PARIS 8 UNIVERSITE VINCENNES - SAINT-DENIS

LIASD LABORATOIRE INFORMATIQUE AVANCÉE DE SAINT-DENIS Université Paris 8

École doctorale : Sciences sociales

THÈSE

pour obtenir le grade de docteur délivré par

Université Paris 8 Vincennes à Saint-Denis

Spécialité doctorale *Informatique*

présentée et soutenue publiquement par

Nourredine ALIANE

le 17 mai 2019

Évaluation des représentations vectorielles de mots

Directeur de thèse : Professeur Gilles BERNARD, U. PARIS 8, LIASD

Jury

M. Arab Ali Chérif,	Professeur	U. Paris 8	LIASD	Examinateur
M. Younès Bennani,	Professeur	U. Paris 13	LIPN	Examinateur
M. Kurosh Madani,	Professeur	U. Paris 12	LISSI	Rapporteur
M. Jean-Jacques Mariage,	MCF	U. Paris 8	LIASD	Examinateur
M. Benoît Sagot,	CR, HDR	INRIA	ALMAnaCH	Rapporteur

Université Paris 8
Laboratoire d'Informatique Avancée de Saint Denis
EA n° 4383 Saint Denis, France

Remerciements

L'aboutissement de ce travail de thèse est le fruit d'échanges, de conseils, de coopérations enrichissantes et de soutiens d'un grand nombre de personnes. Je tiens à adresser ma profonde reconnaissance à tout ceux qui ont contribué à ce projet ou qui m'ont encouragé pendant ces années de thèse.

Je tiens tout d'abord à exprimer ma gratitude et mes sincères remerciements à mon directeur de thèse Monsieur Gilles Bernard, Professeur à l'université de Paris 8, et à Monsieur Jean-Jacques Mariage, Maitre de Conférences à l'université de Paris 8, pour leur disponibilité, leur soutien, leurs conseils avisés, leurs encouragements ainsi que pour leur patience à mon égard, ou encore pour leur capacité à m'amener toujours plus loin dans la réflexion. Ce que j'ai appris en travaillant avec eux ne se limite pas à l'aspect scientifique mais s'étend aux aspects humain et relationnel. Je les remercie infiniment.

Je voudrais, aussi, exprimer mes remerciements aux membres du jury d'avoir accepté d'évaluer mon travail de thèse, et du temps qu'ils ont y consacré. Mes vifs remerciements en particulier aux rapporteurs, Monsieur Kurosh Madani, Professeur à l'université Paris-Est Créteil Val de Marne (UPEC) ainsi que Monsieur Benoît Sagot, Chargé de Recherche à l'Inria, qui ont accepté de juger mon travail. Je remercie également Monsieur Arab Ali Cherif, Professeur à l'université de Paris 8, Monsieur Younès Bennani, Professeur à l'université de Paris 13 et Monsieur Jean-Jacques Mariage Maître de conférence à l'université de Paris 8, pour avoir accepté de juger ma thèse et pour l'intérêt qu'ils ont porté à mon travail. Mes remerciements vont à tous les membres du laboratoire LIASD. À Monsieur Arab Ali Cherif, le directeur du laboratoire, pour son accueil chaleureux, aux enseignants, collègues et amis pour leurs soutiens, conseils et encouragements : Jean Méhat, Patrick Greussay, Jacqueline Signorini, Anna Pappa, Larbi Boubchir, Herman Akdag, Nicolas Jouandeau, Boubaker Daachi, Youcef Touati, Rakia Jaziri, Farès Belhadj,Jean-Noel Vittaut, Abdelkader Berrouachedi, Otman Manad, Adelle Abdallah, Georges Lebboss, Kathia Chenane, Amine Ganibardi, Rabah Mazouzi, Lisa Medrouk, Josué Melka, Youssef Zaki, Akram Hacini, El-Hanafi Teguig, Fathé Manseur, Jérôme Tisset…

Je remercie affectueusement et chaleureusement ma famille, en particulier mes parents, mes frères et mes soeurs pour leurs soutien et aide continuelle.

Mes pensées vont aussi et particulièrement à ma femme Amel qui était à mes côtés dans les moments de joie et de difficultés et qui m'a entourée d'affection. Qu'elle trouve ici l'expression de mes remerciements pour sa patience et son soutien inestimable.

Enfin, à tous ceux que je n'ai pas pu citer, auxquels je réitère mes sincères remerciements.

Sommaire

Résumé

La problématique centrale de ce travail est l'évaluation des méthodes de représentation vectorielle des mots, qui sont pratiquement un passage obligé pour de très nombreuses applications de traitement du langage humain (NLP : Natural Language Processing), comme : l'étiquetage des unités sémantiques ou syntaxiques, l'identification des entités nommées, la classification de textes, l'analyse des sentiments, etc.

Nous avons travaillé sur la portabilité des méthodes d'évaluation actuellement utilisées à d'autres langues que l'anglais, sur la comparaison des méthodes d'évaluation, et sur l'élaboration de méthodes d'évaluation nouvelles. Dans ce contexte, nous nous sommes particulièrement intéressé à une ressource linguistique existant, parfois à l'état larvaire, mais en constant développement, dans de nombreuses langues, WordNet.

Nous avons étudié l'ensemble des méthodes d'évaluation, avec un regard particulier sur la langue concernée, et WordNet, avec ses potentialités d'évaluation encore très peu utilisées. À partir de là, nous avons élaboré cinq méthodes d'évaluation nouvelles : quatre fondées sur WordNet, et une évaluation interne par substitution qui ne nécessite aucune donnée de référence issue de jugements humains. Plus ces cinq methodes, nous avons proposé un nouveau protocole d'évalution par sondage.

Mots clés : Évalution de représentations vectorielles de mots, Relations sémantiques, WordNet, Synsets, Représentations vectorielles de mots, Catégorisation des mots.

Abstract

In Natural Language Processing vectorization of words is a key that enables the use of algorithms based on mathematical models. Recently new methods have appeared, and evaluating their quality is a necessity. At present, evaluations are mostly effective on English, which introduces the question of multilingual evaluations. We worked on generalizing methods, on comparing them, on devising new evaluations, and on WordNet as a multilingual resource used for evaluation.

We choose six vectorization methods : CBOW, SkipGram, GloVe, an older method as baseline, and two more recent methods. Evaluations can be direct, comparing with some gold standard, or indirect, evaluating the result of an application produced with some vectorization. As an indirect method, we choose semantic clustering of words for comparing the underlying vectorizations. The chosen clustering algorithms were : the most used Kmeans, a neuronal one (SOM) and a probabilistic one (EM).

Our system applies evaluation methods on big corpora in English, French and Arabic, then compares underlying vectorizations. We propose five new evaluation methods, with four based on WordNet, and one new protocol for polling. Our results yield three different vectorization orderings agreeing on decisive points, and invalidate some existing evaluations. As for our own evaluations, the protocol is validated, one method is invalidated and the reason analyzed, one is validated for English and French, but not Arabic, two are validated on the three languages, and one is left for further exploration.

Keywords: Semantics Relations, Princeton WordNet, WOLF, Arabic WordNet, Synsets, Word Vectorization, Word Clustering, Word Embedding, Word Representation Evaluation, Multilingual Evaluation.

Introduction

Dans la recherche actuelle sur le traitement automatique des langues, la représentation vectorielle des mots est une question clé, car elle permet l'application d'algorithmes informatiques fondés sur des modèles mathématiques qui permettent l'extraction d'informations sémantiques sur le vocabulaire et sur les documents (ou pages web) à partir de masses de données. Les avancées récentes sur cette question ont permis l'émergence de nouvelles méthodes de représentation.

De cette situation découle l'importance d'évaluer la qualité de ces représentations, en elles-mêmes et pour les comparer entre elles. Les évaluations actuelles portent principalement sur l'anglais, et posent donc le problème de réaliser des méthodes d'évaluation prenant en compte la diversité des langues.

Notre thèse porte sur l'évaluation des méthodes de représentation vectorielle de mots. Nous avons travaillé sur la portabilité des méthodes d'évaluation actuellement utilisées à d'autres langues que l'anglais, sur la comparaison des méthodes d'évaluation, et sur l'élaboration de méthodes d'évaluation nouvelles. Dans ce contexte, nous nous sommes particulièrement intéressé à une ressource linguistique existant, parfois à l'état larvaire, mais en constant développement, dans de nombreuses langues, WordNet.

Nous avons étudié l'ensemble des méthodes de représentation vectorielle des mots (que nous décrivons aussi largement que possible dans la première partie de notre état de l'art) et avons choisi six méthodes : les trois qui constituent aujourd'hui la référence du domaine (CBOW, SkipGram et GloVe), une méthode plus ancienne qui nous sert de point de référence (la méthode utilisée par WebSOM), et deux méthodes plus récentes fondées sur des principes différents, partiellement (FastText) ou totalement (GraPaVec).

Les méthodes d'évaluation se répartissent en deux catégories, les méthodes directes, qui comparent la représentation avec un *gold standard* (données de référence), et les méthodes indirectes, qui évaluent le résultat d'une application produite par cette représentation. Comme méthode indirecte, nous avons choisi l'évaluation de la catégorisation sémantique de mots, que nous avons appelée évaluation semi-directe en comparaison avec des méthodes indirectes qui évaluent

par exemple une catégorisation de documents fondée sur cette catégorisation de mots.

