alors qu'une des particularités de la carrière de chef d'établissement est qu'elle est largement intra-régionale, contrairement par exemple à celle d'inspecteur d'académie. On aurait aussi apprécié des comparaisons avec ce qui se passe dans l'enseignement privé, très présent dans l'académie de Lille, d'autant que c'est un terrain encore peu exploré et où justement certains défauts des directeurs d'établissement (que quelques-uns liront comme des qualités, par exemple la tendance autocratique) sont portés au paroxysme.

Il faut en tout cas saluer les vraies qualités de cette plongée ethnographique dans le monde des chefs d'établissement, qui tiennent d'abord à l'attention aux acteurs et à la précision de la description. C'est ainsi qu'Anne Barrère dresse un panorama très informé, très actuel et très suggestif d'une fonction en pleine évolution.

FIGURE 4.8 – Exemple de la structuration des comptes rendus dans Revues.org. Compte rendu du livre *Sociologie des chefs d'établissement : les managers de la République* de l'auteur *BARRIÈRE Anne*, URL : http://rfp.revues.org/525(*Les cadres en vert regroupent la référence ou une partie de la référence du livre critiqué. Les soulignements en noir représentent les passages de description du contenu du livre et ceux en rouge représentent l'avis de l'auteur du compte rendu.*)

livres d'une manière générale avec une(des) phrase(s) directe(s) comme "*En conclusion, cet ensemble, ouvrage et site Internet, est un outil qui sera fort utile à un public très large d'utilisateurs ...*" [14] et recommandent ou non l'ouvrage comme le cas avec cette phrase "*la lecture en est vivement recommandée*" [15] ou encore "*Les critères qu'elle dégage et leur agencement seront utiles à ceux qui s'intéressent à la terminologie systématique.*" [16].

D'autre part, les comptes rendus de lecture regroupent une caractéristique commune qui est l'absence de la partie bibliographie dans la majorité des cas puisqu'il s'agit de donner une description d'un ouvrage ou article donc on ne cite pas forcement d'autres œuvres scientifiques. La partie bibliographie regroupe un nombre important d'entités nommées (les noms d'auteur, les dates de publication, les lieux de publication, etc.).

Nous avons annoté avec TagEN un corpus composé de 498 $Review$ et 288 $\overline{Review}$ choisis aléatoirement à partir de la plateforme Revues.org. Ensuite nous avons divisé le texte en 10 parties selon le nombre de mots et avons calculé pour chaque partie le taux de répartition de chacune des 3 types d'entités nommées (personnes, date et lieu).

La figure 4.9 illustre les ratios de chaque entité nommée dans chaque partie dans les deux ensembles du corpus. Le premier diagramme représente la distribution de l'entité *Person*. Les courbes montrent la forte présence de cette entité dans la première partie des documents dans les deux classes $Review$ et $\overline{Review}$. Cette partie des documents contient le titre composé, dans la plupart des cas, de la référence ou une partie de celle-ci qui contient le(s) nom(s) de(s) auteur(s) avec l'intitulé du document comme l'exemple illustré dans la figure 4.8. Les courbes des trois entités pour la classe $\overline{Review}$ marquent une croissance dans les dernières parties des textes ce qui est justifié par la présence de la partie bibliographique dans les documents non porteurs d'opinion.

Les documents de la collection ont été transformés en un ensemble de descripteurs suivant l'algorithme 1, chaque descripteur contient la classe ($Review \rightarrow 1$ / $\overline{Review} \longrightarrow 0$) et les taux de répartition des entités nommées qui représentent les caractéristiques statistiques. Au final, nous obtenons des descripteurs contenant 30 caractéristiques ce qui rend le processus d'apprentissage plus rapide par rapport aux descripteurs basés sur des sacs de mots qui sont plus volumineux.

Algorithme 1 Transformation des documents en descripteurs

Pour Chaque document doc de la collection **faire**
  Ajouter la classe du doc (1 si $Review$, 0 sinon) au $Vecteur(doc)$
  docEN ← Annotation des EN de doc
  Diviser doc_EN en 10 parties
  Pour i de 1 à 10 **faire**
    ▷ Pour caque partie calculer les taux de répartition de chaque entité nommée
    R_$person_i$ ← $doc\text{EN}_i(Person)/nbr\_mots(doc_i)$
    R_$location_i$ ← $doc\text{EN}_i(Location)/nbr\_mots(doc_i)$
    R_$date_i$ ← $doc\text{EN}_i(Date)/nbr\_mots(doc_i)$
    Ajouter (R_$person_i$, R_$location_i$ et R_$date_i$) au $Vecteur(doc)$

14. http://apliut.revues.org/2316
15. http://apliut.revues.org/2417
16. http://asp.revues.org/306

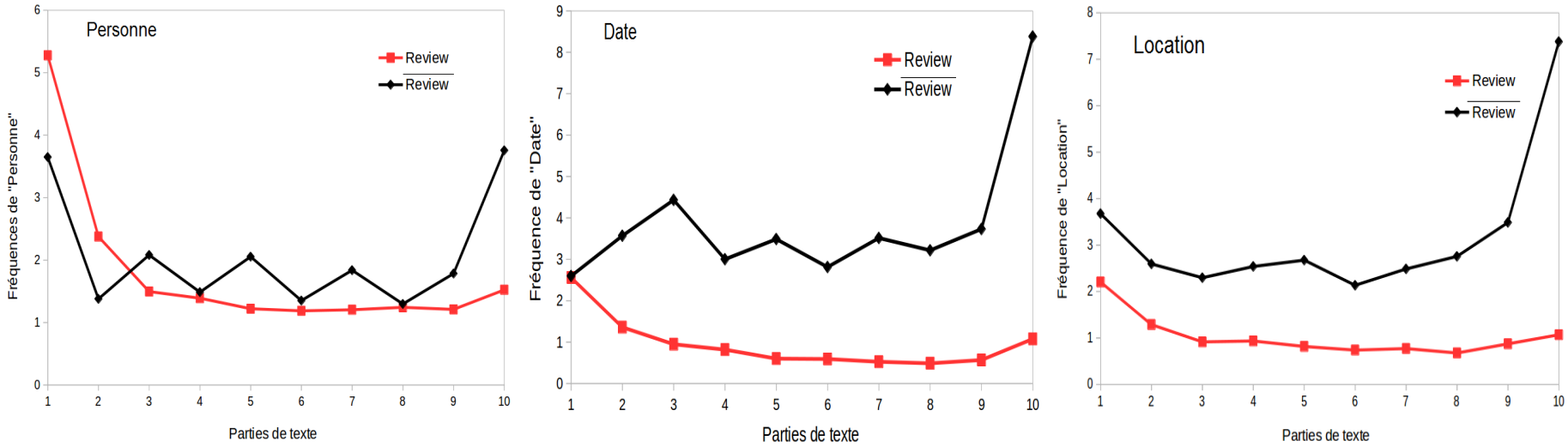


FIGURE 4.9 – Diagrammes des distributions des entités nommées *Person*, *Date* et *Location* dans les deux classes R$e v i e w$ et $\overline{\mathrm{R}e v i e w}$.

4.6 Expérimentations

Dans cette section, nous présentons les métriques d'évaluation et les différentes expérimentations effectuées pour évaluer l'efficacité des méthodes d'apprentissage et classification et, plus particulièrement, l'efficacité des différents schémas d'indexation.

4.6.1 Métriques d'évaluation d'un modèle d'apprentissage

Nous utilisons les mesures classique d'évaluation : Rappel, Précision et F-mesure. En principe, on cherche l'équilibre entre les deux mesures rappel et précision en tenant compte du type d'application visée. Par exemple, si on est dans une logique de veille, le rappel est favorisée. Si on est dans une logique d'application de type extraction d'information pour une alimentation d'un portail, il est important que tous les documents soient affectés à au moins une classe. Dans ce cas c'est le rappel qui est favorisé.

Le rappel mesure la capacité du système à donner tous les résultats pertinents. Il est calculé comme suit :

$$\text{R}appel = \frac{\text{D}ocuments\ correctement\ attribués\ \text{à}\ la\ classe_i}{\text{N}ombre\ de\ documents\ \textbf{appartenant}\ réellement\ \text{à}\ la\ classe_i} \tag{4.20}$$

La précision mesure la capacité du système à refuser tous les résultats non pertinents. Elle est calculée pour la classe i comme suit :

$$\text{P}récision = \frac{\text{D}ocuments\ correctement\ attribués\ \text{à}\ la\ classe_i}{\text{N}ombre\ de\ documents\ \textbf{attribués}\ \text{à}\ la\ classe_i\ par\ le\ système} \tag{4.21}$$

La F-mesure représente la moyenne harmonique du Rappel et de la Précision. Elle mesure la capacité du système à donner toutes les solutions pertinentes et à refuser les autres. Elle se calcule comme suit :

$$\text{F}-mesure = 2 * \frac{\text{R}appel * \text{P}récision}{\text{R}appel + \text{P}réscision} \tag{4.22}$$

4.6.2 Modèles d'apprentissage

Nous avons testé les modèles sur deux corpus de test ; le premier que nous appelons "échantillon de test" (200 R$eview$ et 100 $\overline{\text{R}eview}$) et le deuxième contient tous les documents de Revues.org (31 263 R$eview$ et 71 521 $\overline{\text{R}eview}$). Le tableau 4.6 récapitule toutes les représentations de documents testées sur les deux corpus de test.

Les méthodes de représentation #BoW_XML_TEI et #BoW_NoXML_TEI se basent sur l'approche classique de sac de mots, où dans la première, nous avons gardé les balises XML-TEI présentes dans les documents et dans la seconde, nous les avons retirées des caractéristiques. Les méthodes #FS_tf_id et #FS_Z_norma reposent sur des techniques de réduction de l'espace vectoriel avec une sélection d'un ensemble de mots (les 5 000 premiers) triés respectivement selon leurs valeurs de *tf*idf* et *z-score normalisé*.

TABLEAU 4.6 – Les méthodes de représentation testées sur les deux corpus de test : échantillon de test (corpus 1) et tous les documents de Revues.org (corpus 2)

Descripteurs et représentation	Notation	Corpus 1	Corpus 2
Sac de mots avec les balises XML-TEI	#BoW_XML_TEI	✓	
Sac de mots sans les balises XML-TEI	#BoW_NoXML_TEI	✓	
Sélection des caractéristiques avec *tf*idf*	#FS_tf_idf	✓	✓
Sélection des caractéristiques avec *z-score normalisé*	#FS_Z_norma	✓	✓
Sélection des caractéristiques avec *REF-SVM*	#FS_RFE_SVM	✓	
Répartition des entités nommées	#NED	✓	✓
Combinaison de #NED et #FS_Z_norma	#NED_Z_norma	✓	✓
Combinaison de #NED et #FS_tf_idf	#NED_tf_idf	✓	✓

La méthode #FS_RFE_SVM repose sur l'algorithme RFE-SVM GUYON et collab. [2002] qui est une procédure de sélection décrémentale qui élimine progressivement les attributs de faible poids. Après avoir sélectionné les attributs avec l'algorithme RFE-SVM en utilisant la méthode SVMAttributeEval [17] de Weka [18], nous avons transformé les documents du corpus en descripteurs contenant des caractéristiques de valeurs binaires. Ces caractéristiques représentent les mots sélectionnés par l'algorithme RFE-SVM, et leurs valeurs correspondent à la présence (1) ou l'absence (0) des caractéristiques (mots) dans les documents. Nous obtenions un ensemble réduit de caractéristiques (de 4 783 à 1 199 caractéristiques, réduction de 74,93%).

Le modèle noté #NED utilise la distribution des entités nommées dans le texte qui a donné un ensemble de 30 caractéristiques, 10 pour chaque entité nommée (*Person*, *Date* et *Location*) dont chaque caractéristique est une valeur continue qui indique le taux d'occurrences de l'entité dans chaque partie du texte sachant qu'il est divisé en 10 parties selon le nombre de mots.

Finalement, nous avons effectué une combinaison de caractéristiques dans le but d'évaluer les performances incluant à la fois des caractéristiques lexicales et linguistiques (#FS_Z_norma et #FS_tf_id) et d'autres issues de la répartition des entités nommées dans le texte (#NED).

Vu la quantité des documents du corpus de test (qui inclut toute la plate-forme Revues.org), nous avons opté pour l'algorithme des Machines à Vecteurs de Supports (Support Vector Machine - SVM) avec les deux fonctions de noyau (Radial Basis Function - RBF et linéaire) pour l'apprentissage et la classification avec l'implémentation LIBSVM FAN et collab. [2005]. Les paramètres C et γ (Gamma) du noyau RBF ont été sélectionnés avec la fonction grid de LIBSVM, avec une cross validation (10 plis) sur le corpus d'entraînement.

4.6.3 Résultats

Nous présentons deux tableaux de résultats (4.7, 4.8) qui correspondent respectivement aux deux corpus de test "échantillon de test"et tous les documents de Revues.org

Le tableau 4.7 présente les résultats de l'évaluation du corpus "échantillon de test". Nous

17. Nous avons gardé les valeurs des paramètres par défaut indiqués dans : http://weka.sourceforge.net/doc.stable/weka/attributeSelection/SVMAttributeEval.html

18. http://www.cs.waikato.ac.nz/ml/weka/

avons commencé par évaluer les différents modèles sur un corpus de petite taille (une centaines de documents) pour des raisons de coûts de calcul, les méthodes basées sur le sac de mots ont été testées uniquement avec le corpus “échantillon de test”(plus de 164 000 mots).

TABLEAU 4.7 – Résultats de l’évaluation des performances des modèles de classification en utilisant différents schémas d’indexation sur le corpus “échantillon de test”.

		$Review$			$\overline{Review}$		
#	Modèles	R	P	F-M	R	P	F-M
BoW_XML_TEI	SVM (linéaire)	99.9%	98.0%	99.1%	96.0%	99.1%	98.2%
	SVM (RBF)	99.0%	98.0%	98.5%	96.0%	98.0%	97.0%
	* C = 2.0						
	* γ = 0.00285						
BoW_NoXML_TEI	SVM (linéaire)	100%	97.6%	98.8%	95.0%	100%	97.4%
	SVM (RBF)	97.0%	98.0%	97.5%	96.0%	94.1%	95.0%
	* C = 3.0						
	* γ = 0.00368						
FS_tf_idf	SVM (linéaire)	99.5%	98.0%	98.8%	96.0%	99.0%	97.5%
	SVM (RBF)	97.5%	97.5%	97.3%	95.0%	94.1%	94.5%
	* C = 6.0						
	* γ = 0.00782						
FS_Z_norma	SVM (linéaire)	97.0%	96.5%	96.8%	93.0%	93.9%	93.5%
	SVM (RBF)	97.5%	96.6%	97.0%	93.0%	94.9%	93.9%
	* C = 5.0						
	* γ = 0.00085						
FS_RFE_SVM	SVM (linéaire)	98.0%	97.0%	97.5%	94.0%	95.9%	94.9%
	SVM (RBF)	98.0%	96.1%	97.0%	92.0%	95.8%	93.9%
	* C = 4.0						
	* γ = 0.00071						
NED	SVM (linéaire)	88.0%	77.2%	82.2%	48.0%	66.7%	55.8%
	SVM (RBF)	90.0%	77.3%	83.1%	47.0%	70.1%	56.3%
	* C = 8.0						
	* γ = 0.00851						
NED_Z_norma	SVM (linéaire)	83.0%	79.0%	81.0%	56.0%	62.2%	58.9%
	SVM (RBF)	93.0%	79.1%	85.5%	51.0%	78.5%	61.8%
	* C = 9.0						
	* γ = 0.00418						
NED_tf_idf	SVM (linéaire)	98.0%	97.0%	97.5%	94.0%	95.9%	94.9%
	SVM (RBF)	97.5%	90.7%	94.0%	80.0%	94.1%	86.5%
	* C = 1.0						
	* γ = 0.00921						

Il est clair dans les résultats obtenus avec un corpus de test (tableau 4.7) d’une centaine de documents (“échantillon de test”) que les taux de bonne classification sont particulièrement élevés. Il est même possible d’obtenir plus de 99% de rappel dans les méthodes basées sur les sacs de mots. Ceci montre que la détection des comptes rendus de lecture dans un tel corpus est envisageable avec des techniques purement lexicales. Par contre, les résultats montrent que la suppression des balises XML-TEI de l’ensemble des caractéristiques diminuent légèrement la F-mesure en utilisant les deux fonctions du noyau SVM ce qui signifie que les balises peuvent être considérés comme des caractéristiques discriminantes.

Les performances des méthodes utilisant les techniques de sélection de caractéristiques (#FS_tf_id, #FS_Z_norma et #FS_RFE_SVM) ne sont pas élevées par rapport au système classique basé sur les sacs de mots. Cela est forcement dû aux exemples présents dans le corpus de test qui est peu volumineux. Les documents que l'on trouve sont très bruités (mélange de langues, même au sein d'un seul document) d'autant plus que beaucoup d'entre eux contiennent peu de mots, ce qui fait que la classification pour ces documents n'est pas performante.

Les résultats du système #NED montrent une flagrante diminution des performances de classification surtout pour la classe $\overline{Review}$ (tableau 4.7), ceci est dû à l'étiquetage attribué par TagEN pour les documents du corpus de test. Nous rappelons que ce dernier contient des documents tirés aléatoirement à partir de Revues.org. Ces documents sont de différentes langues (anglais, espagnol, français, portugais) ce qui influence les performance de l'étiquetage de TagEN qui ne traite à la base que la langue française. De plus, dans la classe $\overline{Review}$, on trouve des documents présentant des actualités du domaine des sciences humaines et sociales, des éditoriaux, etc. (voir tableau 4.2). Donc l'utilisation de la répartition des entités nommées pour ces types de documents n'est pas la bonne solution. Ce qui est d'ailleurs justifié par les résultats des combinaisons effectuées avec les modèles #FS_tf_id et #FS_Z_norma où une augmentation des performance est remarquable surtout pour la combinaison avec les caractéristiques sélectionnées par le *tf*idf*.

Dans le tableau 4.8, nous comparons les performances de chaque modèle appris avec les deux fonctions de noyau (RBF et linéaire) en utilisant comme corpus de test toute la plateforme Revues.org qui est, préalablement, manuellement annotée par des experts dans le domaine des sciences humaines et sociales. Nous utilisons les mesures de classification classiques (Rappel, Précision et F-mesure).

Après analyse des résultats de classification, nous avons constaté que les documents dont le contenu est à majorité non francophone (19,15% pour les $\overline{Review}$ et 7,18% pour les $Review$) représentent, dans la plupart des cas, les documents mal classés et cela s'explique par le fait que nous avons un corpus d'apprentissage seulement en français et donc il est très difficile de prédire la classe d'un document dans une autre langue. Une solution possible est de précéder le processus de classification par une reconnaissance de la langue.

D'après les résultats présentés dans le tableau 4.8, suivant les représentations basées sur la sélection de caractéristiques, l'utilisation des scores *tf*idf* donne un résultat plus performant que le Z-score normalisé en terme de F-mesure de la classe $Review$.

Après la combinaison des caractéristiques relatives aux entités nommées (personne, date et location) avec les caractéristiques sélectionnées par le score *tf*idf* et le Z-score normalisé, les performances de classification sont améliorées et nous obtenons un meilleur système de classification (entités nommés et *tf*idf*) atteignant une F-mesure de 72,2% pour la classe $Review$.

De ce fait, nous tirons la constatation suivante : pour une meilleure identification des comptes rendus de lecture, il ne suffit pas de se baser que sur le lexique des textes, l'intégration des entités nommées en considérant leurs positions dans le texte aboutit à de meilleures performances de reconnaissance.

TABLEAU 4.8 – Résultats de l'évaluation des performances des modèles de classification en utilisant différents schémas d'indexation. Les meilleures valeurs de la classe R$e v i e w$ sont notées en gras et celles de la classe $\overline{\text{R}e v i e w}$ sont soulignées

		R$e v i e w$			$\overline{\text{R}e v i e w}$		
#	Modèles	R	P	F-M	R	P	F-M
FS_tf_idf	SVM (linéaire)	66.6%	62.6%	64.3%	82.3%	84.9%	83.6%
	SVM (RBF)	64.3%	63.3%	63.8%	83.7%	84.3%	84.0%
	* C = 7.0						
	* γ = 0.00384						
FS_Z_norma	SVM (linéaire)	42.2%	73.9%	53.7%	93.4%	78.8%	85.4%
	SVM (RBF)	44.5%	72.7%	55.2%	92.7%	79.2%	85.4%
	* C = 7.0						
	* γ = 0.00724						
NED	SVM (linéaire)	73.5%	68.3%	70.8%	85.1%	88.0%	86.5%
	SVM (RBF)	72.1%	66.0%	68.9%	83.8%	87.3%	85.5%
	* C = 5.0						
	* γ = 0.00112						
NED_Z_norma	SVM (linéaire)	56.7%	78.5%	65.9%	93.2%	83.1%	87.8%
	SVM (RBF)	59.0%	80.3%	68.0%	93.6%	83.9%	88.5%
	* C = 6.0						
	* γ = 0.00626						
NED_tf_idf	SVM (linéaire)	69.4%	71.2%	70.3%	87.7%	86.8%	87.2%
	SVM (RBF)	69.8%	74.6%	72.2%	89.6%	87.2%	88.4%
	* C = 4.0						
	* γ = 0.00688						

4.7 Conclusion

Dans ce chapitre, nous avons présenté les techniques du domaine de la classification automatique de texte en genre. Nous avons présenté des méthodes de classification automatique supervisée pour identifier les comptes rendus de lecture au sein du portail OpenEdition. Ce portail contient plusieurs plateformes et nous avons utilisé principalement Revues.org.

Plusieurs méthodes de représentation classique des documents ont été testées (indexation) ainsi que des méthodes que nous avons proposées qui se basent sur la distribution des entités nommées dans le texte. Nous avons montré que la combinaison des caractéristiques issues de la distribution des entités nommées et celles sélectionnées avec la pondération *tf*idf* donne les meilleurs résultats.

En perspectives, nous prévoyons d'appliquer un logiciel tel que BILBO KIM et collab. [2012b] pour relier chaque compte rendu avec son livre et ainsi créer un graphe de documents. La recommandation porte sur ce graphe et intègre l'exploitation des comptes rendus de lecture comme données sociales.

Chapitre 5

Recommandation pour des requêtes longues et complexes

Résumé : Dans ce chapitre, nous présentons la méthodologie générale que nous avons adoptée pour mener la tâche de recommandation de livres pour des requêtes d'utilisateurs complexes écrites en langage naturel. Nous présentons d'une part les données utilisées dans nos expérimentations et d'autre part chacune des méthodes proposées et les résultats obtenus.

Sommaire

5.1 Introduction

Face à l'hétérogénéité des demandes d'information produites par les utilisateurs sur le Web, et plus particulièrement sur les forums de discussion et face à la diversité des utilisateurs, les modèles de recommandation doivent s'adapter.

Les systèmes de recommandation basés sur les contenus fonctionnent bien pour la recommandation de produits sur des boutiques en ligne LEIMSTOLL et STORMER [2007]; SCHAFER et collab. [1999], mais sont difficilement applicables dans le cas de contenus textuels riches et hétérogènes ADOMAVICIUS et TUZHILIN [2005]. C'est le cas notamment des produits où les contenus sont pauvres en textes ou possédant des contenus générés par les utilisateurs eux-mêmes qui sont très hétérogènes (comme les commentaires et les avis). La forte variabilité des caractéristiques en terme de longueur ou de richesse sémantique sont autant de freins à une approche unique. Pour ce type de contenu, il faut utiliser des analyseurs automatiques pour extraire les données nécessaires pour la recommandation et la modéliser de telle sorte d'avoir une base homogène.

Par ailleurs, les systèmes de filtrage collaboratif présentent des items similaires aux profils des utilisateurs. Ces recommandations bien souvent redondantes ont un intérêt limité pour l'utilisateur dans le sens ou par exemple, si un utilisateur a un profil centré sur le domaine de la médecine cela ne doit pas signifier qu'il n'aura besoin que d'informations relatives à ce domaine.

De plus, les systèmes de recommandation collaboratifs sont difficilement utilisables dans le contexte qui nous intéresse : l'ignorance des profils des utilisateurs (nous ne disposons pas de profils utilisateur), présence des besoins des utilisateurs sous forme de requêtes complexes écrites en langage naturel.

Dans ce contexte, notre proposition se matérialise par des modèles de systèmes de recommandation basés sur des modèles de recherche d'information classiques. Il s'agit, dans notre cas, de proposer des recommandations à des utilisateurs en fonction de leurs requêtes complexes qu'on trouve généralement dans les forums de discussion et qui sont adressées à des humains, et non à des machines. Comme hypothèse, nous supposons que l'utilisateur pourra avoir de meilleures recommandations en posant une requête bien précise et détaillée. Ces recommandations reposent en premier sur les contenus textuels des documents d'une part et sur les contenus générés par les utilisateurs d'autre part sans prendre en considération leurs profils. Les modèles de recommandation proposés répondent à un double objectif :

1. améliorer la satisfaction de l'utilisateur en lui proposant des contenus en adéquation avec son besoin ;
2. et augmenter les chances d'obtenir des recommandations pertinentes en élargissant et reformulant les besoins des utilisateurs.

Notre proposition repose sur trois hypothèses :

Hypothèse 1 : La combinaison de différentes approches ne produisant pas le même résultats, améliorent les performances de recommandation.

Hypothèse 2 : L'agrégation des avis des autres utilisateurs dans le processus de recommandation de lecture pour un utilisateur améliore la pertinence des résultats ; autrement dit : une proposition de lecture déjà appréciée et mieux notée par d'autres utilisateurs, peut être la meilleure à recommander.

Hypothèse 3 : Apporter des enrichissements dans la requête initiale de l'utilisateur peut améliorer les performances de recommandation.

Dans ce chapitre, nous définissons en premier, le corpus de test utilisé ainsi que les étapes de prétraitements des documents du corpus. Ensuite, nous exposons le protocole expérimental mis en œuvre pour les différentes méthodes qui correspondent aux hypothèses posées suivi d'une analyse des résultats obtenus.

5.2 Corpus d'évaluation

Notre étude se base sur le corpus utilisé lors de la campagne d'évaluation de la conférence CLEF Social Book Search (SBS) que nous avons décrit dans le chapitre 2 (section 2.5.4). Nous avons présenté la structure des documents qui sont des descriptions de livres issues d'Amazon et enrichies avec des données du site LibraryThing ainsi que la structure des requêtes d'utilisateurs.

Le processus d'évaluation concernant la tâche SBS de cette campagne consiste à :

1. Développer des modèles de recommandation de lecture (livres) pour des requêtes (appelées topics) prédéfinies par les organisateurs de la campagne (crawlées des forums de discussion de LibraryThing). On appelle *run*, la liste des recommandations de chaque modèle développé.
2. Proposer pour chaque topic (requête longue et complexe) 1000 livres dans chaque run soumis.
3. Évaluation de chaque run selon le nDCG@10 et la MAP (voir section 2.4.2).