La catégorisation sémantique de mots repose sur des algorithmes de *clustering* (classification automatique non supervisée), objets de la deuxième partie de notre état de l'art. Notre objectif n'étant pas d'évaluer ces algorithmes pour eux-mêmes mais seulement pour la comparaison des évaluations de représentations vectorielles sous-jacentes, nous avons choisi trois algorithmes représentant des types différents, celui qui est le plus utilisé actuellement (Kmeans dans sa version Kmeans++), un algorithme neuronal (SOM), et un algorithme probabiliste (EM).

Enfin nous avons étudié l'ensemble des méthodes d'évaluation directe (troisième partie de notre état de l'art), avec un regard particulier sur la langue concernée, et WordNet, avec ses potentialités d'évaluation encore très peu utilisées.

Nous avons élaboré un système qui permet d'appliquer la totalité des évaluations directes et les évaluations semi-directes choisies sur des corpus conséquents (en millions et en milliards de mots), pour comparer des représentations vectorielles entre elles et évaluer leur qualité. Nous avons élaboré cinq méthodes d'évaluation nouvelles, dont quatre fondées sur WordNet, et une évaluation par sondage sur un nouveau protocole.

À partir de là, nous évaluons les six méthodes de représentation vectorielle citées sur trois langues, l'anglais, le français et l'arabe, sur plusieurs corpus chacune. Cela a représenté des milliers d'expérimentations différentes, en raison de la multiplicité des paramètres disponibles dans chaque cas (paramètres de prétraitement du corpus, de représentation vectorielle, d'évaluation directe, de clustering, et paramètres de la comparaison), avec une combinatoire conséquente. Nous exposons une sélection des résultats des expériences réalisées, ainsi que les observations les plus générales que nous ayons pu faire, comme le fait que le calcul de la proximité des représentations est meilleur avec le cosinus de l'angle qu'avec la distance euclidienne, ou que le modèle de clustering EM avait toujours des résultats de moins bonne qualité.

Nous obtenons au final un classement des six méthodes, qui indique clairement que la méthode choisie comme base de référence est la plus mauvaise (ce qui était attendu mais permet de vérifier que les classements sont valides). Une large majorité des évaluations donne l'ordre «CBOW > SkipGram > GloVe > FastText > WebSOM», une minorité non négligeable l'ordre «SkipGram > CBOW > GloVe > FastText >WebSOM», et une très petite minorité «GloVe > CBOW > SkipGram > FastText > WebSOM». Concernant GraPaVec, certaines évaluations le mettent au dessus de WebSOM, d'autres en dessous, mais la faible quantité de données dont nous avons disposé rendent ces résultats peu fiables.

Ces variations posent la question de la qualité des méthodes d'évaluation, car

une divergence peut signifier que l'une des méthodes est mal conçue ou qu'elle n'évalue pas la même chose. Cette question reste ouverte.

Concernant nos propres méthodes d'évaluation : le protocole de sondage est validé par l'expérimentation. Des cinq méthodes élaborées, une a été invalidée (évaluation directe à partir de WordNet), et nous avons pu analyser les causes de l'échec. Une deuxième (évaluation semi-directe à partir d'un gold standard tiré de WordNet) a été validée pour l'anglais et le français, mais pas pour l'arabe (en raison de la pauvreté de Arabic WordNet). Deux autres (évaluation semi-directe sans gold standard et évaluation par substitution) ont été validées sur les trois langues. La dernière (évaluation topologique) n'a pu être ni validée ni invalidée et reste une question ouverte.

Notre contribution est donc, à part l'état de l'art présenté selon une perspective originale et les enseignements sur les paramètres, qui devraient intéresser les chercheurs dans ce domaine, de deux évaluations totalement validées et de deux évaluations validées sous condition de l'état du WordNet, après vérification sur trois langues.

0.1 Guide de lecture

Ce mémoire se présente en deux parties et six chapitres. La première partie explore la problématique et l'état de l'art, la deuxième partie présente le système réalisé et les expérimentations. Dans la première partie, le premier chapitre explore la problématique du domaine dans ses grandes lignes. Les chapitres suivants explorent les états de l'art des trois grands domaines que dégage cette problématique : le deuxième sur les représentations vectorielle, le troisième sur la catégorisation sémantique, le quatrième sur l'évaluation directe et indirecte des représentations.

La deuxième partie porte sur notre contribution, en deux chapitres ; dans le premier chapitre nous présentons le système réalisé et les méthodes implémentées, dans le deuxième chapitre nous présentons nos expérimentations et les résultats obtenus.

Nous terminons par une conclusion qui récapitule les points les plus notables et propose des pistes pour la poursuite de la recherche en ce domaine.

Première partie

Problématique et état de l'art

Sommaire

Chapitre 1

Problématique

Sommaire

1.1 Objectif

La problématique centrale de ce travail est l'évaluation des méthodes de représentation vectorielle des mots. Notre objectif est d'élaborer un protocole d'évaluation des représentations vectorielles de mots, autant que possible adapté à toutes les langues.

Les représentations vectorielles de mots sont un point de passage obligé pour de très nombreuses applications de traitement du langage humain (*NLP : Natural Language Processing*), comme l'étiquetage des unités sémantiques ou syntaxiques, l'identification des entités nommées, la classification de textes, l'analyse des sentiments, etc. Il s'agit d'associer à chaque mot d'un corpus un vecteur qui permette de le plonger dans un espace vectoriel et de lui appliquer les techniques de classification, de clustering, d'indexation, etc. qui sont disponibles dans de tels espaces.

Au cours des dernières années, les techniques de représentation vectorielle de mots et en particulier les techniques dites de *word embedding* ont entraîné une amélioration constante des performances de ces applications. En même temps, la multiplication et la complexification de ces représentations vectorielles a fait de la question de l'évaluation de leur qualité une question cruciale.

La vérification de la qualité des vecteurs produits est une tâche difficile, qui implique généralement un travail manuel, avec un calcul de corrélation, en utilisant des jeux de données (*datasets*) de référence appelés ici *gold standard*, produits par des experts. Les chercheurs sont même souvent amenés, en particulier pour les langues autres que l'anglais, à produire leur propre gold standard. La réalisation de ces données de références se fait en général à l'aide de méthodes expérimentales issues des sciences cognitives (psychologie, psycholinguistique…).

Voici quelques exemples de gold standards (voir le chapitre 3 pour un relevé détaillé) : évaluation de la similitude avec WordSim-353 (Finkelstein et al., 2001) ou SimLex-999 (Hill, Reichart et Korhonen, 2015), évaluation de paires ou de quadruplets analogiques avec TOEFL (Landauer et Dutnais, 1997), Google's analogy dataset et MSR's analogy dataset (Mikolov, Yih et Zweig, 2013).

Le premier atelier RepEval2016[1] sur l'évaluation des représentations vectorielles de mots a eu lieu en 2016, lors de la réunion annuelle de l'Association for Computational Linguistics[2] (ACL). Cet atelier a fourni de nombreuses pistes de recherche intéressantes. Il a notamment permis de mettre en évidence les problèmes les plus importants dans le domaine (Faruqui et al., 2016). Parmi ces problèmes, la difficulté de savoir quel type de relations sémantiques les relations

1. `https://aclanthology.coli.uni-saarland.de/volumes/proceedings-of-the-1st-workshop-on-evaluating-vector-space-representations-for-nlp`.
2. `https://www.aclweb.org/portal/what-is-cl`

entre vecteurs de mots permettent de détecter. En effet il existe de nombreux types de relations sémantiques entre les mots (voir plus bas), sachant que leurs définitions sont généralement assez ambiguës.

La première comparaison de la performance de diverses vectorisations a été publiée par McNamara (2011). Auparavant les comparaisons ne concernaient que les applications de ces représentations, comme l'indexation documentaire ou la classification de textes. Baroni, Bernard et Kruszewski (2014) ont proposé un aperçu complet des méthodes d'évaluation de représentations vectorielles de mots. Puis, en 2015, Schnabel et al. (2015) ont distingué deux sortes d'évaluation, évaluation directe et indirecte, distinction que nous retenons ici.

- **Évaluation directe :** on part d'un jeu de données contenant des n-uples de mots dont on connaît la relation (elle a été établie à la main) et on vérifie si les vecteurs correspondants ont la même relation.
- **Évaluation indirecte :** on utilise différentes méthodes de vectorisation dans le cadre d'une application de TLN et on compare les évaluations globales du système obtenues avec chacune, les autres paramètres restant identiques.

1.2 Évaluation directe

Dans les évaluations directes, on peut distinguer deux sous-catégories, en reprenant et adaptant Turney (2006), qui distingue les similitudes attributionnelles et les similitudes relationnelles.

- **Évaluation directe attributionnelle :** Il s'agit d'évaluer les attributs (les composantes des vecteurs) en comparant les similitudes des vecteurs avec les similitudes des mots dans les gold standards. On doit vérifier si des mots synonymes ont bien des vecteurs similaires.
- **Évaluation directe relationnelle :** Il s'agit d'évaluer les relations entre des vecteurs en les comparant aux relations entre les mots dans les gold standards. On doit vérifier si les analogies qui figurent dans le gold standard (par exemple la relation entre «maçon» et «pierre» est analogue à la relation entre «charpentier» et «bois»), se retrouvent dans les relations entre les vecteurs correspondants.

  On peut encore distinguer, dans l'évaluation directe relationnelle, entre les relations sémantiques, comme dans l'exemple ci-dessus, et les relations grammaticales, comme singulier / pluriel, masculin / féminin, etc.

Bien entendu, les gold standards pour les deux types d'évaluation ne sont pas construits de la même façon (on trouvera plus de détail dans le chapitre 3).