Le choix de cette collection d'évaluation a été motivé par le fait qu'elle constitue une référence en recommandation et recherche d'information de par sa taille convenable et son origine (le Web). Un autre élément qui a motivé notre choix est que cette collection regroupe plusieurs points en commun avec la collection issue du portail OpenEdition, Revues.org sur laquelle, nous testons nos approches présentées dans le chapitre suivant. Ces points sont : la taille, l'origine, la structuration des documents, l'hétérogénéité des contenus textuels et le plus important le but de la tâche qui est la *recommandation de lecture*.

5.3 Préparation des documents

Comme nous l'avons déjà dit, les documents de la collection CLEF sont au format XML et chaque balise comporte une description du livre comme (nom d'auteur, date de publication, titre, etc.). Nous ne retenons que le contenu de certaines balises pour la phase d'indexation.

— **isbn** : représente l'identifiant du livre ;

— **title** : le titre du livre ;

— **content** : indique différents contenus : contenu de la description du livre, contenu d'une critique et contenu de la description du livre faite par Amazon ;

— **label** : étiquette qui fait référence à l'éditeur du livre ;

— **name** : le nom de l'auteur/les auteurs du livre ;

— **editorialreview** : contient la description du livre, généralement faite par Amazon ;

— **source** : la source de la description du livre (Amazon ou autre) ;

— **summary** : contient le résumé du commentaire d'un utilisateur donné ;

— **browsenode** : Amazon utilise une hiérarchie de nœuds pour organiser les articles mis en vente. Chaque genre et sous-genre est représenté par une valeur de BrowseNode spécifique ;

— **tag** : contient une étiquette attribuée au livre par les utilisateurs d'Amazon.

— **summary** : cette balise contient une phrase résumant l'avis d'un utilisateur pour un livre ;

— **rating** : la note de l'utilisateur ;

— **totalvotes** : le nombre total des commentaires des utilisateurs ;

— **helpfulvotes** : le nombre total des commentaires jugés utiles sur le livre ;

— **similarproduct** : contient l'ISBN d'un produit similaire selon son contenu, les informations d'achat, les appréciations des utilisateurs, etc.

Ce choix des balises est propre à la collection d'INEX, la tâche Social Book Search (conférence CLEF). Pour une autre collection, il faut adapter l'indexation à son schéma et son format de stockage. Deux étapes de pré-traitements des documents avant la phase d'indexation sont effectuées :

1. **L'élimination des mots outils** : dans cette étape les mots d'usage général et grammatical sont éliminés. On distingue deux techniques pour l'élimination des mots outils : l'utilisation des listes de mots outils appelées également anti-dictionnaires et l'utilisation des mesures statistiques.
2. **La racinisation** : dite stemming en anglais. Il s'agit de prendre la forme canonique du mot. Dans un document les mots peuvent apparaître sous différentes formes. Par exemple : cite, citation, citations, etc. La racinisation permet l'obtention d'une forme tronquée du mot, commune à toutes les variantes morphologique. Elle consiste à supprimer les flexions et les suffixes d'un mot. Dans notre étude, nous avons utilisé la méthode de racinisation de *Porter* WILLETT [2006].

Après ces deux étapes, nous devrons indexer le contenu textuel informationnel des documents.

5.4 Modèles de recommandation proposés

L'idée générale des méthodes proposées est de considérer qu'une requête d'utilisateur est l'utilisateur lui même à qui on recommande des items (des livres). Les requêtes sont des descriptions de besoin en recommandation écrites d'une manière longue et complexe ce qui rends le processus de recommandation compliqué. Nous nous sommes basés sur des modèles de RI classiques en intégrant d'autres données pour proposer des recommandations aux utilisateurs qui en demandent dans leurs requêtes. De ce fait, nous avons utilisé un modèle de langue et un modèle probabiliste comme base pour la recommandation que nous détaillerons dans ce qui suit.

5.4.1 Méthode 1 : Combinaison des recommandations issues de plusieurs méthodes

Dans la première méthode de recommandation proposée, nous souhaitons valider l'hypothèse posée dans l'introduction de ce chapitre. Le principe de l'approche est de combiner les résultats de recommandation des deux modèles de recherche choisis qui sont : le modèle

de langue proposé par METZLER et CROFT [2005] Markov Random Field (MRF), Sequential Dependence Model qui considère la proximité entre les termes de la requête dans les documents en estimant les dépendances des termes. Le second modèle utilisé est le modèle probabiliste Divergence From Randomness proposé par AMATI et VAN RIJSBERGEN [2002] qui mesure l'importance d'un terme par rapport à son usage générale dans la collection des documents.

5.4.1.1 A Sequential Dependence Model (SDM)

Le modèle de dépendance séquentielle (ou Sequential Dependence Model, SDM) est un cas particulier du modèle MRF (Markov Random Field) pour la recherche d'information. Il a montré des performances élevées dans plusieurs contextes de recherche ALLAN et collab. [2008]; METZLER et collab. [2006]. Ce modèle n'agit que sur les mots de la requête et consiste à modéliser les dépendances entre les mots adjacents. Suivant le modèle SDM, la fonction $f_{\mathrm{T}}(q_i, \mathrm{D})$ calculant le poids du mot (1-gramme) q_i de la requête q dans un document D est donnée par l'équation :

$$f_{\mathrm{T}}(q, \mathrm{D}) = \log\left[\frac{occ(q, \mathrm{D}) + \mu.\frac{occ(q,\mathrm{C})}{|\mathrm{C}|}}{|\mathrm{D}| + \mu}\right]$$

avec $occ(q, \mathrm{C})$ le nombre d'occurrence du mot de la requête q dans la collection cible C, |C| la taille de la collection et |D| la taille du document D. μ est le paramètre de lissage de Dirichlet que nous avons fixé à 2500 comme recommandé par ZHAI et LAFFERTY [2004] pour les requêtes constituées de mots-clés. C'est ce qui représente l'estimation par maximum de vraisemblance de l'unité lexicale q dans le document D.

Le modèle SDM propose deux fonctions supplémentaires pour deux autres types de dépendances qui agissent sur les bigrammes de la requête. La fonction $f_{\mathrm{O}}(q_i, q_{i+1}, \mathrm{D})$ considère la correspondance exacte de deux mots adjacents de la requête. Elle est dénotée par l'indice O. La seconde, $f_{\mathrm{U}}(q_i, q_{i+1}, \mathrm{D})$, est dénotée par l'indice U et considère la correspondance non ordonnée de deux mots au sein d'une fenêtre de 8 unités lexicales. Les formules des différentes fonctions de pondération sont détaillées dans le tableau 5.1.

TABLEAU 5.1 – Les fonctions de pondération d'un modèle de langue basé sur les mots-clés. $tf_{e,\mathrm{D}}$ est le nombre d'occurrences du mot e dans le document D, $cf_{e,\mathrm{D}}$ est le nombre d'occurrences du mot e dans toute la collection, |D| est la taille du document D, et |C| est la taille de la collection. Enfin, μ est le paramètre de lissage de Dirichlet que nous avons fixé à 2500 comme recommandé par ZHAI et LAFFERTY [2004] pour les requêtes constituées de mots-clés.

Fonction de pondération	Description
$f_{\mathrm{T}}(q_i, \mathrm{D}) = log[\frac{tf_{q_i,\mathrm{D}} + \mu\frac{cf_{q_i}}{\|\mathrm{C}\|}}{\|\mathrm{D}\|+\mu}]$	Poids du mot q_i dans le document D.
$f_{\mathrm{O}}(q_i, q_{i+1}, \mathrm{D}) = log[\frac{tf_{\#1(q_i,q_{i+1}),\mathrm{D}} + \mu\frac{cf_{\#1(q_i,q_{i+1})}}{\|\mathrm{C}\|}}{\|\mathrm{D}\|+\mu}]$	Poids du bigramme de mots q_i et q_{i+1} dans le document D.
$f_{\mathrm{O}}(q_i, q_{i+1}, \mathrm{D}) = log[\frac{tf_{\#uw8(q_i,q_{i+1}),\mathrm{D}} + \mu\frac{cf_{\#uw8(q_i,q_{i+1})}}{\|\mathrm{C}\|}}{\|\mathrm{D}\|+\mu}]$	Poids de la fenêtre de 8 unités lexicales délimitée par les mots q_i et q_{i+1} dans le document D.

Finalement, le score d'appariement requête-document qui utilise les fonctions ci-dessus définies par le modèle de dépendance séquentielle revient à :

$$\text{SDM}(\text{Q},\text{D}) = \lambda_\text{T} \sum_{q \in \text{Q}} f_\text{T}(q,\text{D}) + \lambda_\text{O} \sum_{i=1}^{|\text{Q}|-1} f_\text{O}(q_i, q_{i+1}, \text{D}) + \lambda_\text{U} \sum_{i=1}^{|\text{Q}|-1} f_\text{U}(q_i, q_{i+1}, \text{D})$$

où, λ_T, λ_O et λ_U sont des paramètres libres. Nous avons suivi les recommandations dans BONNEFOY et collab. [2012] et avons fixé ces paramètres comme suit : $\lambda_\text{T} = 0.85$, $\lambda_\text{O} = 0.1$ et $\lambda_\text{U} = 0.05$.

5.4.1.2 InL2 du modèle DFR (Divergence From Randomness)

Le deuxième modèle de recherche utilisé est le modèle InL2, Inverse Document Frequency avec une distribution doublement normalisée selon une loi de Laplace. InL2 appartient à la famille des modèles DFR proposés par AMATI et VAN RIJSBERGEN [2002]. Ces modèles reposent sur le contenu informatif des mots dans un document, une quantité qui est ensuite corrigée par le risque d'accepter chaque terme comme un descripteur d'un document (*first normalization principle*). La renormalisation des fréquences par rapport à la longueur d'un document est le deuxième principe de normalisation utilisé (*second normalization principle*). Il est en fait commun à tous les modèles de RI.

Les modèles DFR sont basés sur l'idée suivante : "Plus la fréquence du terme dans le document diverge de sa fréquence dans toute la collection, plus ce terme a une importance informative dans le document "ROBERTSON et collab. [1980].

Pour ce modèle, le score de pertinence d'un document D pour une requête Q est calculé par la formule suivante :

$$score(\text{Q},\text{D}) = \sum_{t \in \text{Q}} qtw . \frac{1}{tfn+1} (tfn . log \frac{\text{N}+1}{n_t + 0.5})$$

où, qtw est le poids du terme de la requête calculé avec : qtf/gtf_{max} ; qtf est la fréquence du terme ; qtf_{max} est la fréquence maximale des termes de la requête. N est le nombre de documents dans la collection et n_t représente le nombre de document contenant le terme t. tfn correspond à la version normalisée de la fréquence du terme tf suivant l'équation suivante :

$$tfn = tf . log(1 + c . \frac{avg\_l}{l}),\ (c > 0)$$

où, tf représente la fréquence du terme t dans le document D ; l est la longueur du document en nombre de token et avg_l est la longueur moyenne de tous les documents ; c est un paramètre qui contrôle la normalisation appliquée à la fréquence du terme en respectant la longueur du document.

5.4.1.3 Protocole expérimental

Le protocole expérimental retenu est celui proposé par LEE [1997]. Il consiste à comparer les listes de documents obtenues par différents systèmes pour une même requête. Le protocole repose sur l'utilisation des fichiers de résultats ou runs.

L'ensemble des résultats sont comparés entre eux pour chacune des requêtes. Cette comparaison s'effectue à l'aide de la mesure de chevauchement (*overlap*) proposée par LEE [1997]. Cette mesure permet d'évaluer la ressemblance entre deux listes de résultats. Il s'agit de calculer la proportion de documents communs parmi l'ensemble des documents retournés par les deux runs. Elle est définie comme suit :

$$overlap_k = \frac{Card(run1_k \cap run2_k) * 2}{Card(run1_k) + Card(run2_k)}$$

où :

— k est le nombre de documents considérés ;

— et $run1_k$ (respectivement $run2_k$) est le sous-ensemble de k premiers documents du $run1_k$ (respectivement $run2_k$).

Le chevauchement *overlap* est compris entre 0 et 1. Il vaut 1 si $run1_k$ et $run2_k$ sont identiques, et 0 si $run1_k$ et $run2_k$ n'ont aucun élément en commun.

Afin de montrer que deux approches différentes conduisent effectivement à des ensembles de documents différents, nous calculons la proportion d'éléments communs (taux de chevauchement) pour la paire des runs et pour chaque requête. Cette mesure est tout d'abord appliquée à l'ensemble des documents retournés par chaque système (1000 documents). Ensuite, nous nous intéressons aux taux de chevauchement des documents pertinents et des documents non-pertinents pour chaque requête.

Nous comparons les résultats des runs correspondant au modèle de langue SDM et au modèle probabiliste InL2 de DFR appliqués à la collection de données de CLEF, la tâche SBS pour les années 2014 et 2015. Le choix des runs est motivé par le fait qu'ils ont obtenus des résultats concurrents avec 0.3584 pour le run SDM (classé deuxième en 2012) et 0.101 pour le run INL2 (classé quatrième en 2014) en terme de Mean Reciprocal Rank (MRR) KOOLEN et collab. [2014, 2012].

Les résultats présentés dans le tableau 5.2 correspondent à la moyenne des mesures de chevauchement effectuées pour les deux runs. Nous avons dans un premier temps calculé cette mesure en considérant les documents de manière globale, puis en nous focalisant uniquement sur les documents pertinents et les documents non-pertinents restitués.

TABLEAU 5.2 – chevauchement moyen entre les runs SDM et INL2 en considérant 1000 documents

Année	Documents considérés	Chevauchement
CLEF SBS 2014	Global	26.10%
	Pertinents	47.49%
	Non-pertinents	26.67%
CLEF SBS 2015	Global	22.86%
	Pertinents	45.30%
	Non-pertinents	18.40%

Les résultats présentés (tableau 5.2) montrent que les deux méthodes partagent peu de documents (global ne dépassant pas 27% pour les deux années). En revanche le nombre de documents pertinents communs entre les deux runs n'est pas négligeable et atteint presque la moitié des documents communs entre les runs. Les deux runs donnent un ensemble de documents non-pertinents très différents (26% et 18%). Ces résultats valident en partie la première hypothèse posée qui dit que deux approches de recherche différentes ne produisent pas le même résultat.

Pour valider en totalité la première hypothèse, nous avons combiné les sorties des deux modèles de recherche. Le but est de produire des recommandations issues de deux techniques différentes et de montrer que les performances de la combinaison sont plus élevées que celles de chaque technique séparément. BELKIN et collab. [1995] ont combiné les résultats d'un

modèle probabiliste et d'un modèle à espace de vecteurs et ont abouti à des performances améliorées.

Chacun des modèles donne une liste de documents avec des scores. Ces derniers sont issus de deux schémas de pondération différents et donc doivent être normalisés avant la phase de fusion des recommandations. Pour cela, nous avons utilisé la formule normalisation de Lee Lee [1995] comme suit :

$$normalizedScore = \frac{oldScore - minScore}{maxScore - minScore}$$

Après normalisation des scores, nous avons utilisé la combinaison linéaire des scores selon l'équation suivante :

$$combinedScore(Q,D) = \alpha.(scoreInL2(Q,D)) + (1-\alpha).(scoreSDM(Q,D))$$

Où α est un paramètre d'interpolation = 0,8 que nous avons fait varier pour obtenir la meilleur interpolation donnant des résultats les plus performants (variations effectuées sur les données de CLEF SBS'2014).

5.4.2 Méthode 2 : Agrégation des données sociales pour le ré-ordonnancement des recommandations

Le contexte de recommandation étudié dans cette thèse suit un processus de recherche perçu comme une activité individuelle menée par un unique utilisateur en vue de satisfaire son besoin personnel en information (requête). Cependant, plusieurs situations de recommandation gagneraient en efficacité si elles intégraient des informations partagées par des utilisateurs Shah [2014]. Une des problématiques de tels systèmes est d'introduire le paradigme de la collaboration dans le ré-ordonnancement des documents.

Pour remédier à cette problématique, nous proposons une méthode de ré-ordonnancement des recommandations en intégrant des informations sociales. Pour cela, nous calculons un score social (*Likeliness*) des documents en utilisant les jugements donnés par les utilisateurs. L'idée est de supposer qu'un document qui a été beaucoup et bien noté par les utilisateurs, peut être le plus pertinent à mettre en avant dans la liste des recommandations. Nous avons utilisé la formule suivante pour calculer le score *Likeliness* :

$$Likeliness(D) = \frac{\sum_{r \in R_D} r}{|Reviews_D|}$$

où, R_D est l'ensemble de tous les votes donnés par les utilisateurs pour le document D et $|Reviews_D|$ est le nombres de votes.

Dans la collection de test, on trouve un type particulier de vote. Il ne s'agit pas de noter seulement les livres mais de pouvoir également noter les votes et les commentaires des utilisateurs. On a la possibilité de juger utile ou non un commentaire (vote binaire). De ce fait, on a 3 types de données sociales (comme le montre la figure 5.1) : les votes (*rating*), les votes d'utilité (*helpful votes*) et le nombre total de votes sur les commentaires (*total votes*).

Nous avons exploité les informations d'utilité pour pondérer chaque vote. Le score obtenu appelé *Usefulness* se calcule comme suit :

$$Usefulness(D) = \frac{\sum_{r \in R_D, t \in T_D, h \in H_D} r * (\frac{t}{h})}{|Reviews_D|}$$

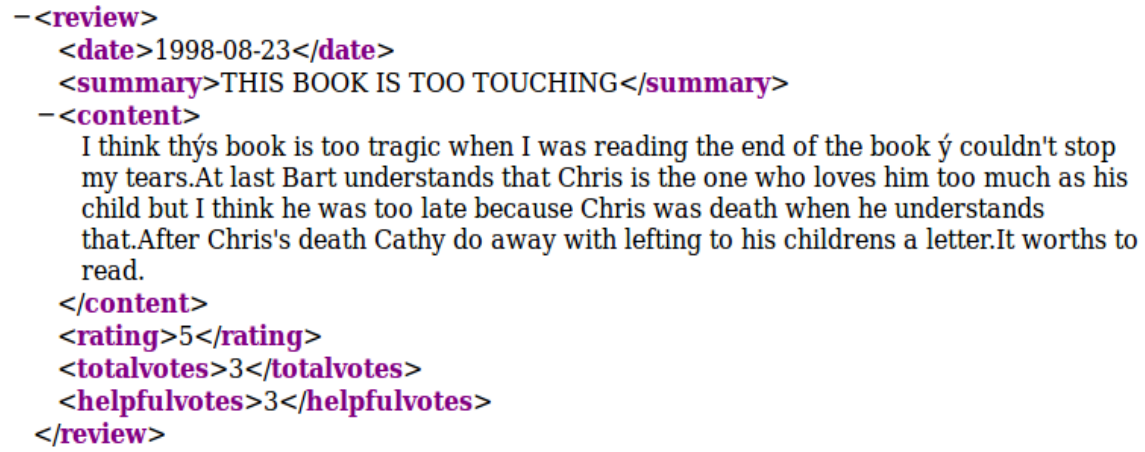


FIGURE 5.1 – Exemple de commentaire d'un utilisateur avec *rating*, *helpful votes* et un *total votes*

où, R_D, T_D et H_D sont respectivement, les ensembles de ratings, helpful votes et total votes donnés pour le livre D.

Nous avons donc deux scores sociaux que nous utiliserons pour le ré-ordonnancement des recommandations après une étape de recherche. Nous combinons les scores avec un paramètre d'interpolation fixé à $0,85$ pour la combinaison avec le score *Likeliness* et à $0,93$ pour la combinaison avec le score *Usefulness* après plusieurs variations effectuées sur les données de CLEF SBS'2012.

5.4.3 Méthode 3 : Reformulation des requêtes par réinjection de pertinence (Pseudo Relevance Feedback)

L'utilisateur formule son besoin en information par une requête composée de ses propres mots clés. Le choix des termes influence directement sur l'ensemble des documents restitués par le système. Il arrive que l'utilisateur utilise des termes qui ne correspondent pas forcément à ceux utilisés pour indexer les documents pertinents.

Le processus de reformulation de requêtes, comme nous l'avons décrit précédemment (chapitre 2, section 2.3), consiste à générer une nouvelle requête plus adéquate que celle initialement formulée par l'utilisateur. L'objectif est de limiter le bruit et le silence dus à un mauvais choix des mots.

Nous proposons une méthode de reformulation de requêtes qui repose sur une expansion directe en rajoutant des mots issus de la collection de documents utilisée. L'architecture générale du processus de réinjection est présentée dans la figure 5.2.

Le point de départ du processus de reformulation est la liste des documents restitués par le système de recherche de base. La démarche que nous adoptons pour reformuler une requête est composée essentiellement de 3 étapes :

1. **Échantillonnage :** Dans cette étape, nous construisons un échantillon d'éléments à partir des résultats donnés par le système de base. Un échantillon se caractérise par sa taille en nombre d'éléments et par le nombre d'éléments pertinents qu'il contient.
2. **Extraction des termes :** Dans cette étape, nous extrayons des informations pour les réinjecter dans la requête initiale. Nous considérons deux types d'information à extraire à partir de l'échantillon défini dans l'étape précédente :
   — **Information descriptive** : il s'agit de sélectionner des termes informatifs et importants dans le contenu descriptif du document. Nous avons utilisé un mécanisme généralisé de la méthode de Rocchio ROCCHIO [1971] appelé le *Pseudo*

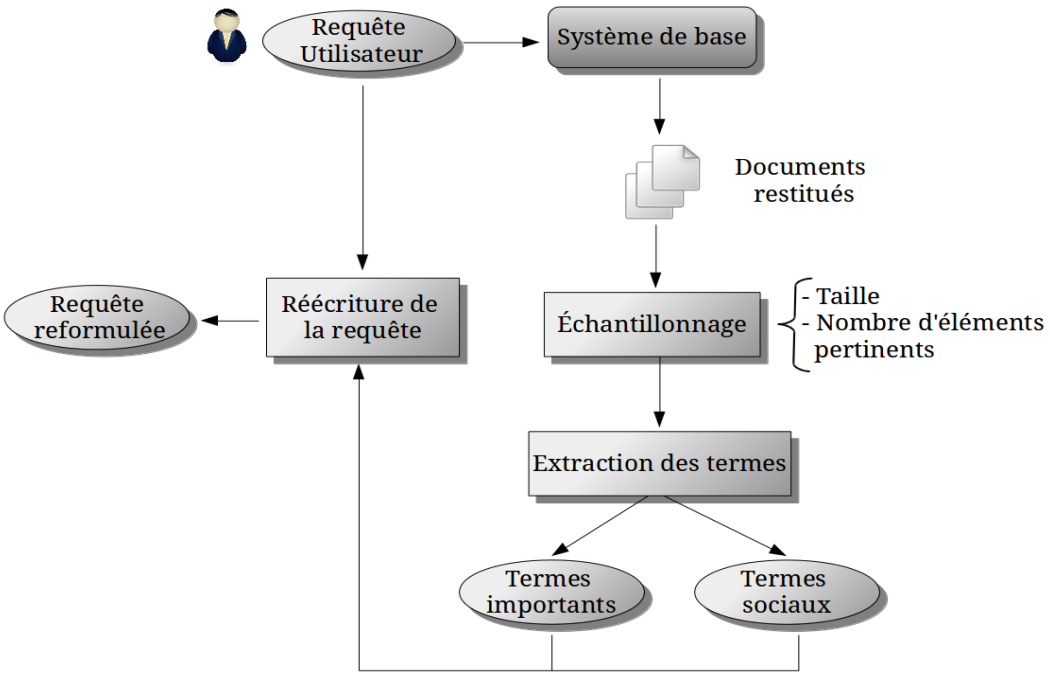


FIGURE 5.2 – Architecture du processus de reformulation de requête

Retour de Pertinence. Le modèle de pondération des termes de l'échantillon utilisé est le modèle *Bo1*. Il est basé sur les statistiques de *Bose-Einstein* AMATI et VAN RIJSBERGEN [2002]. Un classement des termes est effectué selon le poids calculé avec le modèle Bo1. Plus ce poids est élevé plus le terme est informatif dans le texte.

— **Information sociale** : Dans ce cas, nous avons considéré que l'information apportée par les utilisateurs sur les documents, peut exprimer le besoin en information pour d'autres utilisateurs. Un document qui décrit un produit ne contient pas forcement toutes les informations qu'un utilisateur souhaite voir, par exemple, Amazon propose un système d'étiquetage où, les utilisateurs peuvent poser une(des) étiquette(s) ou tag(s) sur les produits qui leurs intéressent pour suggérer une organisation différente et apporter plus de description aux produits. Nous avons exploité ce type d'information pour la phase d'extraction des termes. Nous avons collecté tous les tags attribués par plus de 3 utilisateurs sur les documents de l'échantillon.

3. **Réécriture de la requête :** Cette étape permet de réinjecter les termes extraits aux termes de la requête initiale pour aboutir à une nouvelle requête qu'on utilisera dans le système de recherche. Dans cette étape, aucune pondération n'a été faite sur les termes. Nous avons considéré que tous les termes de la requête composée ont le même poids.

   Prenons l'exemple du topic N° 1116, son contenu est le suivant :

```
<topic id="1116">
<title>Which LISP?</title>
<mediated_query>
  introduction book to Lisp
</mediated_query>
<group>
  Purely Programmers
```

```
</group>
<narrative>
  It'll be time for me to shake things up and learn a
     new language soon. I had started on Erlang a
     while back and getting back to it might be fun.
     But I'm starting to lean toward Lisp--probably
     Common Lisp rather than Scheme. Anyone care to
     recommend a good first Lisp book? Would I be
     crazy to hope that there's one out there with an
     emphasis on using Lisp in a web development and/
     or system administration context? Not that I'm
     unhappy with PHP and Perl, but the best way for
     me to find the time to learn a new language is to
      use it for my work...
</narrative>
</topic>
```

L'utilisateur demande des recommandations de lecture pour apprendre le langage Lisp, et il a précisé qu'il s'intéresse plus particulièrement au développement Web à l'administration système. En appliquant le processus d'extraction des tags par rapport à cette requête, nous avons obtenu la liste des tags suivants : *artificial intelligence*, *Computing*, *Computers*, *non-fiction*, *ai*, *Reference*, *computer science*, *programming*, *programming languages*, *lisp*, *commonlisp*, *cs*, *wishlist*

D'apès ces tags, la requête initiale sera enrichie avec des informations complémentaires, telles que les mots clés *programming*, *artificial intelligence*, *computing*. Ces mots clés restent dans le même contexte du besoin de l'utilisateur mais proviennent d'autres utilisateurs.

Suite aux deux méthodes utilisées pour l'extraction des termes, nous obtenons deux méthodes de réinjection de pertinence dans la requête d'utilisateur. Nous montrons les résultats de ces deux méthodes dans la section suivante.

5.5 Expérimentations et résultats

Dans cette section nous présentons les résultats des différentes expérimentations effectuées sur les données fournies par la conférence CLEF pour la tâche Social Book Search pour l'année 2014 et 2015. Pour rappel, nous disposons d'une collection de 2,8 millions de livres et un ensemble de 680 requêtes fournies en 2014 et 208 requêtes fournies en 2015. Ces requêtes sont écrites en langage naturel par des utilisateurs et postées sur le forum de discussion du site LibraryThing.