Les méthodes d'évaluation directe posent plusieurs problèmes, en particulier concernant les gold standards. Hill, Reichart et Korhonen (2015), entre autres, contestent leur fiabilité, car ils ne distinguent pas la similitude sémantique (*semantic similarity*) et la parenté sémantique (*semantic relatedness*). Généralement les scores attribués aux relations de parenté sont plus élevés que les scores attribués aux relations de similitude. Ils ont donc créé un nouveau gold standard, SimLex-999, pour surmonter les problèmes identifiés.

Mais pour Avraham et Goldberg (2016), il reste d'autres problèmes inhérents aux jeux de données et aux méthodes d'évaluation qui ne sont pas résolus par SimLex-999. On peut ajouter à ces problèmes le biais potentiel dans l'évaluation du fait que certains de ces jeux de données ont été créés pour évaluer telle technique de représentation.

1.3 Relations sémantiques

Les problèmes soulevés par ces critiques montrent que la notion de relation sémantique est probablement encore trop floue pour que l'évaluation directe soit vraiment fiable. Il y a déjà cette distinction entre similitude et parenté, présentée par Pedersen, Patwardhan et Michelizzi (2004) et Hill, Reichart et Korhonen (2015).

- La similitude sémantique est une forme de synonymie, comme entre «crainte» et «peur», «simple» et «facile».
- La parenté sémantique peut être une relation hiérarchique, comme entre «voiture» et «véhicule», «banane» et «fruit», d'inclusion («roue - voiture, vêtements - armoire») ou d'opposition («petit - grand, jour - nuit»).

En fait, il faut probablement disposer d'évaluations directes beaucoup plus fines. Les différentes relations sémantiques qui peuvent exister entre les mots sont connues depuis longtemps par les linguistes. Le tableau 1.1, extrait de Lebboss (2016, chapitre 2), donne un aperçu de cette complexité.

Relation	**Définition**
Monosémie	Un mot monosémique n'a qu'un seul sens, stable dans tous ses emplois (Crystal, 2011).

Polysémie	« *Deux caractéristiques permettent de définir la polysémie ([KLE99], p.55) : (i) une pluralité de sens liée à une seule forme, (ii) des sens qui ne paraissent pas totalement disjoints, mais se trouvent unis par tel ou tel rapport.* » (Jacquet, Venant et Victorri, 2005, p.3).
Métaphore	« *l'approche de la metaphore à partir du mot repose sur de très anciennes représentations métalinguistiques. La définition de la métaphore que donne Dumarsais : "La métaphore est une figure par laquelle on transporte, pour ainsi dire, la signification propre d'un mot à une autre signification qui ne lui convient qu'en vertu d'une comparaison qui est dans l'esprit".* » (Liidi, 1991, p.17)
Métonymie	La métonymie est une figure de style qui consiste à désigner un objet ou une idée par un autre terme que celui qui convient (par glissement de sens).
Synecdoque	Une synecdoque est une figure de style dans laquelle un terme désignant une partie de quelque chose renvoie à l'ensemble de la chose, ou vice versa.
Homonymie	« *Les deux lexies L1 et L2 sont homonymes si elles sont assiciées aux mêmes signifiants, mais ne possèdent aucune intersection de sens notable.* » (Polguère, 2003, p.126).
Homographie	Un homographe est un mot qui partage la même forme écrite avec un autre mot, mais a un sens différent, Exemple : mer, mère et maire(Polguère, 2003).
Homophonie	Un homophone est un mot qui se prononce de la même façon qu'un autre mot, mais ils diffèrent par le sens (Bodson, 2005).
Synonyme	Les synonymes sont des mots avec les mêmes sens ou des sens similaires, exemple : maison, logis, domicile (Moeschler, 2013).
Antonyme	Un antonyme est un mot appartenant à une paire de mots de sens opposés, exemple : jeune # vieux. L'antonyme peut être divisé en deux parties : antonymes complémentaires, antonymes gradables (Moeschler, 2013).
Antonymes complémentaires	Un antonyme complémentaire est un mot appartenant à une paire de mots de significations opposées, où les deux sens ne se trouvent pas sur un spectre continu, exemple : mort # vivant (Moeschler, 2013).

Antonymes gradables	Un antonyme gradable est un mot appartenant à une paire de mots de sens opposés lorsque les deux sens se trouvent sur un spectre continu, exemple : chaud tiède # froid (Moeschler, 2013).
Hyperonymie	Y est un hyperonyme de X si chaque X est une sorte de Y, c'est-à-dire le sens de X est inclus dans le sens de Y, exemple : animal est un hyperonyme de chien (Polguère, 2003).
Hyponymie	Y est un hyponyme de X si chaque Y est une sorte de X, c'est-à-dire le sens de Y est inclus dans le sens de X, exemple : le chien est un hyponyme de l'animal (Polguère, 2003).
Holonymie	Holonymie définit la relation entre un terme désignant l'ensemble et un terme désignant une partie, ou un membre de l'ensemble. C'est-à-dire X est un holonyme de Y si Ys sont parties de Xs ou Y est un holonyme de X si Ys sont membres de Xs, exemple : bras est un holonyme de main (Bodson, 2005).
Meronymie	Un méronyme représente un élément constitutif, ou un membre de quelque chose. Cela signifie que X est un méronyme de Y si Xs sont des parties de Ys, exemple : main est un méronyme de bras (Bodson, 2005).
Implication	« *L'implicaion lexicale est la relation entre deux verbes quand on peut dire que la phrase quelqu'un V-1s implique logiquement la phrase quelqu'un V-2s. Par exemple Snore (ronfler) implique sleep (dormir)* » (Fellbaum, 1999, p.30).
Cause	« *Cette relation lie deux verbes ou deux synsets comme donne et avoir, tels que l'un dénote une cause et l'autre le résultat. L'anglais a quelques paires causatives lexicalisées comme show et see ou raise et rise ; ces paires sont liées dans WordNet par un pointeur* » (Fellbaum, 1999, p.33).
Attribut	Un attribut est une propriété qui a des valeurs décrites par des adjectifs (Fellbaum, 1999).
Dérivation	La dérivation d'un mot à partir d'un autre se fait en ajoutant des affixes (préfixes, infixes, ou suffixes) à la racine selon plusieurs modes (Fellbaum, 1999).

Tab. 1.1: Extrait de Lebboss (2016) : relations lexicales

1.4 Évaluation indirecte

L'évaluation indirecte peut se faire au travers de nombreuses applications de NLP. Dans le cadre de cette thèse, nous avons choisi de nous pencher sur l'évaluation au travers de la catégorisation sémantique de mots à partir d'un grand corpus textuel. Il s'agit donc d'une évaluation indirecte, mais en relation tout de même plus directe avec les données que le clustering documentaire par exemple. Cette évaluation semi-directe nous permet de mieux étudier et comprendre la nature de la sémantique distributionnelle des mots.

Bien qu'une compréhension profonde, semblable à celle de l'humain, reste difficile à atteindre, de nombreuses méthodes ont permis au moins de récupérer certains aspects de la similitude entre mots. Il est important de noter que plusieurs tâches de NLP ont été résolues à l'aide de modèles sémantiques distributionnels. En conséquence, ces tâches peuvent donner lieu à une mesure indirecte suffisamment fiable de la performance des représentations vectorielles de mots (Turney et Pantel, 2010).

L'évaluation indirecte peut être divisée en deux sous-catégories, l'évaluation externe et l'évaluation interne.

Avec l'évaluation externe, on évalue les résultats du système réalisé à l'aide d'un gold standard. Il existe des gold standards différents pour différents types d'application (analyse syntaxique, tagging, classification, etc.). Un des intérêts d'une évaluation semi-directe est la disponibilité de jeux de données existants ayant été réalisés dans un autre but que pour l'évaluation de ces représentations, et possédant de nombreux autres usages, en particulier les ontologies.

L'évaluation interne, concernant la catégorisation sémantique, consiste essentiellement à examiner les catégories produites en fonction de la manière dont les données se répartissent dans chaque catégorie.

1.5 Approche adoptée

Pour les différentes raisons mentionnées, nous avons choisi de privilégier l'évaluation semi-directe par la catégorisation sémantique des mots, et externe à l'aide d'un gold standard général. Pour autant, nous n'avons pas exclu de notre champ d'investigation les évaluations directes et les évaluations semi-directes internes.

Nous nous sommes tout particulièrement intéressés, pour le gold standard, à WordNet, une sorte d'ontologie générique, qui, avec ses équivalents dans d'autres langues et le réseau de liens entre eux, constitue un jeu de données multilingue qu'on ne peut guère soupçonner d'introduire un biais en faveur de tel ou tel algo-

rithme. Cette ontologie est constituée de relations sémantiques fines du type de celles indiquées dans le tableau 1.1, ce qui permet, en perspective, d'envisager une poursuite des recherches dans cette direction.

En effet un dernier aspect important de notre problématique est d'obtenir des évaluations susceptibles de s'appliquer à toutes les langues pour lesquelles un corpus suffisant est constituable, sans utiliser, par conséquent, de connaissances spécifiques à la langue du corpus. Nous avons choisi, pour nos expérimentations, de travailler sur trois langues : le français, l'anglais et l'arabe standard moderne. Ces langues ont des situations diverses. Pour l'arabe, nous nous appuyons en particulier sur les travaux de Georges Lebboss (Lebboss, 2016).

La catégorisation sémantique de mots passe par trois étapes fondamentales.

1. La production d'une représentation des mots ; il s'agit de produire une représentation, en général vectorielle, de taille raisonnable, qui permette de plonger les mots dans un espace mathématique.

   Nous avons choisi de comparer cinq techniques : Websom, les deux techniques proposées dans Word2Vec (SkipGram et CBOW), Glove, et GraPaVec : Word2Vec et Glove sont les méthodes de référence aujourd'hui. GraPaVec est une méthode récente, développée dans notre laboratoire, fondée sur un type de distribution complètement différent, et Websom est une méthode plus ancienne choisie à titre de comparaison. Nous avons également fait quelques tests avec une sixième technique, FastText.

2. Les techniques de clustering : il s'agit de produire des catégories en groupant les représentations de mots en fonction de leur distribution, sans supervision. Rappelons que le clustering pour nous est simplement le contexte d'usage, un moyen pour l'évaluation des représentations et non une fin en soi.

   Parmi la grande quantité de modèles, nous avons choisi ici de sélectionner trois méthodes, Kmeans++ pour sa popularité et sa simplicité, SOM pour sa topologie sur les clusters qui permet d'examiner non seulement les relations sémantiques à l'intérieur des clusters mais aussi les relations sémantiques entre clusters, et Expectation Maximisation, pour tester une catégorisation à base probabiliste.

3. Les techniques d'évaluation : il s'agit de valider les catégories produites à la deuxième étape par des évaluations internes ou externes au modèle de clustering.

   Pour l'évaluation interne, qui consiste à étudier la distribution des vecteurs dans l'espace vectoriel, nous explorons l'inertie ainsi que l'erreur de quantization et U-matrix.

Pour l'évaluation externe, nous avons exploré les *gold standard* WordNet, Wolf et Arabic WordNet, ainsi que les quelques gold standards existants. Notre évaluation utilise différentes métriques.

Pour conclure, notre étude porte sur la relation entre la première et la troisième étape et cherche à évaluer les représentations vectorielles produites dans la première étape à partir des évaluations produites à la troisième étape. Nous comparerons ces évaluations avec les évaluations directes obtenues à l'aide des différents gold standards disponibles, pour les langues où ils existent.

Les chapitres suivants approfondissent la problématique esquissée ici et explorent l'état de l'art de chacune des trois étapes mentionnées, relativement à notre objectif.

Chapitre 2

Représentation des mots

Sommaire

2.1 Introduction

Dans ce chapitre, nous étudions les techniques de représentation qui permettent de plonger les mots dans un espace mathématique.

La plupart de ces méthodes reposent sur la sémantique distributionnelle définie par Harris : «les mots apparaissant dans des contextes similaires ont un sens similaire» (Harris, 1954). Par exemple dans le texte ci-dessous en italique, le mot « magazine » et le mot « journal » sont considérés comme similaires sémantiquement car ils partagent les mêmes contextes :

> *Aujourd'hui, je lisais un* ***magazine****....Le* ***magazine*** *a publié un article.....Il achète ce* ***magazine*** *tous les jours... .Aujourd'hui, je lisais un* ***journal****.....Le* ***journal*** *a publié un article.....Il achète ce* ***journal*** *tous les jours...*

Les modèles de représentation vectorielle des mots ou modèles sémantiques distributionnels transforment l'analyse distributionnelle d'un corpus en espace vectoriel, dans lequel deux vecteurs proches géométriquement représentent deux mots dont la sémantique est proche.

Si nous voulons savoir à quel point le mot « magazine » et le mot « journal » sont similaires, nous calculons la similitude cosinus ou une distance entre leurs vecteurs en fonction de la métrique choisie.

Les modèles sémantiques distributionnels peuvent être divisés en deux catégories :

- Modèles statistiques : connus aussi sous le terme anglophone de *count based models*, la construction des vecteurs mots se fait par un calcul statistique de collocations entre les mots et leurs contextes. Le contexte peut être un mot, un ensemble de mots, une phrase, un paragraphe ou un document entier.
- Modèles prédictifs (*word embedding* en anglais) : ces modèles se focalisent sur l'apprentissage d'une représentation vectorielle de mots. Les données d'apprentissages sont des couples $(mot, contexte)$ extraits à partir d'un corpus textuel. On utilise différentes techniques d'apprentissage, souvent un réseau neuromimétique.

2.2 Modèles statistiques

Les méthodes statistiques sont basées essentiellement sur le calcul d'une matrice de co-occurrences des mots et de leurs contextes, par exemple $mots \times documents$ ou $mots \times mots$.

Les lignes de la matrice de co-occurrences représentent les vecteurs de chaque mot (en toute rigueur, moyennant une transposition), et les colonnes représentent les vecteurs de chaque contexte.

Même pour un corpus de taille modeste, la matrice de co-occurrences comporte plusieurs dizaines de milliers de lignes (nombre de mots du vocabulaire) et de colonnes (nombre de contextes), avec beaucoup de valeurs nulles, en raison du fait que le sous ensemble des mots qui apparaît dans un contexte donné est très petit. Il s'agit donc d'une matrice creuse (*sparse matrix*).

Avec un corpus de taille conséquente, il faut compter en millions de lignes, voire en milliards, et en autant de colonnes si le contexte est un mot, ou en milliers de colonnes en contexte documentaire.

Par conséquent on rencontre très vite le problème de la taille de la matrice, à la fois pour des raisons techniques (temps calcul et espace mémoire) et pour des raisons méthodologiques (exploitabilité des résultats par exemple).

Dans cette section, nous allons d'abord présenter deux types de modèles qui sont à l'origine de la plupart des recherches en méthodes statistiques, le modèle à indexation, avec comme exemple le modèle de Salton, et le modèle à cooccurrences, avec comme exemple *Hypertext Analogue to Language*, puis l'information ponctuelle mutuelle, qui remplace la fréquence dans des variantes de ces deux modèles. Ensuite nous présenterons une sélection des méthodes de réduction de la dimensionnalité de la matrice.

2.2.1 Indexation : Modèle de Salton

Ce modèle a été proposé pour la première fois dans les années 1970, par Gerard Salton (Salton, Wong et Yang, 1975). Bien qu'ancien, il reste utilisé dans la plupart des moteurs de recherche d'informations récents. Ce modèle est basé sur une méthode algébrique. Les contextes sont des documents textuels, et le modèle a été utilisé principalement pour indexer des documents.

Il représente les documents par des vecteurs de valeurs réelles, où chaque composante correspond au poids sémantique d'un terme dans un document.

Il y a trois méthodes principalement utilisées dans ce modèle pour construire la matrice de cooccurrences *mots* × *documents* à partir de laquelle on déduira le poids sémantique des mots dans les documents.

- La méthode binaire consiste à affecter 1 au poids du terme s'il est présent dans le document, ou 0 s'il n'est pas présent. Cette méthode est très peu robuste, puisqu'elle considère comme de même poids un mot qui n'apparaît qu'une fois et un mot qui apparaît souvent dans un document donné.

- La méthode *TF (term frequency)* est basée sur le calcul de la fréquence du terme (son nombre d'occurrences dans le document). Plus cette valeur est grande, plus le terme est supposé être important.

- La méthode *TF-IDF (term frequency - inverse document frequency)* est un correctif apporté à la précédente méthode pour prendre en compte la distribution du mot dans l'ensemble des documents. En effet un mot qui a la même fréquence dans tous les documents n'aura aucun pouvoir discriminant entre les documents. On utilise la «fréquence des documents» pour un mot donné, en calculant le nombre de documents où ce mot apparaît divisé par le nombre total de documents.

  Pour calculer le poids sémantique *tf.idf* d'un terme j dans un document i, on utilise la formule suivante :

$$tfidf(i,j) = tf(i,j) * \log(\frac{N}{df(j)}) \tag{2.1}$$

  $tf(i,j)$ = nombre d'occurrences de j dans i, $df(j)$ = nombre de documents où j apparaît, N = nombre total de documents.

  Plus $df(j)$ est grand, plus $tfidf(i,j)$ est petit. Cela signifie que si un terme apparaît dans beaucoup de documents, il n'a pas beaucoup de poids sémantique.

On éliminera les termes dont le poids sémantique est faible.

Enfin, pour comparer les documents, on calcule la similitude cosinus entre leurs vecteurs (plus le cosinus est élevé plus la similitude est grande) ou des mesures de distance (plus la distance est grande plus la similitude est faible).

On peut également, même si le modèle n'a pas été élaboré pour cela, comparer les vecteurs des termes par les mêmes méthodes ; il suffit pour cela de transposer la matrice.

Formule du cosinus pour comparer deux documents, i et j:

$$\cos(\vec{i},\vec{i}) = \frac{\vec{i}.\vec{j}}{\|\vec{i}\|.\|\vec{j}\|} \tag{2.2}$$

En raison de la loi de Zipf, dès qu'un corpus est important, la majorité des mots n'ont qu'un nombre d'occurrences très faible (entre 50% et 60% d'*hapax*, par exemple, et jusqu'à 90% des mots ont un nombre d'occurrences inférieur à 5).

En conséquence, nous allons avoir des vecteurs creux de grande dimension. Pour diminuer ce phénomène, avant de construire la matrice, le modèle vectoriel

utilise un dictionnaire de termes construit à partir de l'ensemble de documents. La figure 2.1 montre le fonctionnement du modèle de Salton.