Nous avons utilisé deux baselines, la première se base sur le modèle probabiliste InL2 et la deuxième sur le modèle de langue SDM. Nous avons utilisé deux outils :

— *Terrier*[1] (TERabyte RetrIEveR) qui est un framework de RI développé au sein de l'université de Glasgow Ounis et collab. [2006, 2005, 2007]. Il offre des fonctionnalités d'indexation et de recherche et est basée sur un framework DFR que nous utilisons pour déployer le modèle Inl2.

1. http://ir.dcs.gla.ac.uk/terrier,http://terrier.org/

— Indri de Lemur (qui implémente SDM) est un autre outil qui offre presque les mêmes fonctionnalités que Terrier, que nous avons utilisé pour l'implémentation du modèle SDM.

La collection CLEF SBS est sous format XML, un format qui n'est pas supporté par Terrier, principalement pour la phase d'indexation des documents. En conséquence, une étape de conversion des documents en format Trec Collection Format [2] est effectuée en considérant que le contenu de toutes les balises XML dans chaque fichier (description de livre) comme un unique champ dans la version convertie avec comme identifiant l'ISBN du livre.

Pour valider les trois hypothèses posées, nous avons effectué plusieurs expérimentations sur l'ensemble des topics de 2014. Dans la première hypothèse, nous avons supposé que la combinaison de plusieurs approches améliorait les performances. Les listes de recommandations des deux modèles de RI (SDM et InL2) ont été fusionnées. Ensuite, un réordonnancement est fait selon les scores issus de la combinaison linéaire des scores des deux modèles (section 5.4.1.3). La figure 5.3 montre les résultats de variations du paramètre d'interpolation selon la MAP.

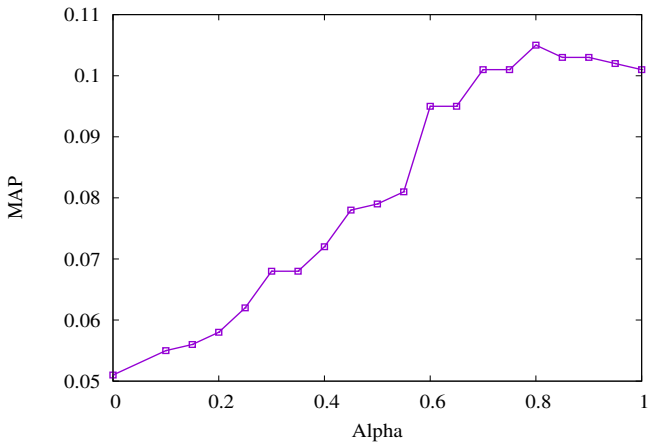


FIGURE 5.3 – Résultats des variations du paramètre d'interpolation α pour la combinaison des modèles SDM et InL2 selon la mesure MAP. (CLEF SBS 2014)

Dans la deuxième hypothèse, nous avons supposé que l'agrégation des avis des utilisateurs, améliorait les performances de recommandation. Il s'agit dans ce cas d'expérimenter le processus de réordonnancement. Le score des modèles de RI sont combinés avec d'autres scores sociaux (Likeliness et Usefulness). Une combinaison avec pondération a été testée pour chaqun des scores. La figure 5.4 montre, la différence entre les scores initiaux du modèle SDM et ceux après pondération avec les scores Likeliness et Usefulness pour la liste des 50 premières recommandations du topic N°1116. Les scores ont été normalisées selon la fonction suivante : $f(x) = (x - min)/(max - min)$.

Les courbes illustrent l'influence des votes des utilisateurs sur l'ordonnancement des livres.

Une autre méthode de combinaison a été testée qui favorise un score par rapport à un autre en utilisant un paramétre d'interpolation. La figure 5.5 montre la différence des scores de réordonnancement après une combinaison linéaire avec les scores *Likeliness* et *Usefulness* pour le modèle SDM. Pour le topic 1116 que nous considérons comme topic représentatif

2. http://lab.hypotheses.org/1129

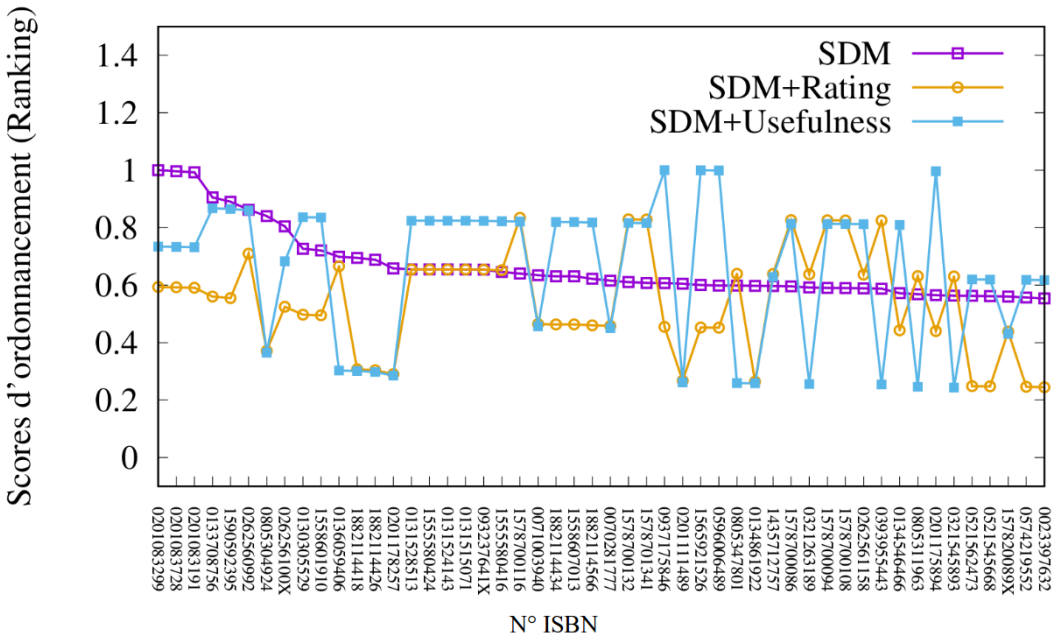


FIGURE 5.4 – Influence des scores sociaux sur les scores du modèle SDM par une simple pondération. Échantillon des 50 premières recommandations pour le topic N 1116.

des requêtes longues et complexes, nous constatons que le modèle SDM a restitué pour les premières recommandations, des livres appréciés et bien notés par les utilisateurs contrairement au reste de la liste des recommandations.

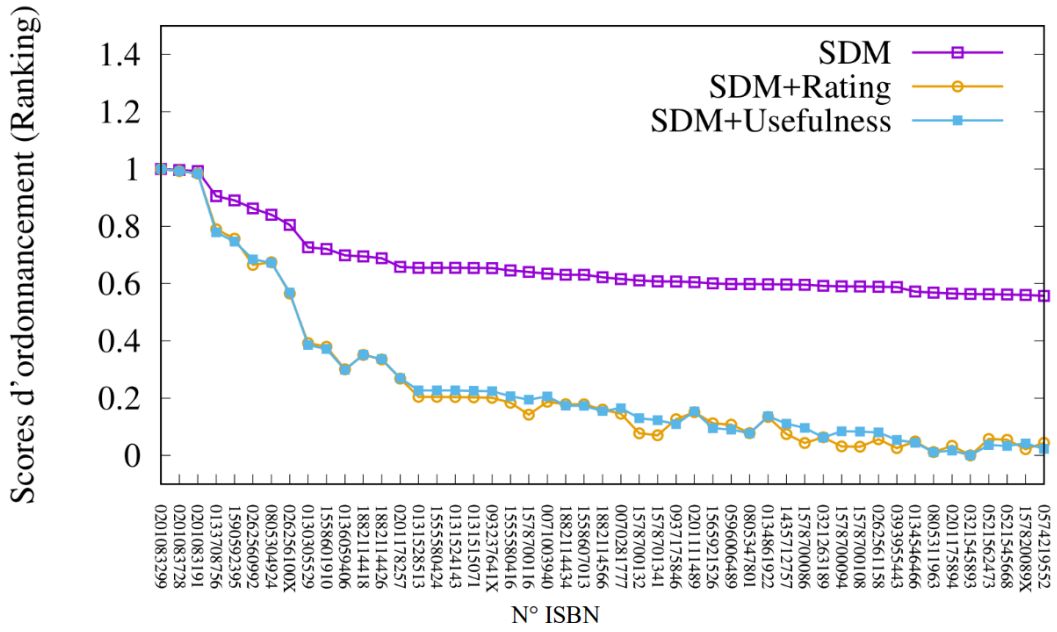


FIGURE 5.5 – Influence des scores sociaux sur les scores du modèle SDM par combinaison linéaire. Échantillon des 50 premières recommandations pour le topic N 1116.

Les figure 5.6 et 5.7 illustrent les valeurs de la nDCG10 en variant le α pour chacun des modèles de RI et des scores sociaux. À partir de ces courbes, nous avons fixés les meilleurs valeurs des paramètres d'interpolation. Nous constatons que les scores du modèle SDM combinés avec *Likeliness* donnent de meilleurs résultats par rapport au score *Usefulness*. Ce

comportement est inversé pour la combinaison avec le modèle InL2. Cependant, la meilleure valeur de nDCG10 est obtenue avec le *Usefulness*.

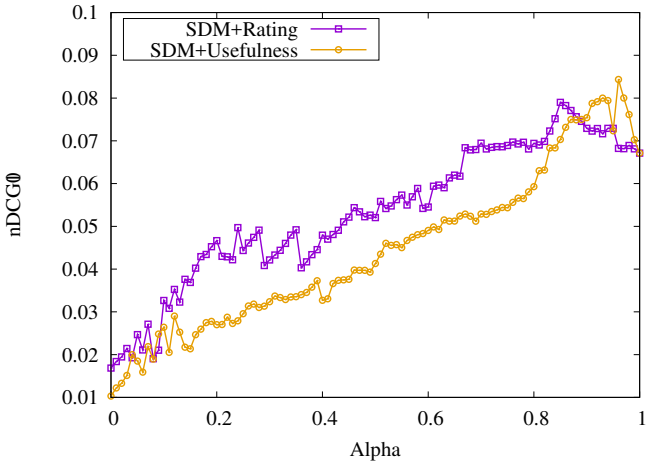


FIGURE 5.6 – Résultats des variations du paramètre d'interpolation α pour la combinaison des scores sociaux avec les score du modèle SDM, selon la mesure nDCG10. (CLEF SBS 2014)

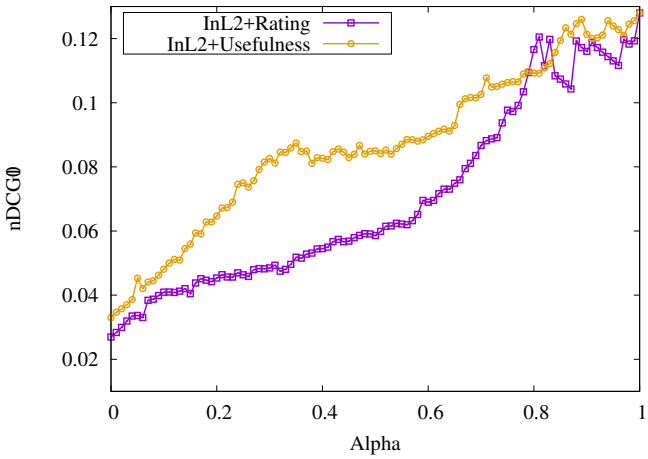


FIGURE 5.7 – Résultats des variations du paramètre d'interpolation α pour la combinaison des scores sociaux avec les score du modèle InL2, selon la mesure nDCG10. (CLEF SBS 2014)

Nous avons supposé dans la troisième hypothèse, que l'enrichissement des requêtes des utilisateurs peut améliorer les performances de recommandation. Pour valider cette hypothèse, nous avons intégré des termes extraits à partir des documents restitués par les modèles de RI (SDM et InL2). Cette méthode a conduit à configurer deux paramètres pour avoir les meilleurs résultats. Il s'agit du nombre de documents à partir lesquels les termes sont extraits (*nbr_doc*) et le nombre de termes dans chaque document (*nbr_term*). Pour les termes informatifs les figures 5.9 et 5.8 montrent les valeurs de la MAP des variations des deux paramètres *nbr_doc* et *nbr_term*. Nous remarquons que plus le *nbr_doc* et le *nbr_term* sont grands plus nous obtenons une dégradation des résultats. Ceci s'explique par la fait que si on rajoute trop de mots, on augmente le risque d'une déviation thématique. Le *nbr_doc* influence également sur les performances puisque les documents sont classés par degré de pertinence, ce qui fait que les documents moins classés ne contiendront par des termes pertinents pour l'enrichissement.

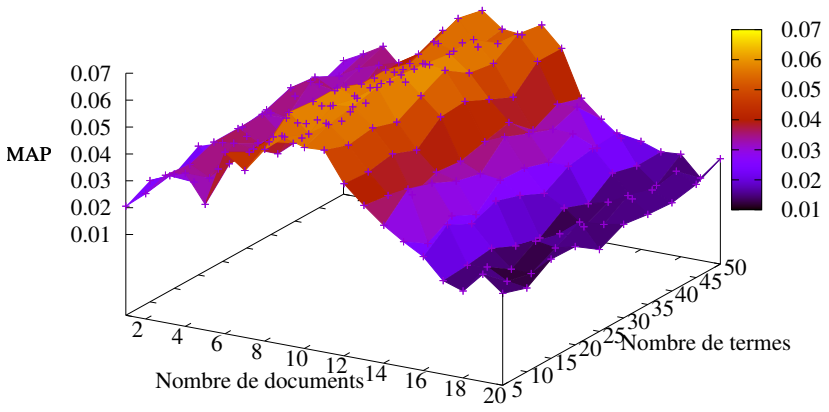


FIGURE 5.8 – Résultats des variations des paramètres *nbr_doc* et *nbr_term* pour la méthode de recommandation utilisant l'enrichissement des requêtes avec les termes informatifs selon la MAP. Modèle de base SDM. (CLEF SBS 2014)

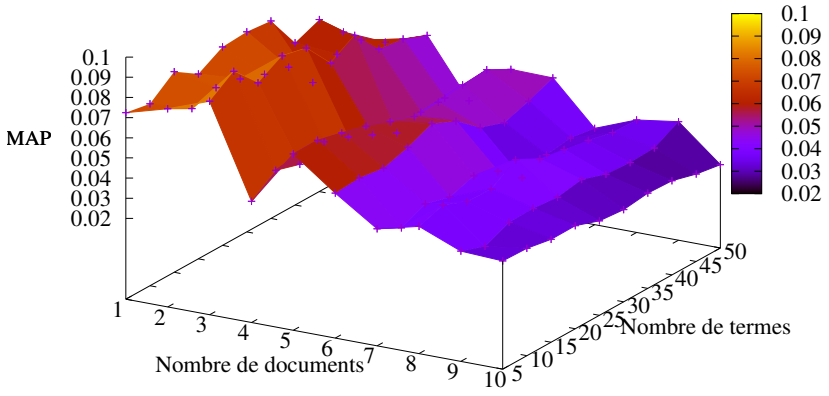


FIGURE 5.9 – Résultats des variations des paramètres *nbr_doc* et *nbr_term* pour la méthode de recommandation utilisant l'enrichissement des requêtes avec les termes informatifs selon la MAP. Modèle de base InL2. (CLEF SBS 2014)

Pour le deuxième type de termes d'enrichissement, nous avons utilisé les tags attribués par plus de 3 utilisateurs. Dans ce cas, seul le *nbr_doc* doit être configuré. La figure 5.10, montre que la meilleure valeur de MAP est obtenue avec un *nbr_doc* = 10 pour les deux modèles (SDM et Inl2).

Pour résumer, voici la liste des runs avec les valeurs des paramètres :

— **InL2** : la première baseline qui implémente le modèle InL2.

— **SDM** : la deuxième baseline qui implémente le modèle SDM.

— **InL2+SDM** : la combinaison linéaire des résultats des deux baselines ($\alpha = 0,8$).

— **InL2+Rating** : application du réordonnancement basé sur les votes (sur la baseline

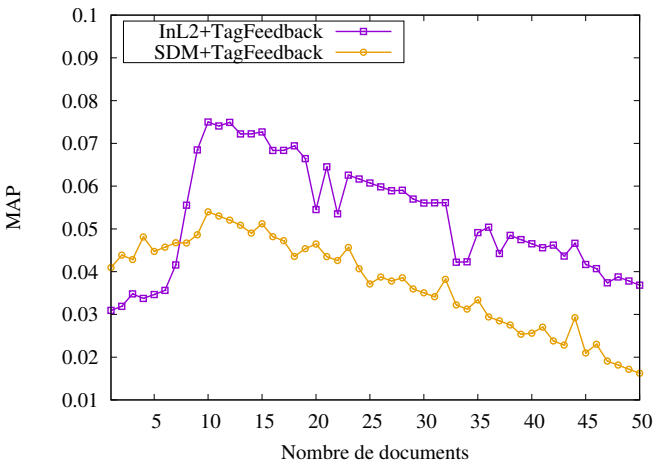


FIGURE 5.10 – Résultats des variations du paramètre *nbr_doc* pour la méthode de recommandation utilisant l'enrichissement des requêtes avec les tags des utilisateurs selon la MAP. Modèles de base SDM et InL2. (CLEF SBS 2014)

InL2). Interpolation linéaire des scores du modèle et les scores Likeliness avec un $\alpha = 0,81$.

— **SDM+Rating** : application du réordonnancement basé sur les votes (sur la baseline SDM). Interpolation linéaire des scores du modèle et les scores Likeliness avec un $\alpha = 0,85$.

— **InL2+Usefulness** : application du réordonnancement basé sur les scores de Usefulness (sur la baseline InL2). Interpolation linéaire des scores du modèle et les scores Usefulness avec un $\alpha = 0,89$.

— **SDM+Usefulness** : application du réordonnancement basé sur les scores de Usefulness (sur la baseline SDM). Interpolation linéaire des scores du modèle et les scores Usefulness avec un $\alpha = 0,93$.

— **InL2+Feedback** : réinjection de pertinence avec les informations descriptives (sur la baseline InL2). Chaque requête a été étendue avec les 10 premiers mots les plus informatifs dans les 3 premiers documents retournés par la première phase de recherche (échantillon).

— **SDM+Feedback** : réinjection de pertinence avec les informations descriptives (sur la baseline SDM). Chaque requête a été étendue avec les 20 premiers mots de l'échantillon composé de 10 documents).

— **InL2+TagFeedback** : réinjection de pertinence avec les informations sociales (sur la baseline InL2). Chaque requête est étendue avec les tags attribués par plus de 3 utilisateurs dans les 10 documents qui composent l'échantillon.

— **SDM+TagFeedback** : réinjection de pertinence avec les informations sociale (sur la baseline SDM). Chaque requête est étendue avec les tags attribués par plus de 3 utilisateurs dans les 10 documents qui composent l'échantillon.

Dans le tableau 5.3 sont présentés les résultats des différents runs appliqués sur les topics de 2014 et de 2015.

Le premier résultat intéressant que l'on observe, pour les expérimentations de 2014 et de 2015 est que la combinaison des deux approches de recherche InL2 et SDM apportent la meilleure amélioration de toutes les mesures d'évaluation comparées aux résultats des

TABLEAU 5.3 – Résultats des expérimentations sur la collection des livres (CLEF, la tâche SBS) et les topics de 2014 et 2015. Les lignes en gris représentent les baselines, celles en jaune représentent la combinaison des deux baselines. (∗) dénote les résultats significatifs selon le test de Wilcoxon CROFT [1978] avec deux faces p-valeur, $\sigma = 0,05$.

	Runs	nDCG@10	Recip Rank	MAP	Recall@1000
2014	InL2	**0.128**	**0.236**	**0.101**	**0.441**
	InL2+Rating	0.120(−6.2%*)	0.227(−3.8%*)	0.095(−5.9%)	0.441(0%*)
	InL2+Usefulness	0.126(−1.5%*)	0.236(0%*)	0.099(−1.9%*)	0.441(0%*)
	InL2+Feedback	0.114(−10.9%)	0.230(−2.5%)	0.094(−6.9%*)	0.434(−1.5%*)
	InL2+TagFeedback	0.102(−20.3%)	0.212(−10.1%*)	0.075(−25.7%)	0.388(−12.0%*)
	SDM	0.067	0.125	0.051	0.325
	SDM+Rating	0.079(+17.9%*)	0.151(+20.8%*)	0.060(+17.6%*)	0.354(+8.9%*)
	SDM+Usefulness	0.080(+19.4%*)	0.152(+21.6%)	0.061(+19.6%*)	0.354(+8.9%*)
	SDM+Feedback	**0.083** (+23.8%*)	**0.157**(+25.6%*)	**0.062**(+21.5%*)	**0.356** (+9.5%*)
	SDM+TagFeedback	0.073(+8.9%*)	0.149(+19.2%)	0.054(+5.8%*)	0.313(−3.6%*)
	InL2+SDM	**0.132**(+3.1%*)	**0.249**(+5.5%*)	**0.105**(+3.9%*)	**0.446**(+1.1%*)
2015	InL2	**0.070**	**0.154**	**0.052**	**0.387**
	InL2+Rating	0.065(−7.1%)*	0.151(−1.9%)	0.049(−5.7%*)	0.387(0%)
	InL2+Usefulness	0.068(−2.8%*)	0.153(−0.6%)	0.051(−1.9%)	0.387(0%)
	InL2+Feedback	0.061(−12.8%*)	0.149(−3.2)%*)	0.051(−1.9%)	0.371(−4.1%*)
	InL2+TagFeedback	0.059(−15.7%*)	0.128(−16.8%*)	0.048(−7.6%*)	0.368(−4.9%)
	SDM	**0.068**	0.154	0.049	0.323
	SDM+Rating	0.066(−2.9%)	0.154(0%*)	0.047(−4%*)	0.323(0%)
	SDM+Usefulness	0.067(−1.4%*)	0.160(+3.8%)	0.049(0%*)	0.323(0%*)
	SDM+Feedback	0.067(−1.4%*)	0.160(+3.8%*)	0.050(+2%)	**0.333** (+3%*)
	SDM+TagFeedback	0.067(−1.4%)	**0.167** (+8.4%*)	**0.051** (+4%*)	0.312(−3.4%*)
	InL2+SDM	**0.077** (+10%*)	**0.176**(+14.2%*)	0.058(+11.5%*)	0.396(+2.3%*)

autres méthodes. Cette amélioration est due au regroupement des documents pertinents non communs que les deux approches ont retourné puisque la combinaison porte sur les recommandations retournées et non sur les modèles eux-mêmes.

Nous remarquons également que nous obtenons deux comportements différents pour le modèle SDM et le modèle InL2. En effet, on constate des améliorations de la baseline SDM et aucune amélioration de la baseline InL2.

Pour les expérimentations de 2014, on constate que toutes les méthodes proposées améliorent les résultats de la baseline SDM plus particulièrement la méthode de réinjection de pertinence avec les mots informatifs qui a donné les meilleures performances (+23,88% en terme de nDCG@10). Ce qui signifie que l'ajout des informations sociales (les votes et les tags) au modèle de langue SDM a un impact très positif dans le processus de recommandation. En revanche, on ne constate pas d'amélioration spécifique de la baseline InL2. Les méthodes proposées baissent légèrement les performances du modèle InL2 ce qui implique que l'intégration des informations sociales dans un processus basé sur le modèle probabiliste (InL2) n'aide pas à proposer de meilleures recommandations aux utilisateurs.

Pour les expérimentations de 2015, on remarque le même comportement pour la baseline SDM. Ses performances ont été améliorées de la même façon que pour l'année 2014 sauf que cette amélioration n'est pas aussi élevée (avec un maximum de 3,9% en terme de MAP, 8,4% en terme de Recip Rank et 3% en terme de Recall@1000) sachant que la méthode de réinjection de pertinence avec les tags des utilisateurs a apporté la meilleure amélioration

en terme de Recip Rank et MAP. Cela montre l'intérêt de la prise en compte des données inscrites par les utilisateurs dans le processus de recommandation basé sur le modèle SDM. D'autre part, on remarque le même comportement pour la baseline InL2 que celui de l'année 2014. La considération des données sociales n'améliore pas les performances de la recommandation basée sur le modèle probabiliste InL2.

La différence des résultats entre les données de 2014 et de 2015 s'explique par le changement dans la méthode d'évaluation pour l'année 2015. Ce changement réside dans l'attribution des valeurs de pertinence aux différentes suggestions pour chaque topic. Nous avons détaillé ce point dans le chapitre suivant (voir la section 6.2.5.3).

5.6 Conclusion

Dans ce chapitre, nous avons proposé et évalué trois méthodes de recommandation basées sur des modèles de recherche d'information. La première méthode repose sur la combinaison d'un modèle de langue et d'un modèle probabiliste en suivant le protocole expérimental proposé par LEE [1997]. Dans la deuxième méthode, nous avons intégré les données produites par les utilisateurs dans le processus de recommandation en supposant qu'une recommandation basée sur l'avis d'autres utilisateurs sera plus pertinente qu'une recommandation basée seulement sur le contenu. La dernière méthode repose sur la transformation de la requête initiale de l'utilisateur. Cette transformation est faite par une agrégation d'un ensemble de termes à partir d'un échantillon de documents retournés par un système de recommandation de base. Cet ensemble de termes représente deux types de données : des tags d'utilisateurs ou les termes les plus informatifs selon une fonction Bo1 (statistiques de *Bose-Einstein*).

Via les résultats des différentes expérimentations faites sur le corpus de CLEF la tâche Social Book Search (2014 et 2015), nous validons les trois hypothèses posées dans l'introduction de ce chapitre (voir section 5.1). Nous avons montré que la combinaison de différentes approches de recherche ne produisant pas les mêmes résultats améliorent les performances de recommandation. Nous avons également montré que l'agrégation des données sociales et la reformulation de requête par injection de pertinence peuvent engendrer des recommandations plus pertinentes par rapport à une base de référence basée sur le modèle de langue SDM. Cependant nos méthodes montrent une dégradation des performances par rapport à une base d'évaluation de référence basée sur le modèle probabiliste InL2. Il faudrait peut être explorer une meilleure façon d'agrégation ou une meilleure méthode de reformulation en utilisant des ressources externes à la collection pour améliorer les performances de recommandation.

Dans le chapitre suivant, nous modélisons les données en structure de graphe. Nous présentons une nouvelle méthode de recommandation qui combine des modèles de recherche d'information et des algorithmes de parcours de graphe.

Chapitre 6

Recommandation sur des données structurelles (graphes)

Résumé : Dans ce chapitre, nous présentons une nouvelle méthode de recommandation qui combine des approches de recherche d'information et des algorithmes de parcours de graphe. Nous décrivons la modélisation des données en structure de graphe ensuite nous introduisons l'architecture globale de notre méthode. Nous avons testé cette dernière sur deux collections de données de natures différentes ; la première est une collection standard de la campagne d'évaluation INEX, la tâche Social Book Search (qui a intégré CLEF Initiative en 2015) et la deuxième est une collection de documents scientifiques issue du portail OpenEdition.org plus précisément la plateforme Revues.org.