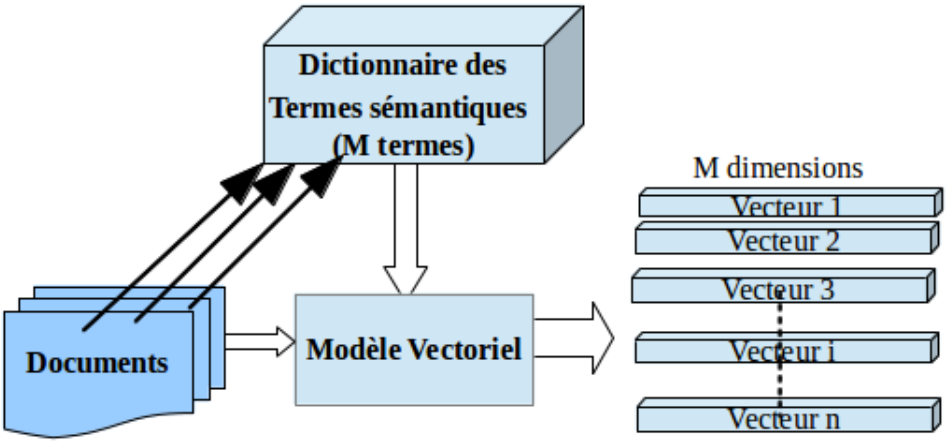

FIG. 2.1: Fonctionnement du modèle de Salton

2.2.1.1 Exemple

Nous considérons un corpus de quatre documents $\{d_1, d_2, d_3, d_4\}$, et un vocabulaire de trois mots : $\{le, chat, boit\}$. Les trois méthodes ci-dessus donneront trois matrices différentes. La table 2.1 est calculée par la méthode binaire, la table 2.2 représente la pondération TF, et la table 2.3 la pondération TF-IDF.

Documents / Mots	d_1	d_2	d_3	d_4
le	1	1	1	1
chat	1	0	0	1
boit	1	0	1	1

TAB. 2.1: Méthode binaire

Documents / Mots	d_1	d_2	d_3	d_4
le	3	6	8	11
chat	2	0	0	5
boit	4	0	6	12

TAB. 2.2: Méthode TF

Documents / Mots	d_1	d_2	d_3	d_4	IDF
le	0	0	0	0	0
chat	0,6	0	0	1,5	0,3
boit	0,48	0	0,72	1,44	0,12

TAB. 2.3: Méthode TF-IDF

Dans la table 2.3, l'IDF montre que le mot «le» ne permet pas de distinguer les documents, et que le mot «chat» est plus discriminant que le mot «boit». Ce dernier mot est sans importance (poids sémantique nul) dans les documents d_2 et d_3, et est plus important dans le document d_4 que dans le document d_1.

Les colonnes de la matrice «$mots \times documents$» représentent les vecteurs documents. Comme le mot «le» n'est pas important, nous pouvons représenter nos documents par des vecteurs à deux dimensions :
$\vec{d_1}\begin{pmatrix} 0,6 \\ 0,48 \end{pmatrix}$, $\vec{d_2}\begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\vec{d_3}\begin{pmatrix} 0 \\ 0,72 \end{pmatrix}$ et $\vec{d_4}\begin{pmatrix} 1,5 \\ 1,44 \end{pmatrix}$.

On pourra ensuite comparer ces vecteurs en calculant leur cosinus, par exemple.

2.2.2 Cooccurrence : Modèle HAL

Le modèle précédent, même s'il peut être transposé pour établir des vecteurs de mots et peut donc servir à coder les mots en fonction de leur contexte documentaire, n'a pas été conçu pour cela. L'un des premiers modèles dont l'objectif était de représenter des vecteurs de mots est introduit par LUND, BURGESS et ATCHLEY (1995), sous le nom de *Hyperspace Analogue to Language* (ci-après HAL). Son but est de construire à partir d'un corpus une représentation de l'espace des mots interprétable cognitivement.

L'hypothèse fondamentale sur laquelle repose ce modèle est que des mots sémantiquement similaires auront tendance à se trouver ensemble dans le même contexte. Ce n'est pas exactement l'hypothèse de Harris, mais elle aboutit au même résultat.

Dans ce modèle, les unités de contexte sont les mots qui entourent le mot considéré dans une fenêtre coulissante de longueur fixe sur les documents du corpus. On construit une matrice de co-occurrences entre mots en considérant deux mots comme co-occurrents lorsqu'ils se produisent dans une même fenêtre. La distance entre les mots dans la fenêtre est prise en compte.

Chaque composante c_{ij} de la matrice $mots \times mots$ est calculée selon la formule :

$$c_{ij} = \sum_{f=1}^{n} (1/(S - d(i,j))) \tag{2.3}$$

Où f est un indice qui parcourt les fenêtres du document, n est le nombre total de mots (et le nombre total de fenêtres), S est la taille de la fenêtre, et d la distance (nombre de mots + 1) entre les mots i et j s'ils figurent dans cet ordre dans la fenêtre.

La matrice n'est pas symétrique, car c_{ij} et c_{ji} ne sont pas identiques. Pour construire le vecteur du mot i, on concatène la ligne i avec la colonne i de la matrice.

Plusieurs expériences ont été faites pour valider le résultat en sélectionnant certains mots pour trouver ceux qui leur ressemblent le plus en fonction de la distance de leurs vecteurs. La sélection peut être aléatoire sur un ensemble de mots fréquents, ou déterminée par une liste préétablie.

Depuis quelques années, de nouvelles versions du modèle sont apparues, par exemple le système HiDEx (Shaoul et Westbury, 2006), qui améliore son interprétabilité cognitive en modifiant la définition du contexte (longueur variable) et en diminuant l'importance de la fréquence dans le calcul de la cooccurrence.

2.2.3 Information ponctuelle mutuelle (PMI)

Aussi bien dans HAL que dans le modèle de Salton, on utilise la fréquence d'un mot dans un contexte donné (mot ou document). Mais il peut être plus intéressant, quel que soit le modèle, de comparer cette fréquence à l'ensemble des fréquences observées du mot dans d'autres contextes. C'est un peu la fonction de IDF dans le modèle de Salton, mais qui se transpose mal à d'autres contextes que les documents.

Dans le cas des cooccurrences (mais aussi dans d'autres cas), on pourra utiliser l'information ponctuelle mutuelle, en anglais *Pointwise Mutual Information* (Church et Hanks, 1989), ci-après PMI, à la place de la fréquence, dans les différentes formules. La PMI de deux mots est une quantité mesurant la corrélation entre ces mots. Elle quantifie l'écart entre la probabilité d'observer les deux mots i et j ensemble et leur probabilité d'être observés séparément, selon l'équation suivante :

$$PMI(i,j) = \log \frac{P(i\&j)}{P(i) \cdot P(j)} \tag{2.4}$$

PMI peut avoir des valeurs négatives ou positives, $PMI(i,j) = 0$ signifie que les mots i et j sont indépendants statistiquement.

En pratique, en estimant la probabilité avec le maximum de vraisemblance (*maximum likelihood estimation*), l'équation 2.4 devient :

$$PMI(i,j) = \log \frac{\#(i,j) \times N}{\#i \times \#j} \tag{2.5}$$

Où N est le nombre total des mots du corpus, $\#(i,j)$ est le nombre de cooccurrences entre les mots i et j, $\#i$ et $\#j$ sont les nombres d'occurrences de chacun d'entre eux.

Par exemple dans un corpus de 50 millions de mots, les mots«hong» et «kong» apparaissent ensemble 2205 fois, «hong» apparaît 2438 fois et «kong» 2694 fois. $PMI(''hong'',''kong'') = 9,72$, et cela signifie une forte corrélation entre ces deux mots.

En revanche, les mots «chat» et «avion» apparaissent ensemble seulement 12 fois, «chat» apparaît 38.745 fois et«avion» 48256 fois. $PMI(''chat'',''avion'') = -1,13$, et cela signifie que les mots «chat» et «avion» ne sont pas corrélés.

Un score élevé indique une collocation et un score faible deux mots indépendants. Dans le cas où $\#(i,j) = 0, PMI(i,j) = \log 0 = -\infty$, donc les valeurs négatives de PMI ont tendance à être peu fiables, C'est pourquoi une version modifiée, PMI positif (PPMI), est généralement utilisée, les valeurs PMI négatives étant remplacées par zéro ($PPMI(i,j) = max(PMI(i,j), 0)$).

On trouvera une description très fouillée de PMI dans Turney et Pantel (2010).

2.2.4 Réduction de la dimensionnalité

Comme nous l'avons vu, les données de grandes dimensions présentent deux défis principaux. Le premier est de nature computationnelle : certains algorithmes s'échelonnent mal avec l'augmentation des dimensions. Le second est théorique et est généralement appelé le fléau de la dimensionnalité (*curse of dimensionality*) ; voir par exemple Friedman (1997) et Daum et Huang (2003).

Des techniques de réduction de dimensionnalité sont donc nées afin de traiter ce problème.

La toute première technique est le filtrage en amont de certains mots, qui permet dans le modèle de Salton d'éliminer un certain nombre de lignes, ou dans HAL d'éliminer lignes et colonnes. On peut ainsi éliminer les mots très fréquents, ou grammaticaux, ainsi que les mots très rares, de la matrice, ce qui en réduit la taille. On peut, à l'inverse, ne garder que les mots d'un vocabulaire ; mais si ceci peut être utile pour l'indexation documentaire, comme dans le modèle de Salton,

ou pour vérifier des hypothèses sur le lexique, comme dans HAL, cela ne nous fournit pas une solution pour vectoriser les mots d'un ensemble quelconque.

On a également vu que IDF pouvait servir, en aval, à réduire le nombre de lignes et donc la taille du vecteur de document. On peut également utiliser PMI avec le même objectif.