Sommaire

6.1 Introduction

Les systèmes de recommandation analysent plusieurs facteurs tels que l'historique de consultation, les achats, les préférences des utilisateurs, etc. En se basant sur cette analyse, ils créent une liste de recommandations pertinentes qui correspond à un utilisateur en particulier ou à son besoin en information. Nous avons vu précédemment qu'il existe plusieurs variétés de systèmes de recommandation, les deux principales sont le filtrage collaboratif et la recommandation basée sur le contenu. Dans ces deux catégories, on doit analyser tout le contenu des entités pour trouver l'item à recommander. Dans les approches collaboratives, cela est représenté par une matrice utilisateurs-items, dans les approches basées sur le contenu, c'est une matrice items-items qui contient les similarités entre les items. Cependant, on n'a aucune garantie que l'estimation est correcte et que les items recommandés sont assez pertinents pour l'utilisateur. Plusieurs systèmes de recommandation essayent de proposer des items en les associant au contexte du profil de l'utilisateur. De tels systèmes peuvent souffrir d'un réel problème de performance ce qui les rend inutilisables en temps réel BILLSUS et PAZZANI. De plus, il est courant que les systèmes de recommandation classiques présentent le problème de démarrage à froid pour un nouvel utilisateur ou un nouvel item.

Dans ce chapitre, nous présentons une nouvelle approche de recommandation dans le but de palier les problématiques engendrées par les systèmes de recommandation classiques. Notre approche combine des techniques de recherche d'information sur des données structurées en graphe. Les systèmes de recommandation basés sur les graphes ont été testés dans le passé et ont montré des résultats prometteurs HUANG et collab. [2002b].

Ces dernières années, une innovation importante dans la recherche d'information et la recommandation est faite en exploitant des relations entre les documents, un des exemples les plus courants est le PageRank de Google PAGE et collab. [1999]. Cela a produit un grand succès dans le web, où les relations sont créées sur la base des hyperliens existant entre les pages web. Dans l'approche que nous proposons, nous exploitons l'algorithme de PageRank pour le réordonnancement des recommandations. Pour l'évaluation de notre méthode, nous utilisons deux collections de données, la première regroupe des descriptions de livres d'Amazon (The CLEF Initiative, INEX pour la tâche Social Book Search - SBS 2015) et nous recommandons des livres aux utilisateurs ayant exprimé leurs besoins sous forme de requêtes complexes (la même collection est utilisée dans le chapitre précédent). La deuxième collection provient du portail d'OpenEdition principalement la plateforme Revues.org (sur laquelle, nous avons effectué une classification automatique pour la détection des comptes rendus de lecture, voir chapitre 4). Dans cette deuxième collection, nous souhaitons faire des recommandations de lecture de documents scientifiques dans le domaine des sciences humaines et sociales.

Ce chapitre est organisé en deux parties, dans la première nous présentons la méthode de recommandation appliquée sur la première collection de test d'Amazon. Nous commençons par décrire la méthode de modélisation des descriptions des livres suivie de l'architecture générale de la méthode de recommandation et ensuite nous présentons les résultats obtenus. Dans la deuxième partie, nous introduisons la méthode de recommandation utilisée sur la deuxième collection de test (Revues.org) avec la méthode de modélisation des documents. Nous présentons par la suite l'architecture générale ainsi que le prototype de recommandation (démonstrateur). Pour cette collection, nous ne disposons pas de phase d'évaluation avec des métriques spécifiques en revanche nous détaillerons le protocole d'évaluation conçu.

6.2 Recommandation basée sur des documents liés

6.2.1 Corpus de test (CLEF Initiative, INEX SBS)

Pour cette partie, nous avons utilisé le corpus fourni par la compagne d'évaluation INEX pour la tâche SBS en 2014 et 2015. Le choix de ce corpus est motivé par la nature des documents qu'il contient ainsi que sa taille. Nous souhaitons expérimenter notre méthode de recommandation sur un corpus volumineux pour évaluer les performances de calcul en plus des performances de recommandation.

Pour la collection de la tâche SBS, nous avons modélisé les documents en structure de graphe et avons appliqué l'algorithme de recommandation proposé que nous décrivons dans ce qui suit.

6.2.2 Modélisation des liens entre les documents à l'aide d'un graphe

Nous avons effectué une analyse de documents pour trouver une nouvelle façon de les relier. Dans le cas de la collection de la tâche SBS, nous avons exploité un type spécifique de similarité basée sur plusieurs facteurs. Cette similarité est pré-calculée par Amazon entre les différents items selon une méthode de filtrage collaboratif item-item. Cette dernière effectue un appariement entre les items achetés et notés de chaque utilisateur avec les items similaires. Pour trouver l'item le plus similaire pour un item donné, une table d'items est construite à partir des achats des utilisateurs LINDEN et collab. [2001]. Ensuite, Amazon utilise un algorithme itératif pour construire et calculer la similarité entre chaque item et tous les items relatifs dans la table d'items comme suit :

Algorithme 2 Algorithme d'Amazon pour les calculs des similarités entre les items. LINDEN et collab. [2003]

1: **Pour** chaque $i_1 \in$ catalogue des produits PC_{Amazon} **faire**
2: **Pour** chaque utilisateur u qui a acheté i_1 **faire**
3: **Pour** chaque i_2 acheté par u **faire**
4: Enregistrer que les items i_1 et i_2 ont été achetés ensemble dans la table d'items
5: **Pour** chaque i_2 **faire**
6: Calculer la similarité ente i_1 et i_2

L'algorithme 2 permet de trouver les items achetés ensemble pour calculer ensuite les similarités entre eux LINDEN et collab. [2003]. La mesure de similarité utilisée est Cosinus où on dispose d'un ensemble de vecteurs où chacun correspond à un item composé de M utilisateurs l'ayant acheté. Suivant la table d'items, Amazon propose des produits similaires pour chaque item. Le résultat de la sélection correspond au champ "Similar Products"dans chaque document de la collection de la tâche SBS.

Pour établir la modélisation de la collection en un graphe que nous avons appelé, "Directed Graph of Documents" (DGD), nous avons extrait les liens de "Similar Products". Dans le DGD, chaque nœud correspond à un document (une description de livre d'Amazon) et possède l'ensemble des propriétés suivantes :

1. *ID* : représente l'ISBN du livre ;
2. *Content* : représente la description du livre qui contient plusieurs autres propriétés (titre, description du livre donnée par Amazon, auteur(s), les tags, commentaires, etc.) ;

3. *MeanRating* : est la moyenne des votes attribués au livre ;
4. *PR* : représente le PageRank du livre.

Les relations dans le DGD sont orientées et correspondent aux similarités d'Amazon. Étant donné les deux nœuds $\{A, B\} \in S$, si A pointe vers B, B est suggéré comme un produit similaire de A par Amazon. Dans la figure 6.1, nous montrons un exemple de la structure du DGD. Ce dernier comporte au total 1 645 355 nœuds (86% des nœuds sont présents dans la collection et le reste n'y figure pas car ils ne possèdent pas de "Similar Products") et 6 582 258 relations, sachant que pour la plupart des documents ne possèdent pas plus de 6 "Similar Produits".

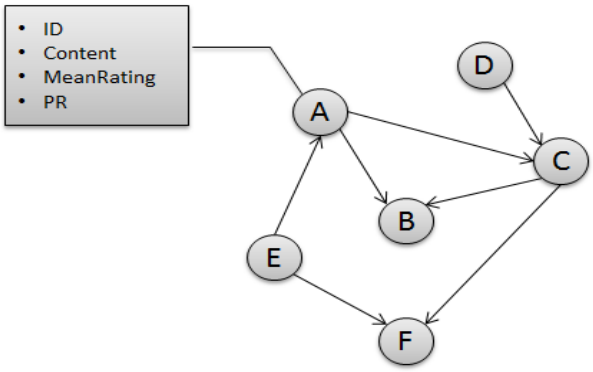


FIGURE 6.1 – Exemple du Directed Graph of Documents (DGD)

La figure 6.2 présente un extrait du graphe DGD où chaque nœud est représenté par l'ISBN du livre. Comme exemple de liaisons entre les documents dans le DGD, on a le livre dont l'ISBN est 0307269752 et le titre est : *The Girl with the Dragon Tattoo* qui inclut dans sa liste des produits similaires les livres suivants (nœuds en vert dans la figure 6.2) :

— ISBN : 1416562605, titre : *The White Tiger : A Novel (Man Booker Prize)*
— ISBN : 0061768065, titre : *The Story of Edgar Sawtelle : A Novel (Oprah Book Club #62)*
— ISBN : 0316166294, titre : *The Brass Verdict : A Novel*

Ce livre est aussi inclus dans les listes des produits similaires d'autres livres (nœuds en orange dans la figure 6.2) tels que :

— ISBN : 0330442422, titre : *Delivery Room*
— ISBN : 1590584422, titre : *Waterloo Sunset (Large Print)*
— ISBN : 1845297180, titre : *The Lost*

6.2.3 Architecture du système de recommandation proposé

La figure 6.3 montre l'architecture générale du système de recommandation proposé qui se compose de deux niveaux de recherche. Dans ce système, nous avons mis en œuvre plusieurs étapes. La première étape est la recherche avec un modèle de RI. elle permet de trouver des documents pour une requête utilisateur, ensuite, dans l'étape *Recherche dans le graphe* on sélectionne un ensemble de documents issus des algorithmes de parcours dans le graphe (décrits dans la section suivante). Le DGD (*Directed Graph of Documents*) est construit en utilisant la matrice des informations sociales et enrichi avec le calcul du PageRank. La matrice des informations sociales est issue des deux étapes d'extraction, l'extraction des votes

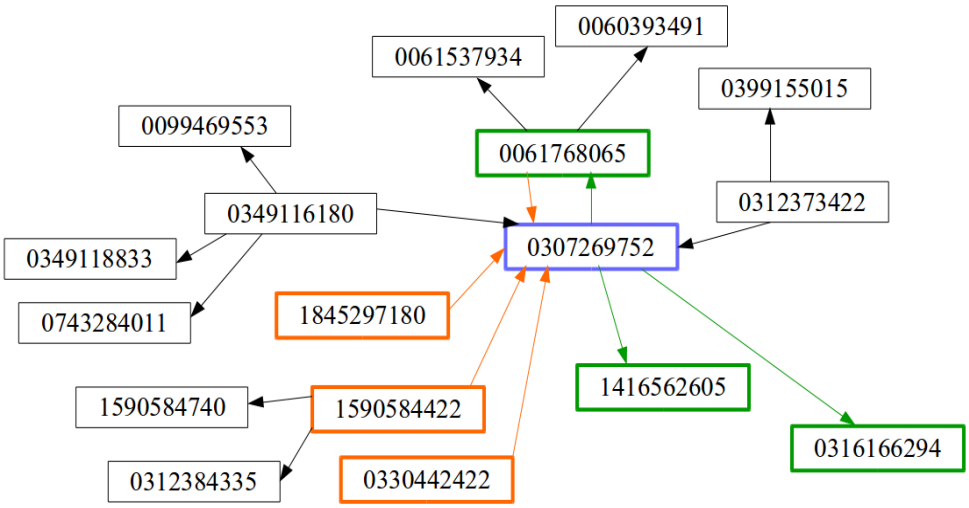


FIGURE 6.2 – Extrait du Directed Graph of Documents (DGD)

des utilisateurs pour chaque document et l'extraction des produits similaires à chaque document à partir de la collection d'Amazon (*Collection de documents*). Cette dernière contient les descriptions des livres d'Amazon sous format XML. La dernière étape évoquée par le système est le *Réordonnancement* qui se charge de la combinaison des scores issus des deux étapes de recherche *Modèle de RI* et *Recherche dans le graphe* et procède ensuite au réordonnancement des recommandations selon les scores combinés.

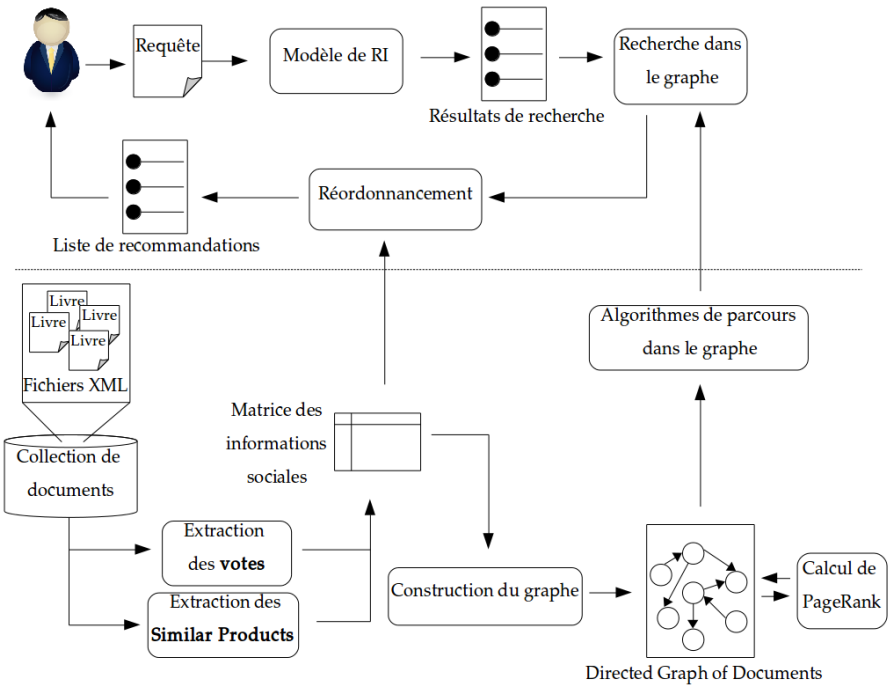


FIGURE 6.3 – L'architecture générale de l'approche de recommandation basée sur le graphe DGD.

Dans cette section, la collection des documents est dénotée par C. Dans C, chaque document d a un unique identifiant ID. L'ensemble des requêtes appelées topics est dénoté par T, l'ensemble $D_{init} \subset C$ fait référence aux documents retournés par le modèle de recherche initial dans le premier niveau du système de recommandation. *NœudDépart* est un document dans l'ensemble D_{init} qui est utilisé comme entrée des algorithmes de parcours du graphe

DGD. L'ensemble des documents présents dans le DGD est noté par S. D_{t_i} indique les documents recommandés pour un topic t_i dans l'ensemble T.

Le graphe DGD contient d'importantes informations sur les documents. Ces informations peuvent être exploitées pour la recommandation. Notre approche est basée, en premier sur le résultat d'une approche de recherche classique, ensuite sur le graphe DGD pour trouver plus de recommandations dans le but d'enrichir le résultat de la première étape.

Nous introduisons l'algorithme 3. Il prend en entrée : D_{init}, la liste de documents retournés pour chaque topic par les techniques de recherche décrites dans le chapitre précédent (section 5.1 et 5.4.1.2), le graphe DGD et le paramètre β qui est le nombre des premiers *NœudDépart* à partir de D_{init} dénoté par $D_{NœudDépart}$. L'algorithme retourne une liste de recommandations pour chaque topic notée par D_{final}. Il itère sur tout l'ensemble des topics et extrait la liste de tous les voisins de chaque *NœudDépart* dans le DGD. Les deux listes qui résultent (les nœuds voisins, $D_{Voisinage}$ et les nœuds des plus courts chemins, D_{PCC}) sont fusionnées après suppression de toutes les duplications. La liste retournée de l'étape précédente est dénotée par D_{graphe}. Une seconde fusion est faite entre la liste initiale D_{init} et la liste D_{graphe} (toutes les duplications sont également supprimées) dans une liste finale de recommandations D_{final} réordonnée en utilisant différents schémas de réordonnancement que nous présentons dans la section suivante.

Algorithme 3 Recommandation basée sur le DGD

$D_{init} \leftarrow$ Rechercher des documents pour chaque $t_i \in T$
Pour chaque $D_{t_i} \in D_{init}$ **faire**
    $D_{NœudDépart} \leftarrow$ premiers β documents $\in D_{t_i}$
    Pour chaque $Nœud Départ$ **dans** $D_{NœudDépart}$ **faire**
        $D_{Voisinage} \leftarrow D_{graphe} + Voisinage(NœudDépart, DGD)$
        $D_{PCC} \leftarrow$ **all** $D \in PCC(NœudDépart, D_{NœudDépart}, DGD)$
        $D_{graphe} \leftarrow D_{Voisinage} + D_{PCC}$
        Supprimer toutes les duplications dans D_{graphe}
    $D_{final} \leftarrow D_{final} + (D_{t_i} + D_{graphe})$
    Supprimer toutes les duplications dans D_{final}
    Réordonner D_{final}

La figure 6.4 illustre le procédé de l'approche de recommandation basée sur le DGD présentée dans l'algorithme 3.

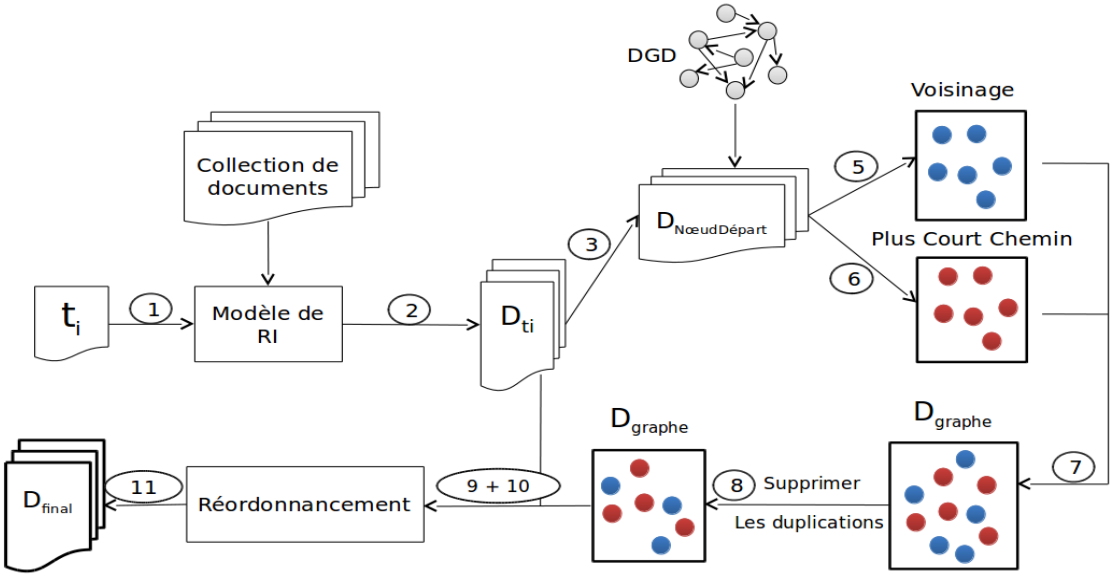


FIGURE 6.4 – Procédé suivi pour l'approche de recommandation basée sur le DGD. Nous avons numéroté chaque étape avec le numéro d'instruction correspondante dans l'algorithme 3.

6.2.4 Réordonnancement de la liste de recommandations

La dernière étape de l'approche de recommandation proposée est le réordonnancement des documents. L'ensemble des documents retournés par le premier niveau de recherche est fusionné à la liste des documents dans l'ensemble des voisins et le résultat de l'algorithme du plus court chemin. Nous avons testé différentes méthodes de réordonnancement qui combinent différents scores. Pour chaque document, nous avons calculé les scores suivants :

— **PageRank** : calculé avec l'outil NetworkX [1] (voir la formule dans la section 3.3.1.1). C'est un algorithme très populaire qui exploite la structure de liens pour donner un score d'importance aux nœuds dans le graphe. Généralement, le PageRank est utilisé dans les graphes d'hyperliens comme le web PAGE et collab. [1998].

— **MeanRatings** : un score calculé à partir des informations générées par les utilisateurs. Il représente la moyenne des votes attribués. Ce score est stocké dans *Social Information Matrix*.

— **SimJaccard** : calculée en utilisant la bibliothèque python "Distance " [2] . Nous avons calculé la similarité de Jaccard entre deux entités de texte, la première provient des topics et est constituée de la concaténation des champs *mediated query* et *title* (pour un topic t_i, on note cette concaténation (A)). La deuxième entité provient des documents restitués par le premier niveau de recherche pour chaque t_i, elle représente la concaténation du titre et la description du livre, notée (B). La distance de Jaccard est le résultat de la division entre le nombre de mots communs et le nombre total des mots dans les deux entités de texte, comme le montre la formule suivante :

$$SimJacc(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

1. C'est un outil programmé en Python pour la gestion des graphes. La version utilisée est la 1.9.1, https://networkx.github.io/documentation/networkx-1.9.1/overview.html
2. https://pypi.python.org/pypi/Distance/

Le score de cette similarité se base uniquement sur le contenu des documents et non-pas sur les informations sociales contrairement aux scores précédents. Nous avons choisi un tel score pour montrer l'impact sur le réordonnancement en utilisant les informations sociales et les informations descriptives.

Les scores calculés sont normalisés en utilisant la formule suivante : $normalized_{score} = old_{score}/max_{score}$. Ensuite, pour combiner les scores du premier niveau de recherche et les scores normalisés, nous avons expérimenté une simple pondération, ce qui favorise les documents avec des valeurs élevées de PageRank et de MeanRatings et même si leurs contenus ne sont pas totalement correspondants à la requête de l'utilisateur. La deuxième combinaison est linéaire (voir section suivante).

6.2.5 Expérimentations

Dans cette section, nous décrivons les différentes expérimentations faites avec la collection de la tâche SBS (2014 et 2015). Nous présentons les configurations et les paramétrages effectués dans l'approche de recommandation décrite précédemment.

6.2.5.1 Outils et Configurations

Pour les différentes expérimentations, nous avons utilisé *Terrier* (TERabyte RetrIEveR) [3] pour déployer le modèle Inl2. Le choix de ce modèle est motivé par le fait qu'il a obtenu les meilleurs résultats dans le chapitre précédent. Un autre outil a été utilisé pour la gestion et le parcours dans le graphe DGD qui est NetworkX [4].

Le deuxième niveau dans l'approche proposée évoque le graphe de documents construit à partir de la collection de la tâche SBS. Cette étape inclut la gestion et l'utilisation des algorithmes de parcours dans le graphe. Nous rappelons que ce dernier contient plus de 1,5 millions de nœuds avec plus de 6 millions de relations. Nous avons stocké le graphe dans un fichier de format GraphML [5]. Ce fichier se compose de trois parties : la première décrit les caractéristiques du graphe (orienté ou non, les propriétés sur les nœuds, les propriétés sur les relations, etc.), la deuxième partie regroupe l'ensemble des nœuds du graphe dans des champs "node" et la dernière partie détaille les relations dans "edge" avec dans chacune, le nœud source "source" et le nœud destination "target" (voir figure 6.5).

Dans le cas de notre graphe DGD, nous n'avons des propriétés que sur les nœuds mais pour des raisons de temps de calcul, nous avons stocké ces informations dans une matrice que nous avons appelée "Matrice sociale". Cette dernière se compose de quatre colonnes (ISBN, chemin vers le document, Moyenne des votes, PageRank).

6.2.5.2 Approches de recommandation

Nous avons testé trois approches de recommandation qui suivent le même processus de sélection. Les recommandations sont obtenues avec le procédé décrit dans la figure 6.4 qui se décompose d'une première sélection de documents avec le modèle de recherche InL2. Ensuite un ensemble de documents est choisi pour effectuer une sélection de voisinage et la recherche des plus courts chemins dans le graphe DGD (voir l'algorithme 3). La différence entre les approches réside dans la méthode de réordonnancement. Les approches sont les

3. http://terrier.org/
4. https://networkx.github.io/documentation/networkx-1.9.1/overview.html
5. http://graphml.graphdrawing.org/

```
−<graphml xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
 http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
 −<graph id="G" edgedefault="directed">
    <node id="087548025X"/>
    <node id="079357725X"/>
    <node id="031202925X"/>
    <node id="0786862297"/>
    <node id="031202925X"/>
      .
      .
      .
    <edge id="1000000" source="087548025X" target="0821406914" label="similarproduct"/>
    <edge id="1000001" source="087548025X" target="087220037X" label="similarproduct"/>
    <edge id="1000002" source="079357725X" target="0634031201" label="similarproduct"/>
    <edge id="1000003" source="079357725X" target="0793576903" label="similarproduct"/>
    <edge id="1000004" source="079357725X" target="0793577241" label="similarproduct"/>
    <edge id="1000005" source="079357725X" target="0793585856" label="similarproduct"/>
    <edge id="1000006" source="079357725X" target="0793592860" label="similarproduct"/>
    <edge id="1000007" source="031202925X" target="0312017952" label="similarproduct"/>
    <edge id="1000008" source="031202925X" target="0312022212" label="similarproduct"/>
    <edge id="1000009" source="031202925X" target="031211415X" label="similarproduct"/>
    <edge id="100000" source="0786862297" target="0684835576" label="similarproduct"/>
    <edge id="1000010" source="031202925X" target="0312164335" label="similarproduct"/>
    <edge id="1000011" source="031202925X" target="0373261012" label="similarproduct"/>
    <edge id="1000012" source="006153725X" target="0307265730" label="similarproduct"/>
    <edge id="1000013" source="006153725X" target="0307269752" label="similarproduct"/>
```

FIGURE 6.5 – Extrait du fichier GraphML qui stocke le DGD.

suivantes :

1. **Recommandation basée sur le graphe DGD et le PageRank**, (nommée *InL2+DGD+PR*) : dans cette approche, nous utilisons comme score de réordonnancement de la liste des recommandations, une combinaison linéaire du score du modèle InL2 avec le PageRank (combinaison linéaire, voir la section suivante).
2. **Recommandation basée sur le graphe DGD et les votes des utilisateurs**, (nommée *InL2+DGD+MnRatg*) : dans cette approche, la moyenne des votes des utilisateurs est combinée avec le score du modèle InL2 (combinaison linéaire, voir la section suivante).
3. **Recommandation basée sur le graphe DGD et la similarité de Jaccard**, (nommée *InL2+DGD+SimJacc*) : le score de réordonnancement utilisé est la similarité de Jaccard (comme décrit dans la section 6.2.4). Elle est combinée au score du modèle InL2 (combinaison linéaire, voir la section suivante).

Toutes ces approches ont été présentées au Workshop de la conférence CLEF BENKOUSSAS et BELLOT [2015a]. Dans la section suivante, nous décrivons les méthodes de combinaison des scores et nous présentons les résultats.

6.2.5.3 Résultats

Dans cette section, nous détaillons les différentes expérimentations et les résultats obtenus. Trois types d'expérimentations ont été testées selon les méthodes de réordonnancement

définies précédemment. A l'issue de l'étape de recherche faite par le modèle InL2, une sélection d'un nombre de "NœudDépart" à partir des résultats de la recherche est nécessaire pour pouvoir faire le parcours dans le graphe. Le nombre de ces nœuds influence sur les performances de recommandation. Nous avons fait varier le nombre de nœuds avec la méthode de recommandation basée sur le PageRank pour le réordonnancement de telle sorte qu'à chaque itération, on augmente le nombre d'entrées dans le graphe (les nœuds de départ) par 100 documents. À partir de ces points d'entrées, on lance l'algorithme de recommandation.

Le graphique 6.6, montre que la meilleur valeur de nDCG@10 est obtenue avec 100 nœuds nœuds de départ (10% de la liste des documents retournée par la première étape de recherche). Nous observons une dégradation des performances au delà de 100 "NœudDépart".

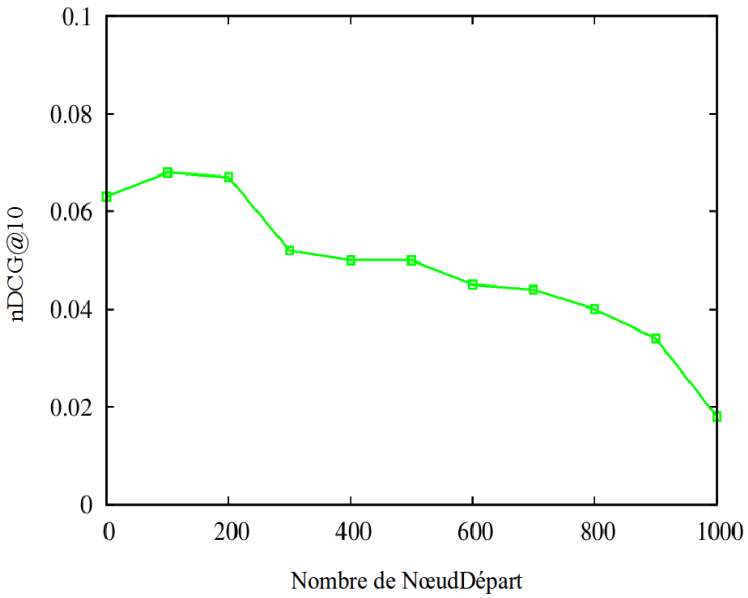


FIGURE 6.6 – Résultats des variations du nombre de nœuds de départ selon la recommandation basée sur le PageRank et la mesure nDCG@10. (CLEF Labs, la tâche SBS 2015)

Nous avons défini des scores utilisés pour le réordonnancement de la liste finale des recommandations (PageRank, MeanRating et SimJacc). Pour la combinaison des scores issus de l'étape de recherche avec ces derniers, nous avons expérimenté deux méthodes, la première avec une interpolation linéaire avec variation du paramètre d'interpolation (α). Les différentes variations pour chacun des scores sont présentées dans la figure 6.7. Nous obtenons de meilleurs résultats avec un $\alpha = 0,75$ pour la combinaison du PageRank avec les scores de InL2 (l'approche InL2+DGD+PR), $\alpha = 0,85$ pour la combinaison du MeanRatg avec les scores de InL2 (l'approche InL2+DGD+MeanRatg) et aucune amélioration pour la combinaison avec SimJacc (l'approche InL2+DGD+SimJacc). Notons que pour $\alpha = 1$, nous n'avons que le résultat de l'étape de recherche (InL2). Il est clair d'après les graphiques présentés que plus le poids des scores de réordonnancement est élevé plus les performances de recommandation se dégradent, ceci montre que ces scores seuls ne peuvent pas aboutir à un réordonnancement qui satisfait l'utilisateur.

La deuxième méthode utilisée pour combiner les scores est une simple pondération. Dans le tableau 6.1, nous présentons les résultats de recommandation avec une comparaison entre les différentes méthodes de réordonnancement. Nous montrons également la différence entre les meilleures valeurs obtenues avec une interpolation linéaire et une simple pondération des

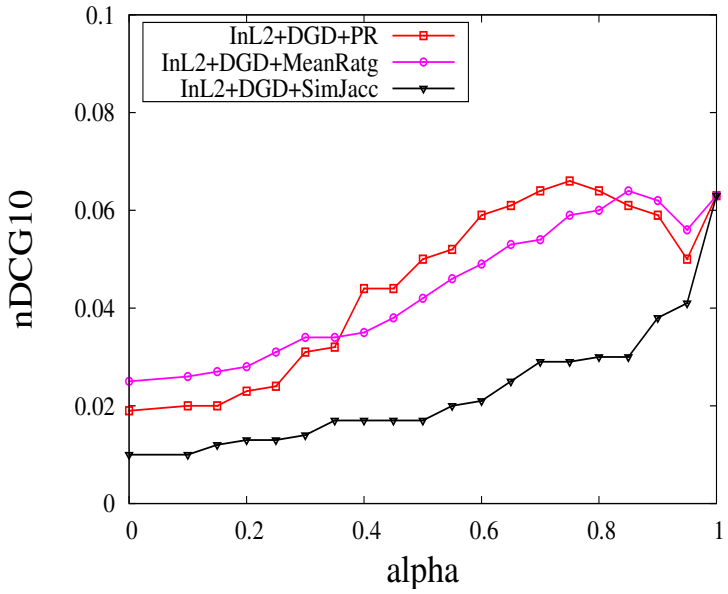


FIGURE 6.7 – Résultats des variations du paramètre d'interpolation α pour les différents scores de réordonnancement selon la mesure nDCG@10. (CLEF SBS 2015)

scores. Nous avons expérimenté sur les données de la tâche SBS 2014 (680 topics) et 2015 (208 topics).

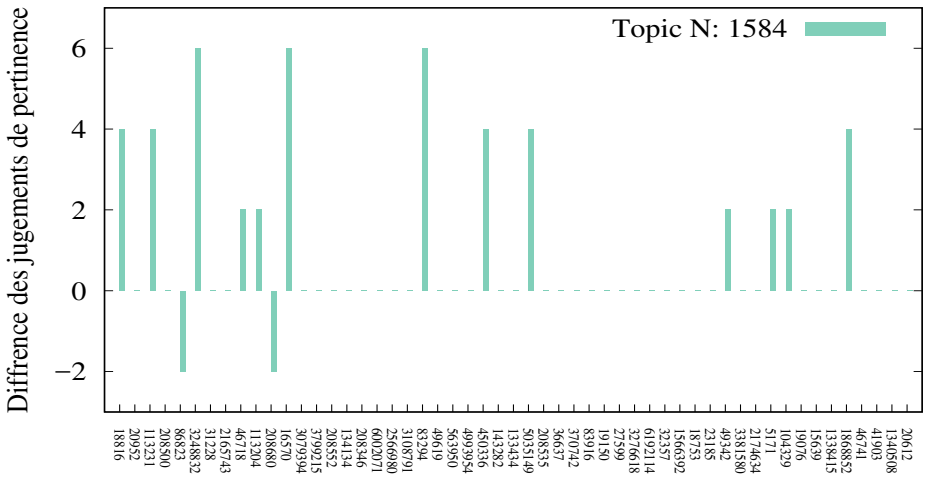
TABLEAU 6.1 – Résultats expérimentaux. Les méthodes sont ordonnées selon la mesure nDCG@10. Les lignes en gris regroupent les méthodes utilisant la multiplication pour combiner les scores. (∗) dénote les résultats significatifs selon le test de Wilcoxon CROFT [1978] avec deux faces p-valeur, $\sigma = 0,05$.

2014 Topic Set				
Approche	nDCG@10	Recip Rank	MAP	P@10
InL2	**0.128**	**0.236**	**0.101**	**0.067**
InL2+DGD+PR ($\alpha = 0.75$)	$0.108^{(-15\%^*)}$	$0.216^{(-8\%*)}$	$0.092^{(\%^*)}$	$0.061^{(-8\%^*)}$
InL2+DGD+PR	$0.122^{(-4\%)}$	$0.239^{(+1\%)}$	$0.090^{(-9\%^*)}$	$0.069^{(+2\%^*)}$
InL2+DGD+MnRatg ($\alpha = 0.85$)	$0.098^{(-23\%)}$	$0.189^{(-19\%^*)}$	$0.074^{(-26\%)}$	$0.052^{(-22\%)}$
InL2+DGD+MnRatg	$0.105^{(-17\%)}$	$0.192^{(-18\%)}$	$0.081^{(-18\%^*)}$	$0.057^{(-15\%^*)}$
2015 Topic Set				
Approche	nDCG@10	Recip Rank	MAP	P@10
InL2	**0.063**	**0.147**	**0.046**	**0.044**
InL2+DGD+PR ($\alpha = 0.75$)	$0.066^{(+4\%^*)}$	$0.149^{(+1\%)}$	$0.046^{(+0\%^*)}$	$0.049^{(+11\%^*)}$
InL2+DGD+PR	$0.068^{(+8\%^*)}$	$0.157^{(+6\%)}$	$0.048^{(+0.6\%^*}$	$0.052^{(+18\%^*)}$
InL2+DGD+MnRatg ($\alpha = 0.85$)	$0.064^{(+1\%^*)}$	$0.146^{(-0.6\%^*)}$	$0.041^{(-10\%^*)}$	$0.047^{(+6\%^*)}$
InL2+DGD+MnRatg	$0.066^{(+4\%^*)}$	$0.148^{(-0.6\%^*)}$	$0.042^{(-8\%)}$	$0.052^{(+18\%)}$

Comme illustré dans le tableau, les performances entre les méthodes qui combinent les scores avec une multiplication des scores sont élevées que celles avec une combinaison linéaire. Cependant pour les topics 2014, aucune amélioration n'est observée. En revanche pour les topics de 2015, on remarque une amélioration significative en terme de nDCG@10 pour chacune des méthodes. Les résultats de l'approche InL2+DGD+PR utilisant les topics 2015 confirme que l'incorporation des scores du PageRank dans l'approche

de recommandation améliore les performances. La méthode qui utilise les scores MeanRatg (InL2+DGD+MnRatg) donne les résultats les plus bas ce qui signifie que les votes donnés par les utilisateurs ne constituent pas une information facilement exploitable pour améliorer l'ordonnancement des documents.

Nous pouvons clairement observer la différence entre les résultats pour les topics de 2014 et de 2015. Nous pensons que la raison principale de cette différence est le processus d'évaluation qui n'est pas le même. Nous avons observé que les valeurs de pertinence sont nettement différentes pour un même livre en raison du changement de la méthode de calcul des valeurs de pertinence KOOLEN et collab. [2015b, 2014]. Nous montrons dans les figures 6.8 et 6.9 la différence entre les valeurs de pertinence dans les deux topics (N°1584 et N°7243).



LibraryThing Id du livre

FIGURE 6.8 – Différence entre les valeurs de pertinence entre 2014 et 2015 pour le topic N°1584 dans les fichiers de référence qrels.

Pour comprendre le comportement des systèmes de recommandation proposés, nous avons effectué une étude sur chacun des topics pour l'année 2014 pour distinguer les topics dont les résultats sont améliorés, détériorés ou ont obtenu les mêmes résultats par rapport à la baseline InL2. L'histogramme dans la figure 6.10 montre qu'une grande partie des topics a connu une amélioration avec la méthode de recommandation à base de graphe. L'utilisation du score PageRank améliore plus de topics (15,05%) que le score MeanRatg.

Nous avons regardé de près la différence entre les topics améliorés et ceux détériorés et nous avons remarqué que ces topics se distinguent par la façon dont l'utilisateur pose sa requête dans la partie narrative. Dans le topic illustré dans la figure 6.11, l'utilisateur exprime son besoin par des exemples de lectures précédentes. L'auteur du topic demande à la fin des suggestions dans le même type des livres qu'il a déjà lus. Ce type de besoin oriente les recommandations vers un sujet précis contrairement à ce qui est connu sur les requêtes où l'utilisateur définit ce qu'il recherche d'une manière générale en décrivant la thématique et le domaine d'intérêt sans donner des exemples similaires à ce qu'il recherche comme le cas du topic illustré dans la figure 6.12. De ce fait et en collaboration avec Anaïs Ollagnier, nous avons divisé le corpus des topics (de l'année 2014) en deux classes que nous avons

FIGURE 6.9 – Différence entre les jugements de pertinence entre 2014 et 2015 pour le topic N°7243 dans les fichiers de référence qrels.

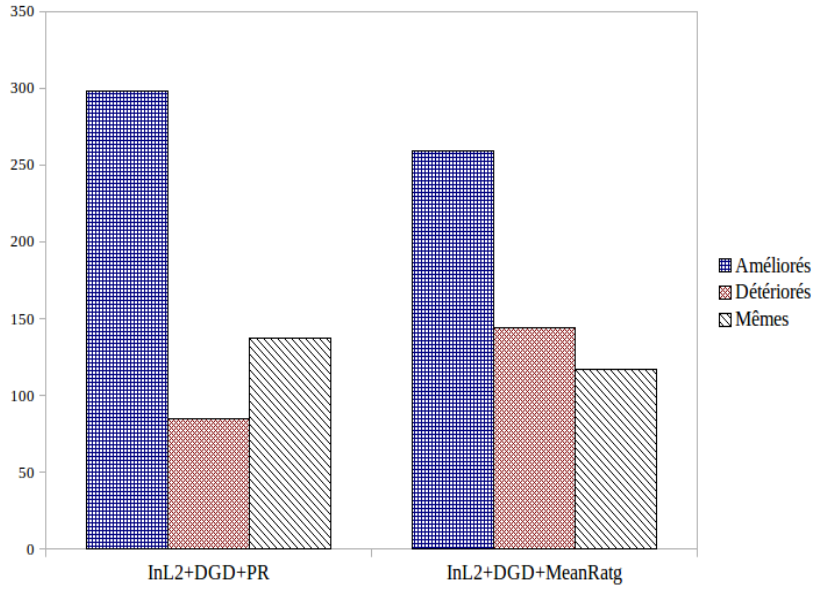


FIGURE 6.10 – Histogrammes qui illustrent et comparent le nombre de topics améliorés, détériorés et ceux qui n'ont eu aucune amélioration en utilisant la méthode de recommandation proposée. Les résultats sont comparés selon la mesure MAP avec la baseline InL2.

appelé *Analogue* (261 topics) et *Non-Analogue* (356 topics) à l'aide d'un processus d'apprentissage. Nous avons vérifié les résultats de classification manuellement pour corriger les erreurs de prédiction et avons ignoré 63 topics qui étaient difficiles à classer. La classe Analogue contient les topics où les utilisateurs donnent des exemples de lectures précédentes pour décrire leurs besoins (exemple dans la figure 6.11) et la classe Non-Analogue contient les topics où les besoins sont décrits d'une manière classique (exemple dans la figure 6.12).

Dans le but d'évaluer la méthode de recommandation basée sur le graphe DGD, nous avons fait deux séries d'expériences de recommandation. Dans la première, nous avons uti-

FIGURE 6.11 – Topic N : 7301 (classe Analogue)

```
<topic id="7301">
<title>
 Military Heroes?
</title>
<mediated_query>
 military romance
</mediated_query>
<group>
 Romance - from historical to contemporary
</group>
<narrative>
 I seem to have a thing for military-related romances. I think I
    just like adventure and men in uniform. I guess my induction
    into this genre was with Christina Skye , but recently, I've
    read Catherine Mann 's Wingmen Warriors series, and I
    absolutely adored it. I had always avoided Harlequins, but
    these were so worth the read. I've also read and enjoyed some
    stuff by Merline Lovelace. Anyone else share a love for this
    type of book and have suggestions of other authors I should
    check out?
</narrative>
</topic>
```

FIGURE 6.12 – Topic N : 17244 (classe Non-Analogue)

```
<topic id="17244">
<title>
 African American soldiers in World War II
</title>
<mediated_query>
 soldiers in World War II
</mediated_query>
<group>
 Military History
</group>
<narrative>
 I am interested in reading about the role of these soldiers in
    World War II. Do any of you have any book suggestions?
</narrative>
</topic>
```

lisé la baseline InL2 pour rechercher des livres pour les topics de la classe Non-Analogue. Dans la deuxième série d'expérience, nous avons utilisé la méthode de recommandation basée sur le graphe pour rechercher des livres pour les topics de la classe Analogue.

Dans le tableau 6.2, nous pouvons constater que le comportement de la baseline est différent pour chacune des classes des topics. Le modèle probabiliste InL2 est plus performant avec les topics de classe Non-Analogue, ce qui s'explique par la nature de la requête qui contient des phrases descriptives du besoin de l'utilisateur. Ce type de requête est plus ex-

TABLEAU 6.2 – Résultats expérimentaux des deux classes de topics Analogues et Non-Analogues. (∗) dénote les résultats significatifs selon le test de Wilcoxon CROFT [1978] avec une p-valeur = 0,05.

	Topics Analogues			
Approche	nDCG@10	Recip Rank	MAP	P@10
InL2	**0.109**	**0.267**	**0.072**	**0.078**
InL2+DGD+PR	$0.111^{(+1\%*)}$	$0.277^{(+3\%*)}$	$0.068^{(-5\%*)}$	$0.082^{(+12\%)}$
InL2+DGD+MnRatg	$0.104^{(-5\%)}$	$0.275^{(+2\%)}$	$0.064^{(-11\%*)}$	$0.082^{(+5\%)}$
	Topics Non-Analogues			
InL2	**0.138**	**0.207**	**0.117**	**0.057**
InL2+DGD+PR	$0.127^{(-7\%*)}$	$0.206^{(-0.6\%*)}$	$0.102^{(-12\%*)}$	$0.057^{(-1\%*)}$
InL2+DGD+MnRatg	$0.130^{(-5\%)}$	$0.214^{(+3\%*)}$	$0.100^{(-14\%^{*})}$	$0.067^{(+16\%)}$

plicite que les requêtes de classe Analogue. Les résultats de la méthode InL2+DGD+PR montrent que l'exploitation de la structure de graphe pour les topics de classe Analogue est significativement plus performant qu'avec les topics de la classe Non-Analogue (+1% de nDCG10 pour la classe Analogue contre -7% pour a classe Non-Analogue et jusqu'à +12% de précision pour la classe Analogue contre -1% pour la classe Non-Analogue). Ces résultats peuvent s'expliquer par le fait que les topics Analogues contiennent des exemples de livres déjà lus ce qui nécessite l'utilisation du graphe DGD pour extraire les livres similaires connectés aux livres exemples donnés par l'utilisateur.

Comme déjà vu dans le tableau de résultats 6.1 pour les topics de 2014, l'utilisation du score social (MeanRating) dans la méthode InL2+DGD+MnRatg dégrade significativement les résultats pour les deux classes de topics comparés aux résultats de la baseline.

Pour conclure, la modélisation des documents en une structure de graphe suivant des informations sociales pour définir les liaisons, peut être un bon moyen pour améliorer les performances de recommandation basée sur des requêtes complexes. Plus précisément, avoir des requêtes complexes contenant des exemples de lectures précédentes peut aider à proposer de meilleurs recommandations.

6.3 Recommandation basée sur un réseau de citations

Dans cette partie du chapitre, nous nous intéressons à un autre contexte de recommandation. Nous nous orientons vers un cas d'application dans le domaine des sciences humaines et sociales. Nous allons dans la suite de ce chapitre, décrire l'application de la même méthode de recommandation définie précédemment avec les différents changements pour l'adapter aux données de test.

6.3.1 Corpus de test (Revues.org)

Nous avons choisi comme corpus de test, la collection des documents de Revues.org qui est l'une des plateformes du portail OpenEdition (section 4.3). Cette collection contient plus de 100 000 documents scientifiques de différentes natures (voir le tableau 4.2 dans le chapitre 4). Dans la majorité de ces documents, on trouve une bibliographie qui contient des références vers d'autres documents internes et externes à Revues.org. Le choix de Revues.org a été motivé par le volume de documents qu'elle contient et par leur homogénéité.

6.3.2 Construction du graphe basé sur les références bibliographiques

Il s'agit d'exploiter les références bibliographiques présentes dans chaque document pour établir des liaisons de citations. Pour cela, nous avons eu recours à un outil développé par l'équipe d'OpenEdition qui s'appelle Bilbo KIM et collab. [2011, 2012a,b]. C'est un outil d'annotation automatique des références bibliographiques. Bilbo permet de détecter, d'analyser et d'annoter sémantiquement les références bibliographiques présentes dans un document, qu'elles soient complètes (dans la bibliographie) ou très partielles (dans les notes de bas de page). Par des méthodes de fouille de texte et d'apprentissage automatique, il identifie le prénom et le nom des auteurs, les titres, les éditeurs, l'année et le lieu d'édition pour une référence donnée. Ce logiciel contient un module en option qui permet de rechercher pour une référence détectée et annotée, son DOI [6] (*Digital Object Identifier*) en interrogeant l'API de CrossRef [7] quand celui-ci existe, et ainsi rendre la citation clicable.

Bilbo repose sur le modèle d'apprentissage automatique CRF (*Conditional Random Fields*) pour l'étiquetage des références bibliographiques et sur le modèle SVM pour l'identification des zones de texte contenant des références bibliographiques (notes de bas de page).

Actuellement Bilbo est implémenté sur près de 80% des revues de la plateforme Revues.org. Il a pu extraire à la date du 31 décembre 2015, 1 042 780 références, dont 9.83% possédaient un DOI.

En se basant sur le résultats de cet outil, nous avons construit un graphe de documents. Ce graphe est orienté. Il est composé de nœuds qui représentent les documents de Revues.org, documents externes à Revues.org (présents dans les autres plateformes d'OpenEdition) et d'autres documents externes au portail OpenEdition et des relations qui sont des relations de citations. Le graphe construit est un graphe à propriétés, c-à-d, les nœuds possèdent des propriétés sous forme de clés-valeurs. Le schéma dans la figure 6.13 illustre la structure du graphe que nous appelons *Grapher*.

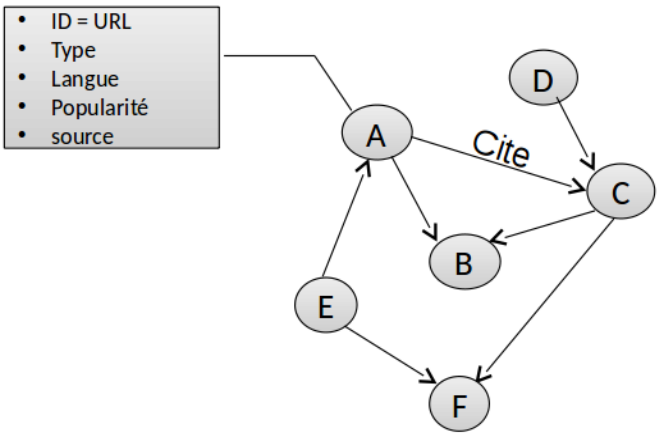


FIGURE 6.13 – Exemple schématisant le Grapher. Les liens représentent des relations de citation entre les documents scientifiques. Chacun de ces derniers dispose d'un ensemble de propriétés.

Chaque nœud contient un identifiant qui est l'URL du document, un type (article, compte rendu, chapitre d'ouvrage, éditorial, etc.), la langue, un score de popularité qui est représenté par le nombre de consultation par des lecteurs et la source du document qui indique si le

6. http://bbf.enssib.fr/consulter/bbf-1998-03-0049-007
7. http://www.crossref.org/

document est interne ou externe à Revues.org. Un nœud A qui pointe vers un nœud B signifie que A cite B autrement dit, B est présent dans la liste des références bibliographiques de A et il a été détecté et annoté automatiquement par Bilbo. Ces liens sont comparables dans leur principe à des liens entre pages Web, que les moteurs de recherche traitent généralement comme des votes en faveur de la page liée (plus il y a de pages qui pointent vers elle, plus une page a de chances d'être fiable et intéressante pour l'utilisateur final). Dans cette logique, nous avons procédé à des calculs de type PageRank de façon à déterminer des scores de pertinence qui sont en fonction des références bibliographiques. L'indice ainsi calculé est ensuite inclus, en tant que propriété dans le Grapher, de telle sorte qu'il peut servir d'indice d'ordonnancement de la liste des recommandations.

Dans le tableau suivant, voici quelques caractéristiques et statistiques sur le Grapher (réalisé avec NetworkX).

TABLEAU 6.3 – Caractéristiques Statistiques sur le Grapher

Connecté	Non
Orienté acyclique	Oui
Eulerien	Non
Nombre de noeuds	84677
Nombre de liens	27611
Nombre de composantes connexes	65983
Nombre de composantes connectées	65853

Nous avons utilisé le format Graphml pour stocker le graphe, de la même manière qu'avec le DGD défini précédemment. La figure 6.14, montre un extrait du graphe construit à partir de Revues.org. Nous avont pris l'exemple du nœud `http://asp.revues.org/1811` dont le titre est : *Grammaire et degré de spécialisation* qui cite par exemple, l'article `http://asp.revues.org/1565` intitulé : *De la contradiction dans la formation en anglais Langue Étrangère Appliquée (LEA)*, et qui est cité par plusieurs autres documents tels que :

— `http://asp.revues.org/3272`, son titre est : *Muriel Grosbois, Didactique des langues et technologies : de l'EAO aux réseaux sociaux*

— `http://asp.revues.org/3066`, son titre est : *L'anglais de spécialité en chimie organique : entre indétermination terminologique et multidimensionnalité*

— `http://asp.revues.org/3340`, son titre est : *Your very first ESP text (wherein Chaucer explaineth the astrolabe)*

Dans le graphe, les documents reliés entre eux partagent généralement une thématique commune avec différents niveaux de spécialisation.

6.3.3 Architecture du système de recommandation pour OpenEdition, Revues.org

La figure 6.15 montre l'architecture générale du système. Ce système se compose de deux niveaux de recherche. Le premier niveau repose sur une requête d'utilisateur classique (ensemble de mots clés) qui renvoie une liste de documents qui correspondent à cette requête. Le deuxième niveau peut être répétitif selon l'envie de l'utilisateur, il propose des documents à partir du Grapher en relation avec un document pré-sélectionné dans la liste du

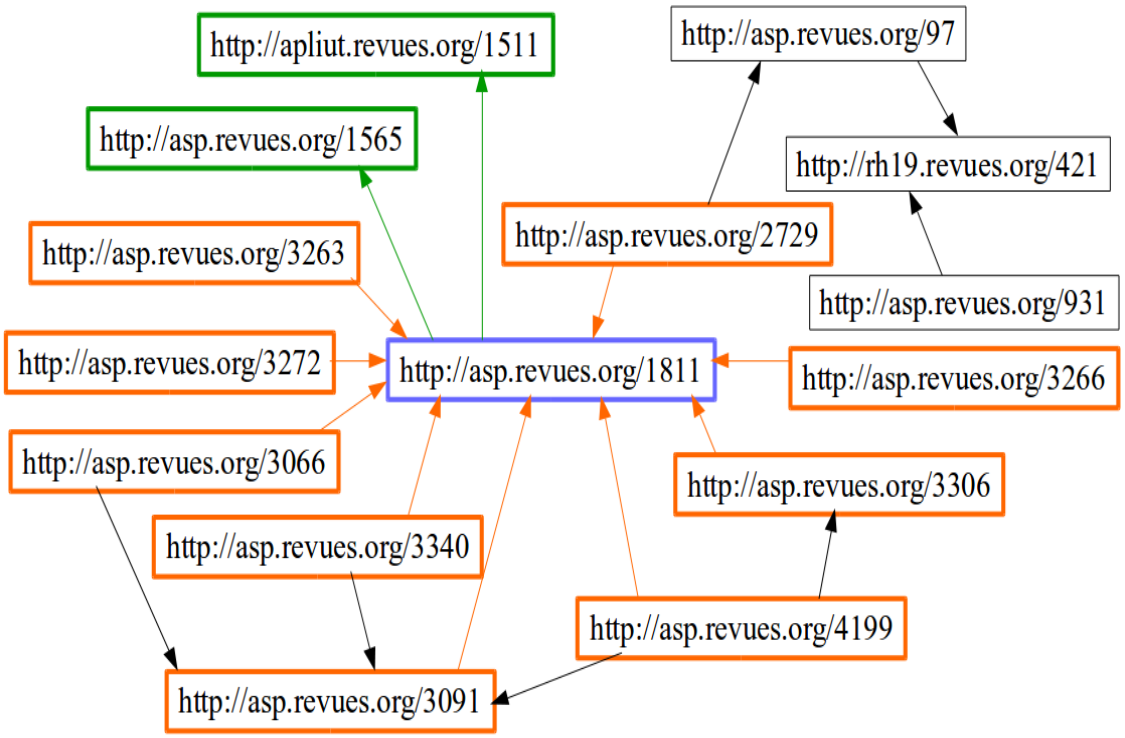


FIGURE 6.14 – Exemple d'un extrait du Grapher

premier niveau de recherche.