Mais ces techniques ne suffisent pas dans le cas d'un grand corpus. Les chercheurs ont donc développé des méthodes qui transforment les représentations vectorielles dans un espace $\mathbb{R}^n$ en des représentations vectorielles dans un espace de dimension inférieure $\mathbb{R}^d$ ($d \ll n$), tout en préservant autant que possible les propriétés de l'espace d'origine.

Nous allons présenter un échantillon de ces méthodes ; pour compléter cet état de l'art, se reporter à Lorenzo et Fabio Massimo (2017).

La plus ancienne de ces méthodes, bien connue et toujours utilisée, est l'analyse en composantes principales, en anglais *Principal Component Analysis*, ci-après PCA (Pearson, 1901). Elle consiste à projeter linéairement les données dans un espace de dimension inférieure tout en maximisant la variance des données dans l'espace réduit. PCA transforme des variables corrélées entre elles en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées « composantes principales ». La distance utilisée est normalement la distance euclidienne ; la variante «Hellinger PCA» utilise la distance d'Hellinger (Rémi et Ronan, 2013).

Nous allons présenter dans les sections suivantes deux classes de méthodes, la projection aléatoire (*random mapping*), parce qu'elle présente l'intérêt de ne dépendre d'aucune hypothèse implicite ou explicite sur la distribution des données, et d'être relativement simple, et l'analyse latente, qui, avec ses nombreuses variantes, est l'une des méthodes les plus utilisées.

2.2.5 Random mapping

Random mapping a été proposée initialement par Kaski (1998) dans le cadre du modèle WebSOM (voir 2.2.10), et, sous le terme de *Random projection*, par Bingham et Mannila (2001) et Achlioptas (2003).

Cette méthode est basée sur le lemme de Johnson-Lindenstrauss (Johnson et Lindenstrauss, 1984), selon lequel il est possible de projeter un espace à n dimensions dans un espace plus petit en préservant, moyennant un facteur réel ϵ dépendant de la dimension de la projection, les distances entre les vecteurs deux à deux.

La méthode consiste à multiplier la matrice de départ par une matrice initialisée aléatoirement. La génération aléatoire peut s'effectuer avec tout type de

distribution, Bingham et Mannila (2001) utilisent, pour des images, une distribution gaussienne avec une moyenne nulle et une variance unitaire $N(0,1)$. Ni Kaski (1998) ni Kohonen, Kaski et al. (1998) ne donnent d'indication sur la distribution utilisée : «The random matrix consists of random values and the Euclidean length of each column has been normalized to unity.» (Kaski, 1998).

Soit la matrice de départ $X \in \mathbb{R}^{n \times d}$ et la matrice aléatoire $W \in \mathbb{R}^{d \times p}$ ($p \ll d$), soit la matrice $Y \in \mathbb{R}^{n \times p}$ la projection de la matrice X dans le nouvel espace $\mathbb{R}^{n \times p}$: $Y = X \times W$. Selon le lemme de Johnson-Lindenstrauss, pour tout $\vec{x_i}, \vec{x_j} \in X$, nous avons des projections $\vec{y_i}, \vec{y_j} \in Y$ telles que :

$$(1-\epsilon)\|\vec{x_i} - \vec{x_j}\|^2 \leq \|\vec{y_i} - \vec{y_j}\|^2 \leq (1+\epsilon)\|\vec{x_i} - \vec{x_j}\|^2 \tag{2.6}$$

La nouvelle dimension p des vecteurs doit être choisie de telle sorte que :

$$p \geq \frac{8\log(d)}{\epsilon^2} \tag{2.7}$$

Achlioptas (2003) a montré que la distribution gaussienne pour générer les valeurs w_{ij} de la matrice aléatoire, peut être remplacée par une distribution d'une moyenne nulle beaucoup plus simple, telle que :

$$w_{ij} = \sqrt{3} \times \begin{cases} +1 & \textit{avec une probabilité de } \frac{1}{6} \\ 0 & \textit{avec une probabilité de } \frac{2}{3} \\ -1 & \textit{avec une probabilité de } \frac{1}{6} \end{cases} \tag{2.8}$$

Cette distribution permet une accélération significative du calcul due à la faible densité de la projection. En effet, la matrice aléatoire résultante contient beaucoup de zéro, c'est pour cette raison que la méthode est parfois appelée *sparse random projection* (Li, Hastie et Church, 2006).

2.2.6 Analyse sémantique latente (LSA)

L'analyse sémantique latente, en anglais *Latent Semantic Analysis*, ou *Latent Semantic indexing*, ci-après LSA, a été proposée par Deerwester et al. (1990). Il s'agit d'une famille de méthodes, parmi lesquelles on trouve aussi *Generalized LSA*, *Non-Negative Matrix Factorization*, *Probabilistic Latent Semantic Analysis* et *Latent Dirichlet Analysis* (Bernard et Lebboss, 2017).

Le principe général consiste à considérer que la distribution des mots dans leur contexte est la manifestation de la distribution des thèmes (en anglais *topics*) sous-jacents à ces mots, supposés être beaucoup moins nombreux que les mots.

On projette donc la matrice *mots* × *mots* sur une matrice *thèmes* × *thèmes* beaucoup plus réduite, à l'aide de différents outils mathématiques. Dans cette section nous présentons la méthode originale, puis deux des méthodes ultérieures dans les sections suivantes.

Dans la méthode originale (Deerwester et al., 1990), la réduction de dimension se base sur la décomposition en valeurs singulières (ci-après SVD), proposée par Eckart et Young dès 1936 (Bestgen, 2006), supposées ici représenter la sémantique cachée et sous-jacente (latente) des mots dans un corpus textuel. Certains utilisent la factorisation matricielle non négative (qui n'exige pas que les facteurs soient orthogonaux) plutôt que SVD (Lee et Seung, 2001). On trouvera une description détaillée des techniques de décomposition de matrices dans Golub et Van Loan (1996).

Comme l'indiquent Bernard et Lebboss (2017), «Vectors in the reduced space do not correspond to words but to weighted groups of words often refered to as 'topics'.».

En effet LSA n'est pas seulement une méthode pour réduire la taille de l'espace vectoriel comme *PCA* ou *Random mapping* citées ci-dessus ; elle a été créée pour répondre à deux problèmes du traitement automatique du langage : la synonymie et la polysémie.

1. La synonymie fait référence au cas où deux mots différents (ex. «voiture» et «automobile») ont la même signification. Le modèle de Salton ne parvient pas à capturer la relation entre des termes synonymes tels que «voiture» et «automobile», il va accorder à chaque terme une dimension distincte dans l'espace vectoriel.

2. La polysémie, quant à elle, désigne le cas où un terme a plusieurs significations ; le modèle de Salton va accorder une seule dimension dans l'espace vectoriel pour toutes les significations.

LSA parvient par un simple calcul matriciel à faire face au problème de la synonymie (beaucoup moins efficacement au problème de la polysémie), c'est pourquoi elle est devenue une méthode à part entière pour la représentation vectorielle de documents textuels.

Au départ, LSA a été appliquée sur des grandes matrices creuses *mots* × *documents* générées par le modèle de Salton. Elle a ensuite été appliquée à des matrices *mots* × *mots*, et à toutes sortes de matrices *mots* × *contextes*.

On commence par construire une matrice *mots* × *contextes*, en utilisant éventuellement *IDF* ou *PMI* pour calculer ses composantes. On procède ensuite à la

décomposition en valeurs singulières de cette matrice. SVD est une méthode algébrique permettant la factorisation de la matrice, afin d'obtenir une matrice ayant beaucoup moins de colonnes, mais plus dense.

La décomposition de la matrice de co-occurrences $X(n \times d)$, donne deux matrices orthonormales $U(n \times p)$ et $V(p \times d)$ et une matrice diagonale $\Sigma(p \times p)$.

On a alors :

$$X = U\Sigma V^t \tag{2.9}$$

Cette matrice diagonale Σ de taille $p \times p$ sera réduite à une autre matrice diagonale de taille $k \times k$ avec$(k \ll p)$, la réduction au rang k permet la valorisation des valeurs singulières les plus fortes, ce qu'on appelle la troncation.

Comme l'indiquent Bernard et Lebboss (2017), «The truncation keeps only the largest singular values, which implicitly assumes a gaussian distribution of the data». Mais les distributions de mots ne sont pas gaussiennes en général ; elles sont plutôt caractérisées par des lois de puissance (*power laws, long tail distributions*), comme la loi de Zipf. Les méthodes suivantes étendent LSA pour remédier à ce problème.

2.2.7 Analyse sémantique latente probabiliste

Hofmann (1999) a proposé le modèle d'analyse sémantique latente probabiliste, en anglais *Probabilistic Latent Semantic Analysis*, ci-après PLSA, fondé sur un modèle de Markov agrégé et une maximisation des espérances mathématiques (maximum de vraisemblance d'un modèle probabiliste).

«The algorithm reconstructs the underlying distribution of topics that generates the observed word vectors.» (Bernard et Lebboss, 2017).

PLSA emploie un mélange de décompositions issues de l'analyse des classes latentes (McCutcheon, 1987), pour réduire la matrice de co-occurrences, au lieu d'une décomposition en valeurs singulières comme dans LSA.

«PLSA s'appuie sur la fonction de vraisemblance de l'échantillonnage multinomial et vise à [...] minimiser l'entropie croisée ou divergence de Kullback-Leibler[1] entre la distribution empirique et le modèle»(Hofmann, 2001, p. 184, ma traduction).