Différents étapes interviennent dans le système, la première consiste à une recherche avec le moteur Apache Solr [8]. Son rôle est de restituer des documents (*Résultats Solr*) pour répondre à la requête de l'utilisateur. Ensuite, l'étape de recherche dans le graphe se déclenche et fait appel aux algorithmes de parcours dans le graphe en prenant en entrée le document sélectionné dans la liste retournée par le moteur Solr. Le *Grapher* est construit en effectuant des interrogations sur l'index Solr (*interrogation Solr*) pour collecter l'ensemble des nœuds avec leurs propriétés et le logiciel *Bilbo* qui lui-même utilise le service *CrossRef* pour trouver les liens.

La dernière étape du système est le réordonnancement qui se charge de la combinaison des scores pour les documents retournés par la recherche dans le graphe et procède ensuite au réordonnancement des recommandations selon ces scores. La liste finale des recommandations est proposée à l'utilisateur, à partir de laquelle il peut choisir un nouveau document et déclenche à nouveau le processus de recommandation par rapport à ce nouveau document.

Nous avons réutilisé l'algorithme 3 présenté précédemment (section 6.2.3).

La dernière étape dans le processus est le réordonnancement des documents (instruction 11 dans l'algorithme 3). Ces derniers sont ordonnés selon un score qui multiplie la popularité de chaque document (son nombre de consultations) et le score du PageRank.

6.3.4 Infrastructure

Cette partie présente l'ensemble de l'infrastructure d'un point de vue technique. La figure 6.16 montre un aperçu de l'infrastructure générale du prototype de recommandation de la plateforme Revues.org. Le serveur (au centre) contient l'implémentation du système de re-

8. http://lucene.apache.org/solr/

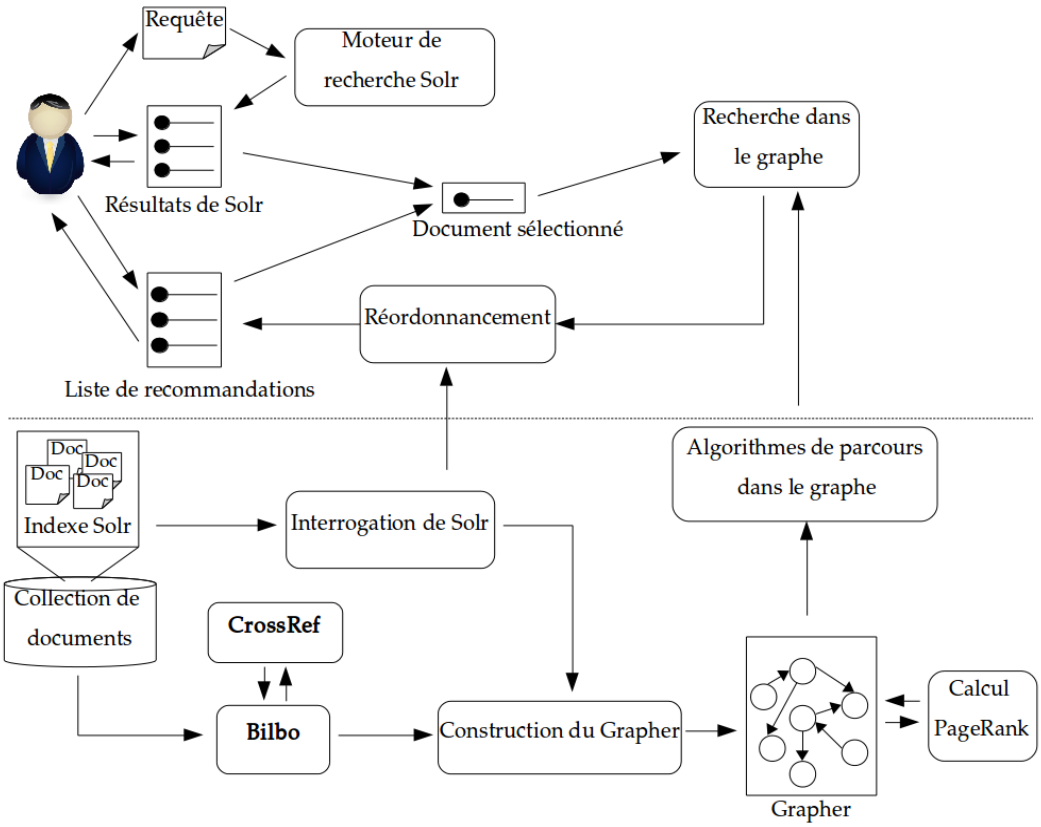


FIGURE 6.15 – L'architecture générale de l'approche de recommandation du Grapher

commandation exposé dans la section précédente. Une interface est utilisée pour transmettre au serveur le besoin en information de l'utilisateur sous forme de requête afin d'obtenir un ensemble de documents à partir de l'index Solr. Ensuite, cet ensemble est transmis à l'utilisateur. Ce dernier avec un clic, choisit un document qu'il jugera pertinent pour sa requête. Dans le cas où il veut des recommandations par rapport à ce document, le Grapher est utilisé par le système de recommandation pour les générer. Elles sont ensuite affichées sur l'interface de l'utilisateur. Les informations sur chacune des recommandations (notamment le contenu) sont récupérées directement dans l'index Solr, n'étant par toutes présentes dans le Grapher.

Dans la figure 6.16, on peut voir deux types de stockage de données, Le premier est un index Solr stockant tout le contenu de Revues.org indexé. Le second type de stockage de données, même si ce terme est un peu réducteur, est un graphe (le Grapher). Ce dernier est transféré en mémoire par le système de recommandation à son démarrage [9]. Comme défini précédemment (section 6.3.2), le graphe modélise le contenu suivant les liaisons de citations entre les documents.

6.3.5 Interface utilisateur

L'introduction du système de recommandation demande la création d'une interface pour dialoguer avec le serveur. Elle doit transmettre les informations sur le besoin de l'utilisateur

9. La communication entre les différentes composantes, dans un but de flexibilité et de compatibilité, des requêtes de type GET sont utilisées pour la communication entre l'interface de l'utilisateur et le serveur. Le serveur reçoit ainsi une requête HTTP de type GET et renvoie une liste d'URL. Le prototype de recommandation est implémenté en utilisant le langage Python et l'outil Flask (`http://flask.pocoo.org/`) pour le démonstrateur. Pour l'interrogation de Solr, nous avons utilisé la bibliothèque *Pysolr*.

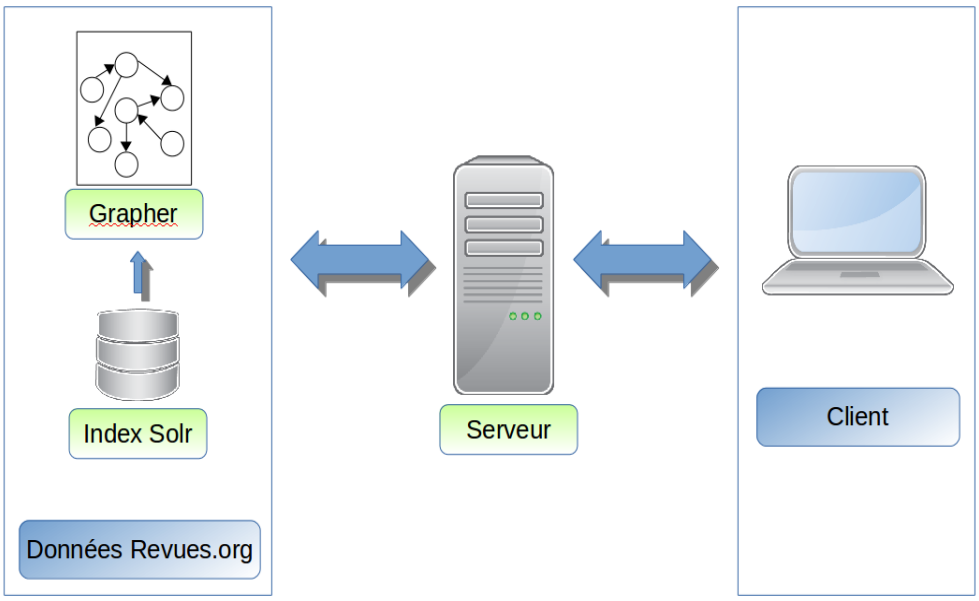


FIGURE 6.16 – Aperçu de l'infrastructure du système de recommandation utilisant le Grapher

afin de lui fournir des recommandations de lecture.

Pour cela, un démonstrateur a été créé. Dans ce qui suit, nous présentons les fonctionnalités du démonstrateur avec en parallèle le processus déclenché à chaque action.

Le lien vers le démonstrateur est : http://grapher.openeditionlab.org/.

La figure 6.17 illustre la page d'accueil dans laquelle l'utilisateur tape une requête donnée. Nous avons pris comme exemple de requête : *les droits de l'homme*. On trouve également deux boutons : *Visualiser le graphe* qui permet de donner une vue générale sur le Grapher et *A propos du grapher* qui renvoie vers une page de description du prototype de recommandation.

FIGURE 6.17 – Page d'accueil du démonstrateur (prototype de recommandation utilisant le Grapher)

Dans la figure 6.18, on trouve la page qui contient les résultats de recherche dans l'index Solr comme le montre le schéma. Il s'agit de la première étape de recherche pour répondre à la requête de l'utilisateur (``les droits de l'homme ´´). La liste affichée contient le titre du document qui est cliquable et renvoie vers la page du document dans la plateforme Revues.org. Chaque titre de document est précédé par son type (article, éditorial, compte rendu de lecture, chronique, etc.) pour plus de clarté. L'utilisateur a le choix de formuler une nouvelle requête et de relancer la recherche sur Solr ou-bien, il peut demander les recommandations

relatives à un document.

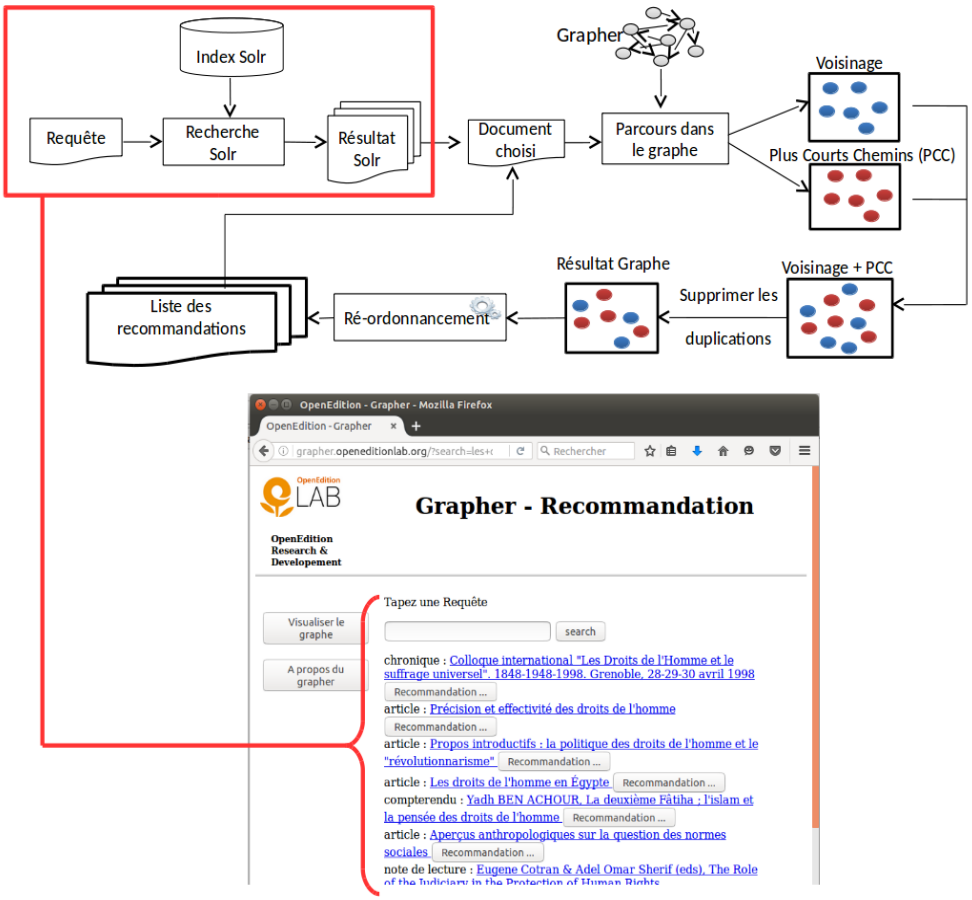


FIGURE 6.18 – Résultats de recherche avec Solr

Un bouton *Recommandation* est présent devant chaque titre. Ce bouton déclenche l'implémentation de l'algorithme 3 de recommandation basé sur le parcours dans le Grapher comme l'indique la figure 6.19. Cet action se traduit, par l'apparition d'un onglet *"Nous vous recommandons ... "* qui s'affichera en-bas de chaque document consulté. A cette étape, l'utilisateur peut redemander des recommandations pour les documents qu'il veut.

6.3.6 Protocole d'évaluation

Nous avons conçu un protocole d'évaluation qui est toujours en cours de développement. Nous nous sommes basés sur le protocole d'évaluation utilisé par la campagne INEX en 2014 KOOLEN et collab. [2014]. La différence entre les deux protocoles est que pour INEX 2014, nous disposons d'un corpus bien détaillé des requêtes (topics) avec des suggestions diverses accompagnées de l'avis de la personne qui a posé la requête. Ce dont nous disposons pour OpenEdition (Revues.org), est un corpus de requêtes simples extraites à partir des logs, nous n'avons pas de forum comme celui associé à INEX SBS (LibraryThing Forum) pour pouvoir collecter des requêtes détaillées et écrites en langage naturel.

6.3.6.1 Besoins

— Un corpus de requêtes (complexes ou simples). Ces requêtes doivent être des demandes d'un/de document(s) scientifique(s) dans le même domaine qu'OpenEdition (science

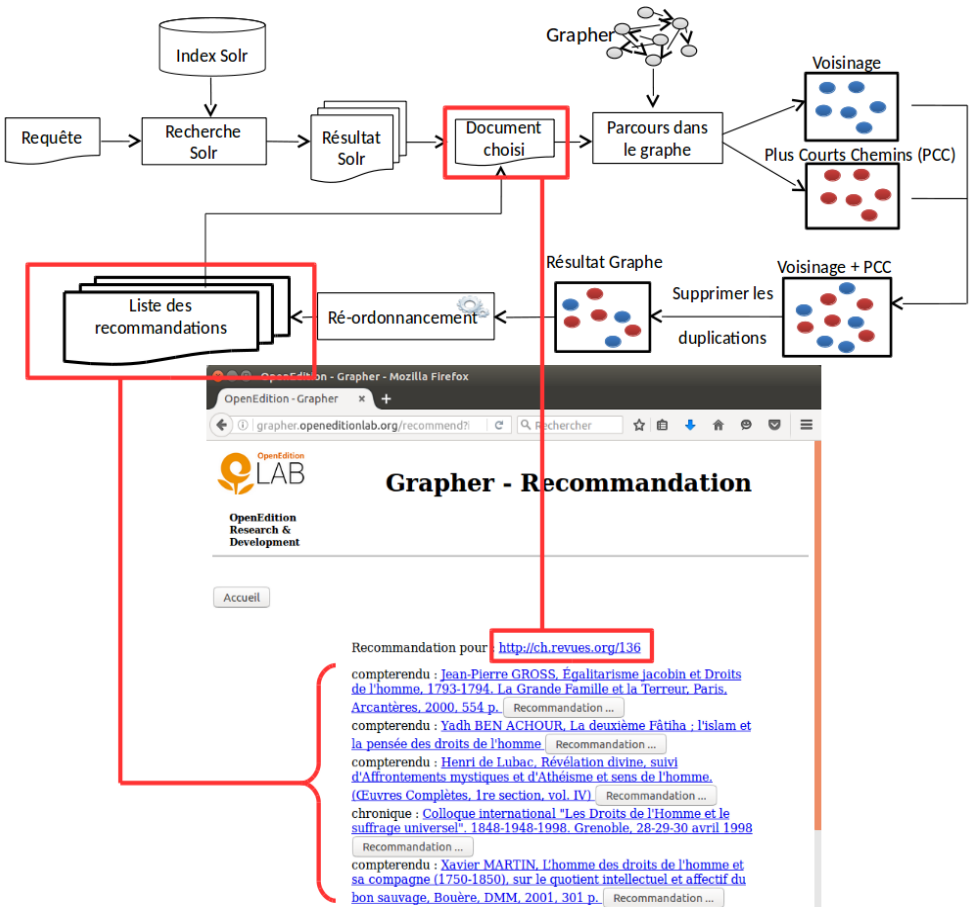


FIGURE 6.19 – Résultats de recommandation à partir du Grapher

humaine et sociale).

— Des suggestions pour chaque requête par des utilisateurs experts, ayant déjà lu leurs suggestions ou les connaissent sans les avoir lues.

— Des annotations manuelles sur chaque suggestion (pour définir le type du “suggesteur ”et son opinion sur sa(ses) suggestion(s)). Ces annotations se font par la personne qui suggère ou par d’autres personnes (annotateurs externes).

6.3.6.2 Procédé

Pour collecter les informations nécessaires pour l’évaluation, nous envisageons de créer une interface web et de la présenter à un public qui a des connaissances dans le domaines des sciences humaines et sociales (des étudiants, des professeurs, etc.). Dans cette interface, nous proposons un ensemble de requêtes et chaque utilisateur de l’interface propose des suggestions ou propose une requête pour que les autres lui proposent des recommandations. Ensuite, sur la base des données regroupées, nous procédons comme suit :

— Nous définissons un premier arbre de décision qui permet de déterminer pour chaque requête, le(s) jugement(s) qu’il faut utiliser pour calculer les valeurs de pertinence des documents suggérés parmi tous les jugements donnés par les utilisateurs qui les ont suggérés. Le jugement final sera celui d’un utilisateur expert ou une combinaison de plusieurs jugements. Cet arbre permet de construire le fichier de référence équivalent du référentiel d’INEX (qrels) qui contient les requêtes (leurs ID) et les suggestions (les

ID des documents recommandés). Dans l'étape suivante les valeurs de pertinence sont calculées pour compléter le fichier de références.

— Nous définissons un deuxième arbre de décision qui permet de définir les valeurs de pertinence selon les avis donnés sur les suggestions. La valeur initiale de pertinence d'un document est $p = 2$ qui sera modifiée selon le schéma suivant :

1. Un seul jugement :
   (a) L'utilisateur qui a posé la requête connaît la suggestion $\Rightarrow p = 0$
   (b) L'utilisateur qui a posé la requête ne connaît pas la suggestion :
      i. Il donne un avis positif $\Rightarrow p = 8$
      ii. Il donne un avis neutre $\Rightarrow p = 2$
      iii. Il donne un avis négatif $\Rightarrow p = 0$
   (c) Un autre utilisateur qui connaît et a lu la suggestion :
      i. Il donne un avis positif $\Rightarrow p = 4$
      ii. Il donne un avis neutre $\Rightarrow p = 2$
      iii. Il donne un avis négatif $\Rightarrow p = 0$
   (d) Un autre utilisateur qui connaît mais n'a pas lu la suggestion :
      i. Il donne un avis positif $\Rightarrow p = 3$
      ii. Il donne un avis neutre $\Rightarrow p = 2$
      iii. Il donne un avis négatif $\Rightarrow p = 0$
2. Plusieurs jugements :
   (a) Les utilisateurs qui ont lu la suggestion :
      i. On ne trouve que des avis positifs $\Rightarrow p = 6$
      ii. On ne trouve que des avis neutres $\Rightarrow p = 2$
      iii. On ne trouve que des avis négatifs $\Rightarrow p = 0$
      iv. #positifs > #négatifs $\Rightarrow p = 4$
      v. #positifs = #négatifs $\Rightarrow p = 2$
      vi. #positifs < #négatifs $\Rightarrow p = 1$
   (b) Les utilisateurs qui n'ont pas lu la suggestion :
      i. On ne trouve que des avis positifs $\Rightarrow p = 4$
      ii. On ne trouve que des avis neutres $\Rightarrow p = 2$
      iii. On ne trouve que des avis négatifs $\Rightarrow p = 0$
      iv. #positifs > #négatifs $\Rightarrow p = 3$
      v. #positifs = #négatifs $\Rightarrow p = 2$
      vi. #positifs < #négatifs $\Rightarrow p = 1$

Le système qu'on souhaite évaluer pour OpenEdition est un système qui inclut à la fois la Recherche d'Information et la Recommandation. Nous l'avons conçu de telle sorte qu'il prenne en entrée une requête utilisateur et retourne une liste de documents. Ensuite, à la base du Grapher, une liste de suggestions est proposée par les algorithmes de parcours. Dans ce cas, les mesures d'évaluation se baseront sur le comportement des utilisateurs comme par exemple le nombre de clics, le temps de consultation d'un document, le téléchargement du document, etc.

Un processus d'évaluation nécessite une étude de plusieurs propriétés qui influencent le système de recommandation comme par exemple :

— la prédiction des votes de l'utilisateur,

— la prédiction de l'usage de l'utilisateur (achat, catalogue, téléchargement, etc.)

— et finalement la prédiction du classement des items.

Le système de recommandation mis en place ne cherche pas à prédire les préférences des utilisateurs selon leurs centres d'intérêt mais plutôt à prédire les documents qui peuvent les intéresser. Ce qui revient à utiliser les mêmes mesures d'évaluation que celles utilisées pour INEX (Rappel@10/20/..., Précision10/20/..., MAP, NDCG@10, Recip_rank)

6.4 Conclusion

Dans ce chapitre, nous avons présenté la méthode de recommandation proposée. Elle repose sur des requêtes d'utilisateurs en premier lieu et sur la structure de graphe en second lieu. Nous avons appliqué cette méthode sur deux domaines différents. Le premier consiste à recommander des livres d'Amazon, il s'agit de la campagne d'évaluation d'INEX/CLEF, la tâche Social Book Search. Nous avons établi une modélisation des livres d'Amazon en un graphe que nous avons appelé DGD (Directed Graph of Documents). Cette modélisation se base sur des informations sociales, plus précisément sur la notion de Similar Products. Le deuxième domaine d'application est OpenEdition, une plateforme de ressources numériques dans les sciences humaines et sociales. Nous avons utilisé un réseau de citations (Grapher) construit avec les résultats du logiciel Bilbo qui permet de détecter, d'analyser et d'annoter les références bibliographiques.

Nous avons montré pour la tâche Social Book Search, que la structuration des documents en graphe améliore les performances par rapport aux méthodes de recherche classique (le modèle utilisé est InL2). Nous avons également constaté un comportement différent du système de recommandation pour quelques requêtes dans le corpus, ce qui nous a emmené à étudier les types et la nature des requêtes. Cette étude a conduit à diviser l'ensemble des requêtes en deux classes (Analogue et Non-Analogue) selon la façon dont les utilisateurs posent leurs besoins en recommandation. La méthode de recommandation basée sur le DGD a montré des améliorations significatives pour les requêtes ou les utilisateurs présentent des exemples de lectures précédentes (classe Analogue) contrairement aux autres requêtes classiques (classe Non-Analogue). Nous avons présenté également, les méthodes de réordonnancement des résultats de recommandation qui utilisent l'indice d'importance de Google, le PageRank qui a donné de meilleurs résultats par rapport au score social qui représente la moyenne des votes pour les livres.

Pour le deuxième cas d'application, dans le cadre du portail d'OpenEdition, nous avons présenté un démonstrateur du prototype de recommandation qui exploite le Grapher. Pour des raisons de manque de temps, nous n'avons pas présenté des résultats numériques. Ceci est dû principalement aux besoins nécessaires pour l'évaluation qui impose l'intervention d'un public expert dans le domaine des sciences humaines et sociales. Néanmoins, nous avons détaillé le protocole d'évaluation conçu en l'attente de collecter toutes les informations nécessaires.

Dans le futur, nous envisageons de déployer la méthode de recommandation sur les autres plateformes d'OpenEdition en commençant par les blogs scientifiques dans Hypothèses.org. Sachant que les documents dans cette dernière plateforme ne sont pas typés, nous nous intéressons à intégrer le détecteur automatique des comptes rendus de lecture présenté dans

le chapitre 4. L'idée est de proposer à un l'utilisateur qui consulte un livre donné, tous ses comptes rendus et critiques. Cela permettrait à l'utilisateur d'avoir un aperçu du livre (ou un autre type de document scientifique) selon plusieurs points de vue.

Conclusion générale et Perspectives

« *Nous ne pouvons, ni ne pourrons atteindre les sommets éclairés sans traverser les profondeurs obscures.* »

Khalil Gibran

Le travail de recherche présenté dans ce mémoire consiste à développer de nouvelles méthodes pour la recommandation de lecture en exploitant de nouvelles structures de documents. Les informations générées par les internautes sont utilisées pour recommander des lectures à partir de requêtes longues et complexes ou par rapport à un item donné. Les méthodes de RI sont utilisées dans ce contexte. Notre travail a porté sur deux domaines, la classification automatique pour détecter les avis des utilisateurs et une combinaison de la RI avec la recommandation. La classification automatique est l'une des tâches majeures dans différents domaines de recherche. Son but est d'identifier et extraire un(des) groupe(s) de données parmi d'autres.

Dans cette thèse, nous avons présenté des méthodes de classification automatique non supervisée pour identifier les comptes rendus de lecture au sein du portail OpenEdition, principalement Revues.org. Notre travail a porté sur cette dernière en raison de sa qualité de contenu (très bien annotée par des experts, contenant beaucoup de documents de différents types notamment les comptes rendus de lecture). Plusieurs méthodes de représentation des documents ont été testées (indexation) ainsi que des méthodes que nous avons proposées qui se basent sur la distribution des entités nommées dans le texte. Nous avons montré que la combinaison des caractéristiques issues de la distribution des entités nommées et celles sélectionnées avec la pondération *tf*idf* donne les meilleurs résultats sur le corpus d'évaluation utilisé.

Ce type de documents (compte rendu ou R$eview$) peut être utilisé pour la recommandation de contenus scientifiques de telle manière qu'il représente des commentaires longs et écrits généralement d'une manière formelle par des personnes expertes. En plus, il est envisageable d'intégrer un analyseur d'opinion pour les comptes rendus de lectures dans le but d'extraire les phrases de polarité positive et négative. Ces phrases seront exploitées dans le processus de recommandation final. Une telle analyse peut aboutir également à un détecteur automatique de documents qui font débat sur un sujet particulier.

Pour ce qui concerne le deuxième domaine traité dans cette thèse (la RI et la recommandation), nous avons proposé et évalué trois méthodes de recommandation basées sur des modèles de recherche d'information pour un ensemble de requêtes complexes écrites en langage naturel. La première méthode repose sur la combinaison d'un modèle de langue et d'un modèle probabiliste DFR. Dans la deuxième méthode, nous avons intégré les données produites par les utilisateurs dans le processus de recommandation en supposant qu'une recommandation basée sur l'avis d'autres utilisateurs sera plus pertinente que celle basée seulement sur le contenu. La dernière méthode repose sur la transformation de la requête initial de l'utilisateur. Cette transformation est faite par une agrégation d'un ensemble de termes à partir d'un échantillon de documents retournés par un système de recommandation de base. Cet ensemble de termes représente deux types de données : des tags d'utilisateurs ou les termes les plus informatifs selon une fonction Bo1.

Les résultats des différentes expérimentations faites sur le corpus de CLEF la tâche Social Book Search (2014 et 2015) ont validé les trois hypothèses posées initialement. Nous avons montré que la combinaison des différentes approches de recherche ne produisant pas les mêmes résultats, améliore les performances de recommandation et que l'agrégation des données sociales et la reformulation de requête par injection de pertinence peuvent engendrer des recommandations plus pertinentes par rapport au modèle de langue SDM. Cependant nos méthodes montrent une dégradation des performances par rapport au modèle probabiliste InL2.

Il faudrait peut être explorer une meilleure façon d'agrégation ou une meilleure méthode de reformulation en utilisant des ressources externes à la collection pour améliorer les per-

formances de recommandation. Ceci peut être une autre piste à explorer. En plus, il peut être envisagé d'étudier de nouvelles méthodes de reformulation de requêtes et d'intégration des informations sociales dans le processus de recommandation, plus précisément celui basé sur le modèle probabiliste InL2.

Pour améliorer les performances des méthodes de recherche, nous avons proposé une méthode de recommandation qui repose sur des requêtes d'utilisateurs en premier lieu et sur la structure de graphe en second. Cette méthode a été appliquée sur deux cas différents. Le premier consiste à recommander des livres d'Amazon, il s'agit de la campagne d'évaluation d'INEX SBS(CLEF, Social Book Search). Une modélisation des livres d'Amazon en un graphe appelé DGD (Directed Graph of Documents) est réalisée. Elle se base sur des informations sociales, plus précisément sur la notion de Similar Products. Le deuxième cas d'application est sur OpenEdition. Un réseau de citations appelé Grapher est utilisé. Ce dernier est construit avec les résultats du logiciel Bilbo qui permet de détecter, d'analyser et d'annoter les références bibliographiques dans un document scientifique.

Les résultats présentés pour la campagne INEX, ont montré que la structuration des documents en graphe peut améliorer les performances par rapport aux méthodes de recherche classique (le modèle utilisé est InL2). Il a été également constaté un comportement différent du système de recommandation pour quelques requêtes dans le corpus ce qui nous a amené à étudier les types et la nature des requêtes fournies par la campagne SBS. Cette étude a conduit à diviser l'ensemble des requêtes en deux classes (Analogue et Non-Analogue) selon la façon dont les utilisateurs posent leurs besoins en recommandation. La méthode de recommandation basée sur le DGD a montré des améliorations significatives pour les requêtes où les utilisateurs présentent des exemples de lectures précédentes (classe Analogue) contrairement aux autres requêtes classiques (classe Non-Analogue).

Nous avons présenté également, les méthodes de réordonnancement des résultats de recommandations qui utilisent l'indice d'importance de Google, le PageRank qui a donné de meilleurs résultats par rapport au score social qui représente la moyenne des votes pour les livres.

Pour le deuxième cas d'application, dans le cadre du portail d'OpenEdition. Un démonstrateur du prototype de recommandation est présenté, il exploite le Grapher construit à partir de Revues.org. Pour des raisons de manque de temps, nous n'avons pas présenté des résultats numériques. Ceci est dû principalement aux besoins nécessaires pour l'évaluation qui impose l'intervention d'un public expert dans le domaine des sciences humaines et sociales. Néanmoins, nous avons détaillé le protocole d'évaluation conçu en l'attente de collecter toutes les informations nécessaires.

En perspective la méthode de recommandation proposée peut être étalée sur les autres plateformes d'OpenEdition telle que Hypothèses.org. Sachant que les documents dans cette dernière ne sont pas typés, le détecteur automatique des comptes rendus de lecture peut être intégré. L'idée est de proposer à un utilisateur qui consulte un livre donné, tous ses comptes rendus de lecture. Cela lui permet d'avoir un aperçu sur le livre (ou un autre type de document scientifique).

Bibliographie

ADOMAVICIUS, G. et A. TUZHILIN. 2005, «Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions», *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, nº 6, p. 734–749. 26, 71

AGGARWAL, C. C., J. L. WOLF, K.-L. WU et P. S. YU. 1999, «Horting hatches an egg : A new graph-theoretic approach to collaborative filtering», dans *Knowledge Discovery and Data Mining*, p. 201–212. 32

ALI, K. et W. VAN STAM. 2004, «Tivo : Making show recommendations using a distributed collaborative filtering architecture», dans *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 31

ALLAN, J., J. ASLAM, B. CARTERETTE, V. PAVLU et E. KANOULAS. 2008, «Million query track 2008 overview», dans *Proceedings of TREC*. 74

ALPAYDIN, E. 2004, *Introduction to Machine Learning*, The MIT Press. 26

AMATI, G. et C. J. VAN RIJSBERGEN. 2002, «Probabilistic models of information retrieval based on measuring the divergence from randomness», *ACM Trans. Inf. Syst.*, vol. 20, nº 4, doi :10.1145/582415.582416, p. 357–389, ISSN 1046-8188. 74, 75, 79

ASHWIN, A., C. CLAYTON, W, P. PANOS, M et S. OLEG. 2007, «Managing network risk via critical node identification», *Risk Management in Telecommunication Networks, Springer*. 34

BARBIERI, N., M. GUARASCIO et E. RITACCO. 2010, «An empirical comparison of collaborative filtering approaches on netflix data.», dans *IIR*, *CEUR Workshop Proceedings*, vol. 560, édité par M. Melucci, S. Mizzaro et G. Pasi, CEUR-WS.org, p. 23–27. 31

BAUMANN, S. et O. HUMMEL. 2005, «Enhancing music recommendation algorithms using cultural metadata», *Journal of New Music Research*, vol. 34, nº 2. 25

BAYES, T. 1763, «An essay towards solving a problem in the doctrine of chances», *Philosophical Transactions of the Royal Society*, vol. 35, doi :10.1098/rstl.1763.0053, p. 370–418. 44

BELKIN, N. J., P. B. KANTOR, E. A. FOX et J. A. SHAW. 1995, «Combining the evidence of multiple query representations for information retrieval.», *Inf. Process. Manage.*, vol. 31, nº 3, p. 431–448. URL http://dblp.uni-trier.de/db/journals/ipm/ipm31.html#BelkinKFS95. 76

BELL, R., Y. KOREN et C. VOLINSKY. 2007, «Modeling relationships at multiple scales to improve accuracy of large recommender systems», dans *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 32

BELL, R. M. et Y. KOREN. 2007, «Lessons from the netflix prize challenge», *SIGKDD Explorations*, vol. 9, nº 2, doi :10.1145/1345448.1345465, p. 75–79. 36

BENCHETTARA, N., R. KANAWATI et C. ROUVEIROL. 2010, «Supervised machine learning applied to link prediction in bipartite social networks», dans *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, p. 326–330, doi : 10.1109/ASONAM.2010.87. 33

BENKOUSSAS, A., CHAHINEZ. OLLAGNIER et P. BELLOT. 2015a, «Book recommendation using information retrieval methods and graph analysis», dans *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. URL http://ceur-ws.org/Vol-1391/8-CR.pdf. 5, 96

BENKOUSSAS, C. et P. BELLOT. 2013, «Book recommendation based on social information.», dans *CLEF (Working Notes)*, *CEUR Workshop Proceedings*, vol. 1179, édité par P. Forner, R. Navigli, D. Tufis et N. Ferro, CEUR-WS.org. 5

BENKOUSSAS, C. et P. BELLOT. 2015b, «Cross-document search engine for book recommendation.», dans *CBRecSys@RecSys*, *CEUR Workshop Proceedings*, vol. 1448, édité par T. Bogers et M. Koolen, CEUR-WS.org, p. 42–49. 5

BENKOUSSAS, C. et P. BELLOT. 2015c, «Information retrieval and graph analysis approaches for book recommendation», *The Scientific World Journal*, vol. 2015. 5

BENKOUSSAS, C., H. HAMDAN, P. BELLOT, B. FRÉDÉRIC, E. FAATH et M. DACOS. 2014, «A collection of scholarly book reviews from the platforms of electronic sources in humanities and social sciences openedition.org», dans *9th International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavik, Islande. 54

BENKOUSSAS, H. A. S. O. A., CHAHINEZ. HAMDAN et P. BELLOT. 2014, «Collaborative filtering for book recommandation.», dans *CLEF (Working Notes)*, *CEUR Workshop Proceedings*, vol. 1180, édité par L. Cappellato, N. Ferro, M. Halvey et W. Kraaij, CEUR-WS.org, p. 501–507. 5

BENKOUSSAS, H. B. P. B. F., CHAHINEZ. HAMDAN et E. FAATH. 2014, «A collection of scholarly book reviews from the platforms of electronic sources in humanities and social sciences openedition.org.», dans *LREC*, édité par N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk et S. Piperidis, European Language Resources Association (ELRA), p. 4172–4177. URL http://dblp.uni-trier.de/db/conf/lrec/lrec2014.html#BenkoussasHBBF14. 5

BENKOUSSAS, P., CHAHINEZ. BELLOT et A. OLLAGNIER. 2015, «The impact of linked documents and graph analysis on information retrieval methods for book recommendation.», dans *WI-IAT (1)*, IEEE Computer Society, ISBN 978-1-4673-9618-9, p. 385–392. 5

BENNETT, J. et S. LANNING. 2007, «The netflix prize», dans *Proceedings of the KDD Cup Workshop 2007*, ACM, New York, p. 3–6. 36

BERGER, A. L. et J. D. LAFFERTY. 1999, «Information retrieval as statistical translation.», dans *SIGIR*, p. 222–229, doi :http://doi.acm.org/10.1145/312624.312681. 13

BERNOTAS, M., K. KARKLIUS, R. LAURUTIS et A. SLOTKIENE. 2007, «The peculiarities of the text document representation, using ontology and tagging-based clustering technique», . 51

BIBER. 1988, *Variation across Speech and writing. Cambridge : CUP*. 42

BILLSUS, D. et M. PAZZANI. dans *15th International Conference on Machine Learning*, p. 46–54. 89

BO PANG, L. L. U. S. V. 2002, «Thumbs up ? sentiment classification using machine learning techniques», . 41

BOGERS, T. et A. VAN DEN BOSCH. 2008, «Recommending scientific articles using citeulike», dans *RecSys '08 : Proceedings of the 2008 ACM conference on Recommender systems*, ACM, New York, NY, USA, ISBN 978-1-60558-093-7, p. 287–290, doi : http://doi.acm.org/10.1145/1454008.1454053. 25

BOLLACKER, K. D., S. LAWRENCE et C. L. GILES. 2000, «Discovering relevant scientific literature on the web.», *IEEE Intelligent Systems*, vol. 15, nº 2, p. 42–47. 34

BONNEFOY, L., R. DEVEAUD et P. BELLOT. 2012, «Do social information help book search ?», dans *CLEF (Online Working Notes/Labs/Workshop)*, édité par P. Forner, J. Karlgren et C. Womser-Hacker, ISBN 978-88-904810-3-1. URL http://dblp.uni-trier.de/db/conf/clef/clef2012w.html#BonnefoyDB12. 75

BONNIN, G. 2010, *Towards Robust Recommender Systems for Web Navigation : Inspiration from Statistical Language Modeling*, Theses, Université Nancy II. 31

BOUGHANEM, M., C. CHRISMENT et C. SOULE-DUPUY. 1999, «Query modification based on relevance back-propagation in an ad hoc environment», *Information Processing & Management*, vol. 35, nº 2, doi :10.1016/S0306-4573(99)00008-4, p. 121–139. 15

BOURAMOUL, A. 2014, *Recherche d'Information Contextuelle et Sémantique sur le Web : Comment augmenter la sélectivité des outils de recherche d'information sur le web*, PAF, ISBN 9783841630445. URL https://books.google.fr/books?id=1kklrgEACAAJ. 8

BRADLEY, P. S., O. L. MANGASARIAN et W. N. STREET. 1998, «Feature selection via mathematical programming.», *INFORMS Journal on Computing*, vol. 10, nº 2, p. 209–217. 46

BRIN, S. et L. PAGE. 1998, «The anatomy of a large-scale hypertextual Web search engine», *Computer Networks and ISDN Systems*, vol. 30, p. 107–117. 34

BU, J., S. TAN, C. CHEN, C. WANG, H. WU, L. ZHANG et X. HE. 2010, «Music recommendation by unified hypergraph : combining social media information and music content», dans *Proceedings of the international conference on Multimedia*, ACM, p. 391–400. 25

BURGES, C. J. 1998, «A tutorial on support vector machines for pattern recognition», *Data Mining and Knowledge Discovery*, vol. 2, p. 121–167. 45

BURKE, R. 2002, «Hybrid recommender systems : Survey and experiments.», *User Modeling and User-Adapted Interaction*, vol. 12, nº 4, p. 331–370. 31

BURKE, R. 2007, «Hybrid web recommender systems», dans *The Adaptive Web*, *Lecture Notes in Computer Science*, vol. 4321, édité par P. Brusilovsky, A. Kobsa et W. Nejdl, Springer, ISBN 978-3-540-72078-2. 25, 32

CALLAN, J. P. 1996, «Document filtering with inference networks.», dans *SIGIR*, p. 262–269. 13

CELMA, RAMIREZ et HERRERA. 2005, «foafing the music : a music recommendation system based on rss feeds and user preferances», *ismir05*. 25

CHARLE, M. 1992, «Principes et méthodes de statistique lexicale», . 49

CHARTON, E., N. CAMELIN, R. ACUNA-AGOST, P. GOTAB, R. LAVALLEY, R. KESSLER et S. FERNANDEZ. 2008, «Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour deft08», *Actes DEFT08-TALN*, vol. 8. 44

CHEE, S. H. S., J. HAN et K. WANG. 2001, *Data Warehousing and Knowledge Discovery : Third International Conference, DaWaK 2001 Munich, Germany, September 5–7, 2001 Proceedings*, chap. RecTree : An Efficient Collaborative Filtering Method, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-44801-3, p. 141–151, doi :10.1007/3-540-44801-2_15. URL http://dx.doi.org/10.1007/3-540-44801-2_15. 31

CHEKKAI, N., S. CHIKHI et H. KHEDDOUCI. 2011, «Nouvelle approche à base de graphes pour les systèmes de recommandation collaboratifs.», dans *CIIA*, *CEUR Workshop Proceedings*, vol. 825, édité par A. Amine, O. A. Mohamed, B. Benatallah et Z. Elberrichi, CEUR-WS.org. 34

CHEN, Y.-L., J.-J. WEI, S. YI WU et Y.-H. HU. 2006, «A similarity-based method for retrieving documents from the sci/ssci database.», *J. Information Science*, vol. 32, n° 5, p. 449–464. 34

CHIEN, Y.-H. et E. GEORGE. 1999, «A bayesian model for collaborative filtering», dans *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*. 31

CHINCHOR, N., E. BROWN, L. FERRO et P. ROBINSON. 1999, «Named entity recognition task definition», cahier de recherche Version 1.4, The MITRE Corporation and SAIC. 58

CLEVERDON, C. 1967, «The cranfield tests on index languages devices», . 10

CLEVERDON, C. et M. KEAN. 1968, «Factors determining the performance of indexing systems», Aslib Cranfield Research Project, Cranfield, England. 16

CLEVERDON, C., J. MILLS, M. KEEN et A. C. R. PROJECT. 1966, *Factors Determining the Performance of Indexing Systems*, n° vol. 1,ptie. 1 dans Factors Determining the Performance of Indexing Systems, College of Aeronautics. URL https://books.google.fr/books?id=XQbhAAAAMAAJ. 16

CONNOR, M. et J. HERLOCKER. 2001, «Clustering items for collaborative filtering», . 31

CORNUÉJOLS, A. et L. MICLET. 2011, *Apprentissage artificiel : Concepts et algorithmes*, Algorithmes, Eyrolles, ISBN 9782212083019. URL https://books.google.fr/books?id=FLIU5ozmY2sC. 30

CRAW, S. 2010, *Encyclopedia of Machine Learning*, chap. Manhattan Distance, Springer US, Boston, MA, ISBN 978-0-387-30164-8, p. 639–639, doi : 10.1007/978-0-387-30164-8_506. URL http://dx.doi.org/10.1007/978-0-387-30164-8_506. 33

CROFT, W. B. 1978, *Organizing and searching large files of document descriptions*, thèse de doctorat, Cambridge University. 1, 86, 98, 102

DASH, M. et H. LIU. 1997, «Feature selection for classification», *Intelligent Data Analysis*, vol. 1, nº 3. 46, 47

D'HONDT, E. 2014, «Genre classification using balanced winnow in the deft 2014 challenge», dans *TALN-RECITAL 2014 Workshop DEFT 2014 : DÉfi Fouille de Textes (DEFT 2014 Workshop : Text Mining Challenge)*, Association pour le Traitement Automatique des Langues, Marseille, France, p. 31–35. 42

DIESTEL, R. 2005, *Graph Theory*, Springer. 33

DUAN, K. et J. C. RAJAPAKSE. 2004, «A variant of svm-rfe for gene selection in cancer classification with expression data.», dans *CIBCB*, IEEE, ISBN 0-7803-8728-7, p. 49–55. 49

DUAN, W., B. GU et A. B. WHINSTON. 2008, «Do online reviews matter ? - an empirical investigation of panel data», *Decis. Support Syst.*, vol. 45, nº 4, doi :10.1016/j.dss.2008.04.001, p. 1007–1016, ISSN 0167-9236. 53

ESSEGHIR, M. A., G. GONCALVES et Y. SLIMANI. 2010, «Memetic feature selection : Benchmarking hybridization schemata.», dans *HAIS (1)*, *Lecture Notes in Computer Science*, vol. 6076, édité par M. G. Romay, E. Corchado et M. T. García-Sebastían, Springer, ISBN 978-3-642-13768-6, p. 351–358. 45, 47

FAN, R.-E., P.-H. CHEN et C.-J. LIN. 2005, «Working set selection using second order information for training support vector machines», *J. Mach. Learn. Res.*, vol. 6, p. 1889–1918, ISSN 1532-4435. 66

FANG, H. 2008, «A re-examination of query expansion using lexical resources.», dans *ACL*, édité par K. McKeown, J. D. Moore, S. Teufel, J. Allan et S. Furui, The Association for Computer Linguistics, ISBN 978-1-932432-04-6, p. 139–147. 15

JILLY FERNY. 2010, *intellegent food planiong : personalize recipe recommendation*. 25

FINN, A. et N. KUSHMERICK. 2006, «Learning to classify documents according to genre», *Journal of the American Society for Information Science and Technology*, vol. 7, nº 5. 42

FISK, D. 1997, «An application of social filtering to movie recommendation», *Lecture Notes in Computer Science*, vol. 1198, p. 116. 25

FREUND, L., C. L. A. CLARKE et E. G. TOMS. 2006, «Towards genre classification for ir in the workplace.», dans *IIiX*, édité par I. Ruthven, ACM, ISBN 1-59593-482-0, p. 30–36. 42

FREYNE, J. et S. BERKOVSKY. 2010, «Recommending food : Reasoning on recipes and ingredients.», dans *UMAP*, *Lecture Notes in Computer Science*, vol. 6075, édité par P. D. Bra, A. Kobsa et D. N. Chin, Springer, p. 381–386. 25

FÜRNKRANZ, J. 1999, «Exploiting structural information for text classification on the www.», dans *IDA*, *Lecture Notes in Computer Science*, vol. 1642, édité par D. J. Hand, J. N. Kok et M. R. Berthold, Springer, ISBN 3-540-66332-0, p. 487–498. 51

GALIBERT, O., S. ROSSET, C. GROUIN, P. ZWEIGENBAUM et L. QUINTARD. 2012, «Extended named entities annotation on ocred documents : From corpus constitution to evaluation campaign.», dans *LREC*, édité par N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk et S. Piperidis, European Language Resources Association (ELRA), ISBN 978-2-9517408-7-7, p. 3126–3131. 58

GODOY, D. et A. AMANDI. 2008, «Hybrid content and tag-based profiles for recommendation in collaborative tagging systems», dans *LA-WEB '08 : Proceedings of the 2008 Latin American Web Conference*, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-3397-1, p. 58–65, doi :http://dx.doi.org/10.1109/LA-WEB.2008.15. 27

GOLDBERG, K., T. ROEDER, D. GUPTA et C. PERKINS. 2001, «Eigenstate : A constant time collaborative filtering algorithm», *Information Retrieval*, vol. 4, n° 2, p. 133–151. 28

GUNAWARDANA, A. et C. MEEK. 2009, «A unified approach to building hybrid recommender systems», dans *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, ACM, New York, NY, USA, ISBN 978-1-60558-435-5, p. 117–124, doi : 10.1145/1639714.1639735. 31

GUYON, I., J. WESTON, S. BARNHILL et V. VAPNIK. 2002, «Gene selection for cancer classification using support vector machines.», *Machine Learning*, vol. 46, n° 1-3, p. 389–422. 45, 48, 66

HALL, M. M., H. C. HUURDEMAN, M. KOOLEN, M. SKOV et D. WALSH. 2014, «Overview of the inex 2014 interactive social book search track.», dans *CLEF (Working Notes)*, *CEUR Workshop Proceedings*, vol. 1180, édité par L. Cappellato, N. Ferro, M. Halvey et W. Kraaij, CEUR-WS.org, p. 480–493. 20

HAMZAOUI, N. 2014, *Nouvelles Techniques de Recommandation et de Détection des Communautés*, Theses, Université d'Abdelmalek Essaadi Tanger. 31

HARMAN, D. 1992, «Overview of the first text retrieval conference (trec-1).», dans *TREC*, vol. Special Publication 500-207, édité par D. K. Harman, National Institute of Standards and Technology (NIST), p. 1–20. 18

HERLOCKER, J., J. KONSTAN, L. TERVEEN et J. RIEDL. 2004, «Evaluating collaborative filtering recommender systems», *ACM Transactions on Information Systems*, vol. 22, n° 1, p. 5–53. 36

HUANG, Z., W. CHUNG, T.-H. ONG et H. CHEN. 2002a, «A graph-based recommender system for digital library», *JCDL '02*, p. 65–73. 34

HUANG, Z., W. CHUNG, T.-H. ONG et H. CHEN. 2002b, «A graph-based recommender system for digital library», *JCDL '02*, p. 65–73. URL http://delivery.acm.org/10.1145/550000/544231/p65-huang.pdf?key1=544231&key2=8265870311&coll=GUIDE&dl=ACM&CFID=56965103&CFTOKEN=1581829. 89

HUANG, Z. et Y. QIU. 2010, «A multiple-perspective approach to constructing and aggregating citation semantic link network.», *Future Generation Comp. Syst.*, vol. 26, n° 3, p. 400–407. 35

HULL, D. A. 1993, «Using statistical testing in the evaluation of retrieval experiments.», dans *SIGIR*, édité par R. Korfhage, E. M. Rasmussen et P. W. 0002, ACM, ISBN 0-89791-605-0, p. 329–338. 16

JAILLET, S., M. TEISSEIRE, J. CHAUCHE et V. PRINCE. 2003, «Classification automatique de documents, le coefficient des deux écarts», . 51

JOACHIMS, T. 1998, «Text categorization with suport vector machines : Learning with many relevant features.», dans *ECML*, *Lecture Notes in Computer Science*, vol. 1398, édité par C. Nedellec et C. Rouveirol, Springer, ISBN 3-540-64417-2, p. 137–142. 41

JOHN, G. H., R. KOHAVI et K. PFLEGER. 1994, «Irrelevant features and the subset selection problem.», dans *ICML*, édité par W. W. Cohen et H. Hirsh, Morgan Kaufmann, ISBN 1-55860-335-2, p. 121–129. 46

KARLGREN, J. et D. CUTTING. 1994, «Recognizing text genres with simple metrics using discriminant analysis», dans *Proceedings of COLING*, p. 1071–1075. 42

KASTRIN, A., T. C. RINDFLESCH et D. HRISTOVSKI. 2014, «Link prediction on the semantic medline network - an approach to literature-based discovery.», dans *Discovery Science*, *Lecture Notes in Computer Science*, vol. 8777, édité par S. Dzeroski, P. Panov, D. Kocev et L. Todorovski, Springer, ISBN 978-3-319-11811-6, p. 135–143. 33

KAZAI, G., M. KOOLEN, J. KAMPS, A. DOUCET et M. LANDONI. 2010, «Overview of the inex 2010 book track : Scaling up the evaluation using crowdsourcing.», dans *INEX*, *Lecture Notes in Computer Science*, vol. 6932, édité par S. Geva, J. Kamps, R. Schenkel et A. Trotman, Springer, p. 98–117. 40

KENT, A., M. M. BERRY, F. U. LUEHRS et J. W. PERRY. 1955, «Machine literature searching viii. operational criteria for designing information retrieval systems», *American Documentation*, vol. 6, n° 2, doi :10.1002/asi.5090060209, p. 93–101, ISSN 1936-6108. 16

KERN, R., K. JACK et M. GRANITZER. 2014, «Recommending scientific literature : Comparing use-cases and algorithms.», *CoRR*, vol. abs/1409.1357. 25

KESSLER, B., G. NUNBERG et H. SCHÜTZE. 1997, «Automatic detection of text genre», dans *Proceedings of 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain. 42

KESSLER, M. 1963, «Bibliographic coupling between scientific papers.», *American Documentation 14*, p. 10–25. 35

KHABBAZ, M., K. KIANMEHR et R. ALHAJJ. 2012, «Employing structural and textual feature extraction for semistructured document classification», *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, vol. 42, n° 6, doi : 10.1109/TSMCC.2012.2208102, p. 1566–1578, ISSN 1094-6977. 51

KIM, Y.-M., P. BELLOT, E. FAATH et M. DACOS. 2011, «Automatic annotation of bibliographical references in digital humanities books, articles and blogs.», dans *BooksOnline*, édité par G. Kazai, C. Eickhoff et P. Brusilovsky, ACM, ISBN 978-1-4503-0961-5, p. 41–48. URL http://dblp.uni-trier.de/db/conf/cikm/books2011.html#KimBFD11. 103