2.2.8 Two Step CCA (TSCCA)

Dhillon et al. (2012) ont proposé TSCCA (Two Step CCA (*Canonical Correlation Analysis*)). Leur méthode est basé sur CCA (Hotelling, 1935), qui remplace

1. Mesure de dissimilitude entre deux distributions de probabilités.

PCA ou LSA pour la réduction de la matrice de cooccurrences de départ.

Le but de l'analyse canonique des corrélations (CCA) est de comparer deux groupes de variables quantitatives du même individu pour savoir s'ils décrivent un même phénomène. Elle est souvent utilisée dans des analyses médicales effectuées sur les mêmes échantillons par deux laboratoires différents.

L'idée principale de TSCCA est d'appliquer deux fois CCA sur les trois matrices L, W et R, en première fois entre la matrice de contextes gauche L et la matrice de contexte droit R et en seconde fois entre les projections résultants de $CCA(L, R)$ et la matrice W de cooccurrences $mot \times mot$.

CCA est l'analogue de l'analyse en composantes principales (PCA) pour deux matrices. PCA calcule les directions de covariance maximale entre les éléments d'une même matrice, alors que CCA calcule les directions de corrélation maximale entre deux matrices.

Selon les auteurs de TSCCA, CCA présente deux avantages par rapport à PCA ou LSA :

- Elle est invariante à l'échelle[2].
- Contrairement à LSA, CCA peut capturer des informations multi-vues. Dans les applications textuelles, les contextes gauche et droit des mots fournissent une scission naturelle en deux vues totalement ignorées par LSA qui ne prend pas en compte l'ordre des mots dans leur contexte.

Dhillon et al. (2012) illustrent l'efficacité empirique et la richesse des représentations apprises par TSCCA sur les tâches d'étiquetage morpho-syntaxique (POS tagging : part-of-speech tagging) et sur la classification des sentiments.

2.2.9 Latent Dirichlet Allocation (LDA)

L'allocation de Dirichlet latente proposée par Blei, Ng et Jordan (2003) est un modèle probabiliste génératif qui permet de découvrir des structures thématiques (*topics*) latentes dans une grande collection de documents textuels. Comme LSA et PLSA, LDA est fondée sur l'hypothèse que chaque document est un mélange de thèmes (*topics*), et que chaque mot w du document d est attribuable à l'un des thèmes. De la même façon que LSA, donc, on aboutira ici à des représentations de thèmes.

LDA est modélisée par un réseau bayesien hiérarchique à 3 couches, dans lequel on cherche à calculer les probabilités d'association entre chaque document et

2. l'invariance d'échelle est une caractéristique d'objets ou de lois qui ne changent pas si les échelles de longueur, d'énergie ou d'autres variables sont multipliées par le même facteur.

chaque thème. Les probabilités de thèmes associées à un document fournissent une représentation explicite de ce document. Pour estimer les paramètres du modèle, on utilise souvent l'algorithme Expectation-Maximization (Dempster, Laird et Rubin, 1977), décrit ici section 3.3.4, page 68.

Pour décrire le fonctionnement du modèle LDA, nous définissons les éléments suivants :

- Un vocabulaire V de n mots,
- Un document est une séquence de l_d mots, w_{ij} désigne le mot w_i dans le document j (i est l'indice de mot w dans le vocabulaire).
- Un corpus est une collection de m documents, $\{d_1, d_2, ..., d_m\}$,
- Les variables t_{ij} représentent le thème choisi pour le mot w_{ij},
- Les paramètres θ_d et β_t représentent respectivement la distribution en thèmes du document d et la distribution du thème t sur tout les documents.
- α et η définissent les distributions a priori sur θ et β respectivement.

On fixe un nombre k de thèmes et on cherche à apprendre les thèmes représentés dans chaque document et les mots associés à ces thèmes. Pour ce faire, l'algorithme se déroule comme suit :

1. **Initialisation :** au départ, on associe un thème à chaque mot du vocabulaire, selon une distribution de Dirichlet[3] sur l'ensemble de k thèmes. Ce premier modèle de thèmes « topic-model » est très peu vraisemblable car généré aléatoirement.
2. **Apprentissage :** dans chaque document d, on met à jour le thème t de chaque mot w, en maximisant la probabilité $p(thème\ t|document\ d) \times P(mot\ tw|thème\ tt)$ (la probabilité que le thème t génère le mot w dans le document d).

   Le processus génératif pour déterminer les probabilités de thèmes de chaque documents d est le suivant :

   - Pour chaque thème t, choisir les paramètres $\beta_t = (\beta_{t1}, \beta_{t2}, ..., \beta_{tn}) \sim Dirichlet(\eta)$, β_{tn} s'interprète comme la probabilité de l'occurrence du mot w_n dans un document du thème t.

3. La loi de Dirichlet, souvent notée $Dir(\alpha)$, est une loi de probabilité continue pour des variables aléatoires multinomiales (catégorielles). Elle est paramétrée par le vecteur α de nombres réels positifs.

- Pour chaque document d, choisir les paramètres de la distribution des k thèmes dans d selon $\theta_d = (\theta_{d1}, \theta_{d2}, ..., \theta_{dk}) \sim Dirichlet(\alpha)$, θ_{dt}: s'interprète comme la proportion des occurrences du document d qui sont associées au thème t.
- Pour chaque position i dans d, $i \in \{1, ..., l_d\}$:
  - Choisir un thème $t_{id} \sim Multinomial(\theta_d)$,
  - choisir un mot w_n conditionnellement au thème t_n selon une probabilité $P(w_n|t_n, \beta)$.

Loi de Dirichlet : la loi de Dirichlet permet de tirer une variable θ telle que $\forall i, \theta_i \geq 0$ et $\sum_{i=1}^{k} \theta_i = 1$.

Sa densité est de la forme :

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{2.10}$$

Avec $\alpha_i > 0$ et $\Gamma(x)$ la fonction Gamma.

2.2.10 WebSOM

Le modèle WebSOM (Kohonen, Kaski et al., 1998), comme le modèle de Salton, est un modèle d'indexation documentaire fondé sur le réseau neuromimétique Self Organizing Map (SOM) proposé par Kohonen (1982). Il est constitué de deux couches indépendantes, chacune composée d'un SOM, la première pour le clustering de mots et la deuxième pour le clustering de documents (voir figure 2.2). La partie de ce modèle qui nous intéresse ici est la phase initiale de la première couche, qui construit les représentations vectorielles de mots. Dans la suite de ce travail, quand nous ferons référence à WebSOM, c'est en fait à cette partie du modèle que nous faisons référence.

Après un prétraitement incluant la suppression des mots grammaticaux, les mots sont représentés par des vecteurs binaires orthogonaux produits par la méthode *one hot vector* (une seule valeur égale à 1, les autres égales à 0). Le nombre de dimensions est égal au nombre de mots.

On applique ensuite la méthode de réduction de dimensionnalité *Random mapping* (ici section 2.2.5) sur cet ensemble de vecteurs, qui les projette dans un espace beaucoup plus petit. Chaque mot m_i est associé à un vecteur x_i, de dimension d (en pratique $d = 90$), dont les composantes sont des nombres réels compris entre 0 et 1. Ces vecteurs ne sont plus orthogonaux, mais quasi orthogonaux, c'est-à-dire que leur produit est faible (en fonction du facteur ϵ du lemme cité).

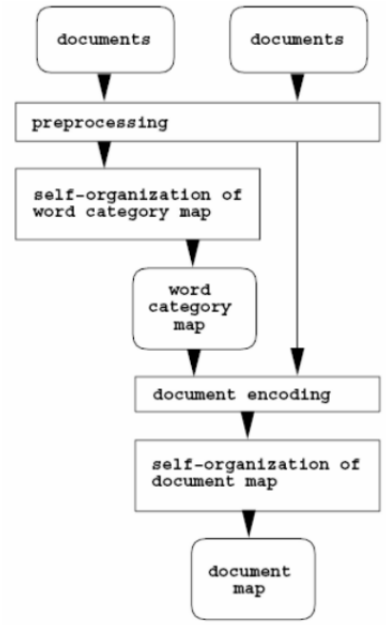

FIG. 2.2: Extrait de KOHONEN, KASKI et al. (1998) : architecture de WebSOM

À l'étape suivante, on va, pour chaque mot, récupérer l'ensemble de ses voisins à gauche, calculer la moyenne de leurs vecteurs, et de même pour la droite. Supposons qu'on veuille prendre en compte 2 mots à gauche et 2 mots à droite (fenêtre de diamètre 2), on va donc obtenir 4 vecteurs représentant la moyenne des vecteurs dans chaque position.

Pour obtenir le vecteur final pour chaque mot, on concatène, dans l'ordre, les vecteurs moyens. Au centre, on place le vecteur du mot, multiplié par un facteur η inférieur à 1 qui affaiblit son importance sur le vecteur d'ensemble. Ce paramètre permet d'ajuster l'importance attribuée au contexte. Si le diamètre de la fenêtre est D, le vecteur sera produit par la concaténation de $2D + 1$ vecteurs. Chacun des vecteurs étant de dimension d, la dimension du vecteur résultant sera $2Dd + d$. Dans l'application originale (KOHONEN, KASKI et al., 1998), le diamètre était de 1 et la dimension réduite de 90, le vecteur final avait donc 270 composantes ; le paramètre η était fixé à $0,2$.