KIM, Y.-M., P. BELLOT, E. FAATH et M. DACOS. 2012a, «Automatic annotation of incomplete and scattered bibliographical references in digital humanities papers.», dans *CORIA*, édité par M. Beigbeder, V. Eglin, N. Ragot et M. Géry, p. 329–340. URL http://dblp.uni-trier.de/db/conf/coria/coria2012.html#KimBFD12. 103

KIM, Y.-M., P. BELLOT, J. TAVERNIER, E. FAATH et M. DACOS. 2012b, «Evaluation of bilbo reference parsing in digital humanities via a comparison of different tools.», dans *ACM Symposium on Document Engineering*, édité par C. Concolato et P. Schmitz, ACM, ISBN 978-1-4503-1116-8, p. 209–212. URL http://dblp.uni-trier.de/db/conf/doceng/doceng2012.html#KimBTFD12. 69, 103

KOOLEN, M., T. BOGERS, M. GÄDE, M. HALL, H. HUURDEMAN, J. KAMPS, M. SKOV, E. TOMS et D. WALSH. 2015a, *Experimental IR Meets Multilinguality, Multimodality, and Interaction : 6th International Conference of the CLEF Association, CLEF'15, Toulouse, France, September 8-11, 2015, Proceedings*, chap. Overview of the CLEF 2015 Social Book Search Lab, Springer International Publishing, Cham, ISBN 978-3-319-24027-5, p. 545–564, doi :10.1007/978-3-319-24027-5_51. URL http://dx.doi.org/10.1007/978-3-319-24027-5_51. 20

KOOLEN, M., T. BOGERS, M. GÄDE, M. A. HALL, H. C. HUURDEMAN, J. KAMPS, M. SKOV, E. TOMS et D. WALSH. 2015b, «Overview of the clef 2015 social book search lab.», dans *CLEF*, *Lecture Notes in Computer Science*, vol. 9283, édité par J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. J. F. Jones, E. SanJuan, L. Cappellato et N. Ferro, Springer, ISBN 978-3-319-24026-8, p. 545–564. URL http://dblp.uni-trier.de/db/conf/clef/clef2015.html#KoolenBGHHKSTW15. 99

KOOLEN, M., T. BOGERS, J. KAMPS, G. KAZAI et M. PREMINGER. 2014, «Overview of the inex 2014 social book search track.», dans *CLEF (Working Notes)*, *CEUR Workshop Proceedings*, vol. 1180, édité par L. Cappellato, N. Ferro, M. Halvey et W. Kraaij, CEUR-WS.org, p. 462–479. URL http://dblp.uni-trier.de/db/conf/clef/clef2014w.html#KoolenBKKP14. 76, 99, 108

KOOLEN, M., G. KAZAI, J. KAMPS, M. PREMINGER, A. DOUCET et M. LANDONI. 2012, «Overview of the inex 2012 social book search track.», dans *CLEF (Online Working Notes/Labs/Workshop)*, *CEUR Workshop Proceedings*, vol. 1178, édité par P. Forner, J. Karlgren et C. Womser-Hacker, CEUR-WS.org, ISBN 978-88-904810-3-1. URL http://dblp.uni-trier.de/db/conf/clef/clef2012w.html#KoolenKKPDL12. 76

KOOLEN, M., G. KAZAI, M. PREMINGER et A. DOUCET. 2013, «Overview of the inex 2013 social book search track.», dans *CLEF (Working Notes)*, *CEUR Workshop Proceedings*, vol. 1179, édité par P. Forner, R. Navigli, D. Tufis et N. Ferro, CEUR-WS.org. 20

KOREN, Y. 2009, «The bellkor solution to the netflix grand prize», . 37

KUMMER, O. 2012, *Feature Weighting Approaches in Sentiment Analysis of Short Text*, thèse de doctorat, Université de Neuchâtel. 49, 50

KUO, W.-T., Y.-C. WANG, R. T.-H. TSAI et J. Y. JEN HSU. 2015, «Contextual restaurant recommendation utilizing implicit feedback.», dans *WOCC*, IEEE, ISBN 978-1-4799-8854-9, p. 170–174. 25

LAI, S., Y. LIU, H. GU, L. XU, K. LIU, S. XIANG, J. ZHAO, R. DIAO, L. XIANG, H. LI et D. WANG. 2012, «Hybrid recommendation models for binary user preference prediction problem.», dans *KDD Cup*, *JMLR Proceedings*, vol. 18, édité par G. Dror, Y. Koren et M. Weimer, JMLR.org, p. 137–151. 29

LAINÉ-CRUZEL, S. 1999, «Profildoc : Filtrer une information exploitable», *Bulletin des bibliothèques de France [en ligne] n° 5 [consulté le 20 avril 2016]*, vol. ISSN 1292-8399. 27

LAL, T., O. CHAPELLE, J. WESTON et A. ELISSEEFF. 2006, *Embedded methods*, Studies in Fuzziness and Soft Computing ; 207, Springer, Berlin, Germany, p. 137–165. 48

LAROUM, S., N. BÉCHET, H. HAMZA et M. ROCHE. 2010, «Classification automatique de documents bruités à faible contenu textuel», *Revue des Nouvelles Technologies de l'Information*, vol. E-18, nº Numéro spécial : Fouille de Données Complexes, p. 25. 46

LECLUZE et LEJEUNE. 2014, «nalyse automatique de textes littéraires et scientifiques en langue française(stylometrie et quelques catégories lexicales)», dans *TALN-RECITAL 2014 Workshop DEFT 2014 : DÉfi Fouille de Textes (DEFT 2014 Workshop : Text Mining Challenge)*, Association pour le Traitement Automatique des Langues, Marseille, France. 42

LEE, J. H. 1995, «Combining multiple evidence from different properties of weighting schemes», dans *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, ACM, New York, NY, USA, ISBN 0-89791-714-6, p. 180–188, doi :10.1145/215206.215358. URL http://doi.acm.org/10.1145/215206.215358. 77

LEE, J. H. 1997, «Analyses of multiple evidence combination», *SIGIR Forum*, vol. 31, nº SI, p. 267–276. URL http://doi.acm.org/10.1145/278459.258587. 75, 87

LEIMSTOLL, U. et H. STORMER. 2007, «Collaborative recommender systems for online shops», dans *Proceedings of the Thirteenth Americas Conference on Information Systems*. 71

LEVI, A., O. MOKRYN, C. DIOT et N. TAFT. 2012, «Finding a needle in a haystack of reviews : cold start context-based hotel recommender system demo.», dans *RecSys*, édité par P. Cunningham, N. J. Hurley, I. Guy et S. S. Anand, ACM, ISBN 978-1-4503-1270-7, p. 305–306. 28

LI, G.-Z., J. Y. 0002, G.-P. LIU et L. XUE. 2004, «Feature selection for multi-class problems using support vector machines.», dans *PRICAI*, *Lecture Notes in Computer Science*, vol. 3157, édité par C. Zhang, H. W. Guesgen et W.-K. Yeap, Springer, ISBN 3-540-22817-9, p. 292–300. 49

LI, Q. et B. M. KIM. 2003, «An approach for combining content-based and collaborative filters», dans *Proceedings of the Sixth international workshop on Information retrieval with Asian languages (ACL-2003)*, p. 17–24. 32

LI, Q. et B. M. KIM. 2004, «Constructing user profiles for collaborative recommender system.», dans *APWeb*, *Lecture Notes in Computer Science*, vol. 3007, édité par J. X. Yu, X. Lin, H. Lu et Y. Zhang, Springer, ISBN 3-540-21371-6, p. 100–110. 27

LIANG, Y., Q. LI et T. QIAN. 2011, «Finding relevant papers based on citation relations.», dans *WAIM*, *Lecture Notes in Computer Science*, vol. 6897, édité par H. Wang, S. Li, S. Oyama, X. Hu et T. Qian, Springer, ISBN 978-3-642-23534-4, p. 403–414. 35, 36

LINDEN, G., J. JACOBI et E. BENSON. 2001, «Collaborative recommendations using item-to-item similarity mappings», URL https://www.google.com/patents/US6266649. 90

LINDEN, G., B. SMITH et J. YORK. 2003, «Amazon.com recommendations. item-to-item collaborative filtering», *IEEE Internet Computing*, vol. 7, nº 1, p. 76–80. 30, 31, 90

LIU, H. et L. YU. 2005, «Toward integrating feature selection algorithms for classification and clustering.», *IEEE Trans. Knowl. Data Eng.*, vol. 17, nº 4, p. 491–502. 47, 48

MAHMOOD, T. et F. RICCI. 2009, «Improving recommender systems with adaptive conversational strategies.», dans *Hypertext*, édité par C. Cattuto, G. Ruffo et F. Menczer, ACM, ISBN 978-1-60558-486-7, p. 73–82. 25

MARON, M. et J. KUHNS. 1960, «On relevance, probabilistic indexing and information retrieval.», *Journal of the ACM 7*, p. 216–244. 12

MARSHAKOVA, I. 1973, «System of document connections based on references», *Nauchno-Tekhnicheskaya Informatsiya Seriya 2-Informatsionnye Protsessy I Sistemy*, vol. 6, p. 3–8. 35

MCNEE, S., I. ALBERT, D. COSLEY, P. GOPALKRISHNAN, A. M. RASHID, J. A. KONSTAN et J. RIEDL. 2002, «On the recommending of citations for research papers», dans *Proceedings of ACMCSCW 2002*, ACM Press, New York. 36

MEHLITZ, M., C. BAUCKHAGE, J. KUNEGIS et S. ALBAYRAK. 2007, «A new evaluation measure for information retrieval systems», dans *Proc. Int. Conf. on Systems, Man and Cybernetics*. 16

MEMMI, D. 2000, «Le modèle vectoriel pour le traitement de documents», dans *Cahiers Leibniz 2000-14*, INPG. 43

METZLER, D. et W. B. CROFT. 2005, «A markov random field model for term dependencies», dans *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, ACM, New York, NY, USA, ISBN 1-59593-034-5, p. 472–479, doi :10.1145/1076034.1076115. URL http://doi.acm.org/10.1145/1076034.1076115. 74

METZLER, D., T. STROHMAN et W. B. CROFT. 2006, «Indri trec notebook 2006 : Lessons learned from three terabyte tracks.», dans *TREC*, vol. Special Publication 500-272, édité par E. M. Voorhees et L. P. Buckland, National Institute of Standards and Technology (NIST). URL http://dblp.uni-trier.de/db/conf/trec/trec2006.html#MetzlerSC06. 74

MILLER, G. 1995, «Wordnet A lexical database for English», *Communications of ACM*, vol. 38, nº 11, p. 39–41. 14

MIYAHARA, K. et M. J. PAZZANI. 2000, «Collaborative filtering with the simple bayesian classifier.», dans *PRICAI*, p. 679–689. 31

MOONEY, R. J. et L. ROY. 1999, «Content-based book recommending using learning for text categorization», dans *Proceedings of the ACM SIGIR Workshop Recommender Systems : Algorithms and Evaluation*. 25

MUI, L., P. SZOLOVITS et C. ANG. 2001, «Collaborative sanctioning : applications in restaurant recommendations based on reputation.», dans *Agents*, p. 118–119. 25

NANBA, H. et M. OKUMURA. 1999, «Towards multi-paper summarization using reference information.», dans *IJCAI*, édité par T. Dean, Morgan Kaufmann, ISBN 1-55860-613-0, p. 926–931. 35

NAVIGLI, R. et P. VELARDI. 2003, «An analysis of ontology-based query expansion strategies», dans *Workshop on Adaptive Text Extraction and Mining, (Cavtat Dubrovnik, Croatia, Sept 23)*. 15

NOBATA, C., S. SEKINE et H. ISAHARA. 2003, «Evaluation of features for sentence extraction on different types of corpora», dans *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 29–36, doi : 10.3115/1119312.1119316. 46

NORIKO, K. et P. K. MAKOTO. 2016, «Overview of ntcir-12», dans *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, National Institute of Standards and Technology (NIST). 19

OBERLANDER, J. et S. NOWSON. 2006, «Whose thumb is it anyway ? classifying author personality from weblog text», dans *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, Australia, p. 627–634. 41

OLLAGNIER, A., S. FOURNIER, P. BELLOT et B. FRÉDÉRIC. 2014, «Impact de la nature et de la taille des corpus d'apprentissage sur les performances dans la détection automatique des entités nommées», dans *TALN*. 59

OUNIS, I., G. AMATI, V. PLACHOURAS, B. HE, C. MACDONALD et C. LIOMA. 2006, «Terrier : A High Performance and Scalable Information Retrieval Platform», dans *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*. 80

OUNIS, I., G. AMATI, P. V., B. HE, C. MACDONALD et JOHNSON. 2005, «Terrier Information Retrieval Platform», dans *Proceedings of the 27th European Conference on IR Research (ECIR 2005)*, *Lecture Notes in Computer Science*, vol. 3408, Springer, ISBN 3-540-25295-9, p. 517–519. 80

OUNIS, I., C. LIOMA, C. MACDONALD et V. PLACHOURAS. 2007, «Research directions in terrier : a search engine for advanced retrieval on the web», *Novatica/UPGRADE Special Issue on Web Information Access*. 80

PAGE, L., S. BRIN, R. MOTWANI et T. WINOGRAD. 1998, «The pagerank citation ranking : Bringing order to the web», dans *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, p. 161–172. URL citeseer.nj.nec.com/page98pagerank.html. 94

PAGE, L., S. BRIN, R. MOTWANI et T. WINOGRAD. 1999, «The pagerank citation ranking : Bringing order to the web», cahier de recherche, Stanford University. 89

PAZZANI, M. et D. BILLSUS. 2007, «Content-based recommendation systems», dans *The Adaptive Web*, *Lecture Notes in Computer Science*, vol. 4321, édité par P. Brusilovsky, A. Kobsa et W. Nejdl, Springer, Berlin / Heidelberg, p. 325–341. 26

PETERS, C., éd.. 2001, *Cross-Language Information Retrieval and Evaluation, Workshop of Cross-Language Evaluation Forum, CLEF 2000, Lisbon, Portugal, September 21-22, 2000, Revised Papers*, *Lecture Notes in Computer Science*, vol. 2069, Springer. URL http://dblp.uni-trier.de/db/conf/clef/clef2000.html. 19

PETRENZ, P. et B. L. WEBBER. 2011, «Stable classification of text genres.», *Computational Linguistics*, vol. 37, nº 2, p. 385–393. 42

PICOT-CLÉMENTE, R. 2011, *A generic framework for recommender systems generating combination of items : application to the tourism domain*, Theses, Université de Bourgogne. 25, 26

PISETTA, V., H. HACID, F. BELLAL, G. RITSCHARD et D. A. ZIGHED. 2006, *Traitement automatique de textes juridiques*, Actes de la Semaine de la Connaissance. ID : unige :4537 ; http ://sdc2006.org. 43

POIBEAU, T. 2003, «The multilingual named entity recognition framework.», dans *EACL*, The Association for Computer Linguistics, p. 155–158. 59

POIRIER, D. 2011, *From texts to recommendation*, Theses, Université d'Orléans. 29

PONTE, J. et W. CROFT. 1998, «A language modeling approach to information retrieval», , p. 275–281. 13

POULIQUEN, B. 2002, *Medical texts indexation using concepts extraction, and its use*, thèse de doctorat, Université Rennes 1. 43

PRONOZA, E. V., E. YAGUNOVA et S. VOLSKAYA. 2014, «Corpus-based information extraction and opinion mining for the restaurant recommendation system.», dans *SLSP*, *Lecture Notes in Computer Science*, vol. 8791, édité par L. Besacier, A. H. Dediu et C. Martín-Vide, Springer, ISBN 978-3-319-11396-8, p. 272–284. 25

PU, L. et B. FALTINGS. 2013, «Understanding and improving relational matrix factorization in recommender systems.», dans *RecSys*, édité par Q. Y. 0001, I. King, Q. Li, P. Pu et G. Karypis, ACM, ISBN 978-1-4503-2409-0, p. 41–48. 30

RAMASWAMY, S., P. TAMAYO, R. RIFKIN, S. MUKHERJEE, C.-H. YEANG, M. ANGELO, C. LADD, M. REICH, E. LATULIPPE, J. P. MESIROV, T. POGGIO, W. GERALD, M. LODA, E. S. LANDER et T. R. GOLUB. 2001, «Supplementary information : Multiclass cancer diagnosis using tumor gene expression signatures», Technical report. 48

RATPRASARTPORN, N., J. PO, A. CAKMAK, S. BANI-AHMAD et G. ÖZSOYOGLU. 2009, «Context-based literature digital collection search.», *VLDB J.*, vol. 18, nº 1, p. 277–301. 34

RICCI, F., L. ROKACH et B. SHAPIRA. 2011a, *Introduction to recommender systems handbook*, Springer. 26

RICCI, F., L. ROKACH et B. SHAPIRA. 2011b, *Recommender Systems Handbook*, chap. Introduction to Recommender Systems Handbook, Springer US, Boston, MA, ISBN 978-0-387-85820-3, p. 1–35, doi :10.1007/978-0-387-85820-3_1. URL http://dx.doi.org/10.1007/978-0-387-85820-3_1. 25

ROBERTSON, S. E., C. J. VAN RIJSBERGEN et M. F. PORTER. 1980, «Probabilistic models of indexing and searching», dans *SIGIR*, p. 35–56. 75

ROBERTSON, S. E. et S. WALKER. 1999, «Okapi/keenbow at trec-8.», dans *TREC*, vol. Special Publication 500-246, édité par E. M. Voorhees et D. K. Harman, National Institute of Standards and Technology (NIST). 13

ROBERTSON, S. E., S. WALKER, S. JONES, M. HANCOCK-BEAULIEU et M. GATFORD. 1994, «Okapi at trec-3.», dans *TREC*, vol. Special Publication 500-225, édité par D. K. Harman, National Institute of Standards and Technology (NIST), p. 109–126. 13

ROCCHIO, J. 1971, «Relevance feedback in information retrieval», *The SMART retrieval system : experiments in automatic document processing*, p. 313–323. 78

ROCCHIO, JR., J. J. 1971, «Relevance feedback in information retrieval», dans *The SMART Information Retrieval System*, Prentice Hall, p. 313–323. 15, 16

SAID, A., S. BERKOVSKY et E. W. D. LUCA. 2010, «Putting things in context : Challenge on context-aware movie recommendation», dans *Proceedings of the Workshop on Context-Aware Movie Recommendation*, ACM, New York, NY, USA, ISBN 978-1-4503-0258-6, p. 2–6, doi :10.1145/1869652.1869665. 25

SALTON, G. 1968, «A comparison between manual and automatic indexing methods», cahier de recherche, Université de Cornell. 12

SALTON, G. 1971, *The SMART Retrieval System - Experiments in Automatic Document Processing*, Prentice Hall. 12

SALTON, G. 1973, «Recent studies in automatic text analysis and document retrieval», *J. ACM*, vol. 20, nº 2, doi :10.1145/321752.321757, p. 258–278, ISSN 0004-5411. 11

SALTON, G. et C. BUCKLEY. 1988, «Term-weighting approaches in automatic text retrieval», *Information Processing and Management*, vol. 24, nº 5, p. 513–523. 46

SALTON, G. et C. BUCKLEY. 1990, «Improving retrieval performance by relevance feedback», *Jornal of the American Society for Information Science*, vol. 41, nº 4, p. 288–297. 15

SALTON, G. et E. FOX. 1983, «Extended boolean information retrieval», *Communications of the ACM*, vol. 26, nº 11, p. 1022. 17

SALTON, G. et M. J. MCGILL. 1983, *Introduction to Modern Information Retrieval*, McGraw Hill. 12

SALTON, G. et M. J. MCGILL. 1986, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA. 51

SALTON, G., A. WONG et C. YANG. 1975, «A vector space model for automatic indexing», *Commun. ACM*, vol. 18, nº 11, p. 613–620. 43

SANG, E. F. et F. D. MEULDER. 2003, «Introduction to the CoNLL-2003 Shared Task : Language-Independent Named Entity Recognition», dans *Proceedings of CoNLL-2003*, Edmonton, Canada, p. 142–147. 58

SARWAR, B., G. KARYPIS, J. KONSTAN et J. RIEDL. 2000, «Analysis of recommendation algorithms for e-commerce», p. 158–167. Proc. ACM Conference on Electronic Commerce. 31

SARWAR, B. M., G. KARYPIS, J. KONSTAN et J. REIDL. 2002, «Recommender systems for large-scale e-commerce : Scalable neighborhood formation using clustering», dans *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT)*. 31

SARWAR, B. M., G. KARYPIS, J. A. KONSTAN et J. REIDL. 2001, «Item-based collaborative filtering recommendation algorithms», dans *World Wide Web*, p. 285–295. 30

SAVOY, J. et O. ZUBAREYVA. 2010, «Classification automatique d'opinions dans la blogosphère», dans *10th International Conference, Journée d'Analyse statistique des Données Textuelles (JADT2010)*, Sapienza University of Rome. 49

SAWANT, S. 2013, «Collaborative filtering using weighted bipartite graph projection : A recommendation system for yelp», dans *Proceedings of the CS224W : Social and Information Network Analysis Conference*. 33

SCHAFER, J. B., J. KONSTAN et J. RIEDL. 1999, «Recommender systems in e-commerce», dans *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, ACM, New York, NY, USA, ISBN 1-58113-176-3, p. 158–166, doi :10.1145/336992.337035. URL http://doi.acm.org/10.1145/336992.337035. 71

SCHWARTZ, M. et D. WOOD. 1993, «Discovering shared interests using graph analysis», dans *Communications of the ACM*, vol. 36, p. 78–89. 33

SEBASTIANI, F. et C. N. D. RICERCHE. 2002, «Machine learning in automated text categorization», *ACM Computing Surveys*, vol. 34, p. 1–47. 41, 42

SEKI, Y., D. K. EVANS, L.-W. KU, H.-H. CHEN, N. KANDO et C.-Y. LIN. 2007, «Overview of opinion analysis pilot task at ntcir-6.», dans *NTCIR*, édité par N. Kando, National Institute of Informatics (NII). 50

SHAH, C. 2014, «Collaborative information seeking.», *JASIST*, vol. 65, n° 2, p. 215–236. URL http://dblp.uni-trier.de/db/journals/jasis/jasis65.html#Shah14. 77

SHANG, M.-S., Y. FU et D.-B. CHEN. 2008, «Personal recommendation using weighted bipartite graph projection», dans *Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference on*, p. 198–202, doi :10.1109/ICACIA.2008.4770004. 33

SHAW, W. M., R. BURGIN et P. HOWELL. 1996, «Performance standards and evaluations in ir test collections : Cluster-based retrieval models», *Information Processing and Management*, vol. 33, n° 1, p. 1–14. 17

SHIVASHANKAR, B.SIVAKUMAR et G.VARAPRASAD. 2012, «Identification of Critical Node for the Efficient Performance in Manet», *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 3, nº 1. 34

SMALL, H. 1973, «Co-citation in the scientific literature : a new measure of the relationship between two documents.», *Journal of the American Society for Information Science*, vol. 24, nº 4, p. 265–269. 35

STAMATATOS, E., N. FAKOTAKIS et G. KOKKINAKIS. 2000, «Text genre detection using common word frequencies», dans *Proceedings of the 18th international conference on computational linguistics*, vol. 2, p. 808–14. 41

SU, X. et T. M. KHOSHGOFTAAR. 2009, «A survey of collaborative filtering techniques», *Adv. in Artif. Intell.*, vol. 2009, doi :http://dx.doi.org/10.1155/2009/421425, p. 4 :2–4 :2, ISSN 1687-7470. 31

SUGIYAMA, K., K. HATANO et M. YOSHIKAWA. 2004, «Adaptive web search based on user profile constructed without any effort from users», dans *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, ACM, New York, NY, USA, ISBN 1-58113-844-X, p. 675–684, doi :10.1145/988672.988764. 27

TANG, J., J. ZHANG, J. X. YU, Z. YANG, K. CAI, R. MA, L. ZHANG et Z. SU. 2009, «Topic distributions over links on web.», dans *ICDM*, édité par W. Wang, H. Kargupta, S. Ranka, P. S. Yu et X. Wu, IEEE Computer Society, ISBN 978-0-7695-3895-2, p. 1010–1015. 35

TENG, C., Y.-R. LIN et L. A. ADAMIC. 2011, «Recipe recommendation using ingredient networks», *CoRR*, vol. abs/1111.3919. 25

TURTLE, H. R. et W. B. CROFT. 1990, «Inference networks for document retrieval.», dans *SIGIR*, édité par J.-L. Vidick, ACM, ISBN 0-89791-408-2, p. 1–24. 13

TURTLE, H. R. et W. B. CROFT. 1991, «Evaluation of an inference network-based retrieval model.», *ACM Trans. Inf. Syst.*, vol. 9, nº 3, p. 187–222. 13

UNGAR, L. et D. FOSTER. 1998, «Clustering methods for collaborative filtering», dans *Proceedings of the Workshop on Recommendation Systems*, AAAI Press, Menlo Park California. 31

VAPNIK, V. N. 1995, *The Nature of Statistical Learning Theory*, Springer. 44, 45, 48

DE VEL, O., A. ANDERSON, M. CORNEY et G. MOHAY. 2001, «Mining e-mail content for author identification forensics», *SIGMOD Rec.*, vol. 30, nº 4, doi :10.1145/604264.604272, p. 55–64, ISSN 0163-5808. 41

VINOT, R., N. GRABAR et M. VALETTE. 2003, «Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'internet», dans *Actes de TALN 2003*. 43

VLACHOS, M. et D. SVONAVA. 2013, «Recommendation and visualization of similar movies using minimum spanning dendrograms.», *Information Visualization*, vol. 12, nº 1, p. 85–101. 25

VOLINSKY, C. 2009, «Matrix factorization techniques for recommender systems», p. 30–37. 31

VOORHEES, E. 2002, «The philosophy of information retrieval evaluation», dans *In Proceedings of the The Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, Springer-Verlag, Berlin, Heidelberg, p. 355–370. 16

VOORHEES, E. M. 1994, «Query expansion using lexical-semantic relations.», dans *SIGIR*, p. 61–69. 14, 15

VUCETIC, S. et Z. OBRADOVIC. 2005, «Collaborative filtering using a regression-based approach.», *Knowl. Inf. Syst.*, vol. 7, nº 1, p. 1–22. 31

WILLETT, P. 2006, «The porter stemming algorithm : then and now.», *Program*, vol. 40, nº 3, p. 219–223. URL http://dblp.uni-trier.de/db/journals/program/program40.html#Willett06. 73

YIN, Z., M. GUPTA, T. WENINGER et J. HAN. 2010, «A unified framework for link recommendation using random walks», dans *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, p. 152–159, doi :10.1109/ASONAM.2010.27. 33

ZHAI, C. et J. D. LAFFERTY. 2004, «A study of smoothing methods for language models applied to information retrieval», *ACM Transactions on Information Systems*, vol. 22, nº 2, p. 179–214. 1, 74

ZHOU, T., J. REN, M. C. V. MEDO et Y.-C. ZHANG. 2007, «Bipartite network projection and personal recommendation», *Phys. Rev. E*, vol. 76, doi :10.1103/PhysRevE.76.046115, p. 046 115. 33

ZUBARYEVA, O. et J. SAVOY. 2010, «Evaluation de modéles de classification automatique appliqués à la détection d'opinions.», dans *CORIA*, p. 271–286. 51