Plus généralement, le vecteur résultant $Final(i)$ pour le mot i se présente sous la forme :

$$Final(i) = \begin{pmatrix} m(i,-D) \\ m(i,-D+1) \\ \vdots \\ m(i,-1) \\ \eta \cdot x_i \\ m(i,1) \\ \vdots \\ m(i,D) \end{pmatrix} \tag{2.11}$$

Où $D \in \mathbb{N}$ est le diamètre de la fenêtre, $0 < \eta < 1$ le paramètre de l'importance du contexte, et $m(i,n)$ la fonction définie comme suit :

$$m(i,n) = \frac{\sum_{f=1}^{N} (Vect(neighbour_f(i,n)))}{N} \tag{2.12}$$

Où i est l'indice du mot, n la distance, f un indice qui parcourt les fenêtres, N le nombre total de fenêtres (et le nombre total de mots). $Vect(i)$ ramène le vecteur réduit du mot i et $neighbour_f(i,n)$ ramène le mot à la distance n par rapport au mot i dans la fenêtre f.

Illustrons avec un exemple simple, en partant d'un corpus contenant la phrase «L'ignorant aime à nier, le savant aime à croire», qui devient «ignorant aime nier savant aime croire» après suppression des mots grammaticaux et des ponctuations.

Ce corpus contient M mots. On associe à chaque mot un vecteur binaire de dimension M selon la méthode *one hot vector*. Après application du random mapping, les mots ont un vecteur réel réduit à d dimensions ($d \ll M$).

Fixons le diamètre de la fenêtre à 1 et intéressons-nous à «aime», qui se retrouve dans les trigrammes <ignorant **aime** nier>, <savant **aime** croire>. On calcule les moyennes sur le contexte :

$m(aime,-1) = (Vect(ignorant) + Vect(savant))/2$

$m(aime,1) = (Vect(nier) + Vect(croire))/2$

Puis on concatène le tout, ce qui donne le vecteur final du mot « aime » :

$$Final(aime) = \begin{cases} \left.\begin{matrix}\cdot\\\cdot\end{matrix}\right\} m(aime, -1) \\ \left.\begin{matrix}\cdot\\\cdot\end{matrix}\right\} \eta \cdot Vect(aime) \\ \left.\begin{matrix}\cdot\\\cdot\end{matrix}\right\} m(aime, 1) \end{cases}$$

2.2.11 Méthode par marques grammaticales

Cette méthode a été proposée par Bernard (1997). En contraste avec la plupart des autres méthodes, elle se fonde sur la distribution des mots grammaticaux, utilisés comme contextes pour les mots. L'un des intérêts est que la liste des mots grammaticaux, ou *stopwords*, est facilement disponible pour beaucoup de langues, un autre est que cette liste est courte (311 éléments pour le français), ce qui élimine le problème de la dimensionalité.

Les mots grammaticaux sont groupés en catégories, avec des tests sur plusieurs groupements. Dans tous les cas, une unique catégorie regroupe l'ensemble des mots non grammaticaux comme contexte. Seulement le contexte gauche des mots a été pris en compte.

Pour tester cette approche, un modèle Self Organizing Maps a été utilisé, avec les mots les plus fréquents pour ensemble d'apprentissage. Les clusters obtenus ont montré que ces vecteurs contenaient aussi bien de l'information grammaticale que sémantique, puisque les mots du corpus français étaient répartis selon leur catégorie grammaticale (noms, nombres, adjectifs, verbes à l'infinitif, verbes conjugués), ou selon leur catégorie sémantique dans les tests effectués sur les noms (principalement distinguant les noms abstraits des autres).

L'originalité de cette méthode repose donc sur l'idée que les mots grammaticaux peuvent fournir au codage des vecteurs des informations sémantiques, même si le corpus utilisé à l'époque était beaucoup trop petit (600.000 mots) pour éviter les influences dues à un texte particulier (par exemple, un roman de Jules Verne influençait la classification du mot «pied» comme terme de mesure).

2.2.12 GraPaVec

Lebboss et al. (2017) ont proposé une nouvelle méthode de vectorisation de mots, GraPaVec (*Grammatical Pattern Vectors*). Ils se sont inspirés de l'idée générale de la méthode citée précédemment (méthode par marqueurs grammaticaux), princi-

palement pour développer une méthode aussi indépendante que possible de langues spécifiques, appliquée à l'arabe standard moderne. Il y a deux innovations importantes : la liste des mots grammaticaux est remplacée par la liste des mots les plus fréquents (ci-après appelés *marques*), et le contexte fixe est remplacé par la recherche d'un pattern "grammatical" au voisinage du mot (ici grammatical signifie simplement composé de mots très fréquents).

Un pattern est un schéma textuel composé de marques (sorte de skipgram grammatical) qui se répète plusieurs fois dans le texte, par exemple le pattern « le...de...» est observé au voisinage de «livre» dans : « le livre de Jacques », « le livre de géographie », « le grand livre de Claude ». L'ordre compte.

GraPaVec se déroule en cinq étapes comme le montre la figure 2.3 :

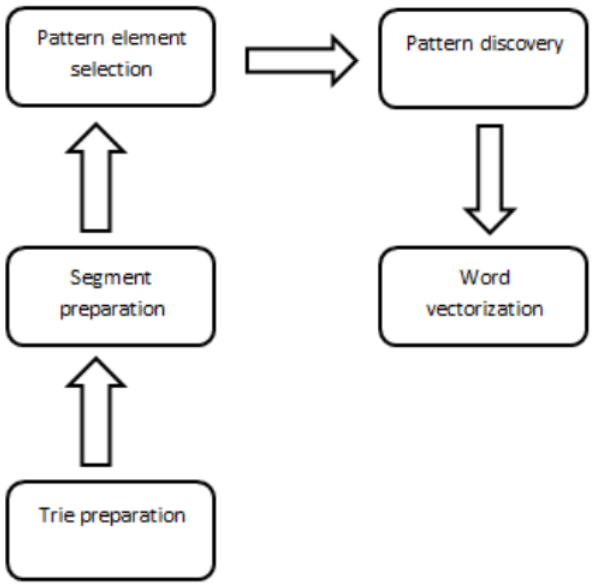

Fig. 2.3: Extrait de Lebboss et al. (2017) : Architecture de GraPaVec

On parcourt les textes du corpus pour intégrer les mots dans un arbre préfixe *Trie*[4]. En effet, «[u]n arbre Trie est plus efficace que les arbres binaire de recherche ou les graphes de mots acycliques dirigés qui perdent certaines informations qu'on veut conserver.» (Lebboss et al., 2017). Un seuil établi par l'utilisateur sélectionne la liste des marques.

On parcourt une deuxième fois les textes du corpus pour construire les patterns (également dans un arbre Trie), à l'aide de la liste de marques. Les patterns sont

4. En informatique, un Trie est une structure qui représente un très grand nombre de mots dans un format à la fois économique et rapide à explorer

construits selon les règles suivantes (m représente une marque, x un mot ordinaire, p une ponctuation) :

- Un pattern ne contient pas p,
- Un pattern est une séquence de m et $*$,
- Un pattern contient au moins un $*$,
- $*$ est une chaîne de x avec n comme longueur maximale,
- $*$ contient au moins un x.

Prenons la séquence suivante, représentant un extrait du corpus :

$xmmxxmxxpmpmxmmxxxxmpmpxxxx$

L'objectif est de générer tous les patterns possibles compatibles avec cette séquence. Ces patterns seront représentés par des séquences de m et $*$, comme dans $<*mm*m>$. Un paramètre P fixe le nombre maximum de x pouvant se trouver dans le pattern (autrement dit, il fixe la taille maximale du trou dans « le...de...»). Avec $P = 3$, on obtient les patterns suivants :

- $*mm*m*$ (suivi de p),
- $m*mm*$ (suivi de plus de 3 x),
- $*m$ (suivi de p),
- $mm*m$ (fin de fichier considérée comme p)

On élimine tous les patterns peu fréquents et on stocke dans une base de données toutes les informations utiles pour construire la matrice de cooccurrences $mots \times patterns$.

Chaque composante c_{ij} dans cette matrice représente le nombre de fois où apparaît le mot i au voisinage du pattern j. Ce processus produit une matrice creuse (sparse matrix), chaque ligne est une représentation vectorielle du mot correspondent. La représentation vectorielle de mots dépend donc de trois paramètres : le seuil des marques, la taille maximale du trou et le seuil de pattern.

2.3 Modèles prédictifs

Au début du XXIe siècle, une nouvelle idée a émergé : les représentations vectorielles de mots, au lieu d'être calculées en préalable d'un traitement (indexation documentaire, clustering), comme avec les modèles statistiques, peuvent être apprises pendant le traitement. C'est pour cette raison que le terme *word embedding* a été inventé, qui veut dire en français « enchâssement de mots », la représentation des mots étant "enchâssée" dans l'apprentissage.

Pour le dire autrement, au lieu de calculer $\vec{X}$ pour ensuite calculer $f(\vec{X}) = \vec{Y}$, on va simultanément apprendre $\vec{X}$ et $f(\vec{X})$; ou encore, on va écrire un programme où l'entrée est $\vec{X}, \vec{Y}$ et la sortie également. Si on concatène $\vec{X}$ et $\vec{Y}$ en $\vec{X'}$, le problème consiste donc à associer $\vec{X'}$ à lui-même. C'est pourquoi on appelle aussi ces modèles des *auto-encodeurs*. Cette idée avait déjà été proposée dans certains modèles neuronaux, dès 1986 (Hecht-Nielsen, 1987 ; Kosko, 1988).

Ces nouvelles méthodes produisent des vecteurs denses de taille raisonnable, contrairement à la plupart des modèles statistiques qui produisent des vecteurs creux de très grande dimension.

2.3.1 Modèle de Bengio

Le premier modèle a été proposé par Bengio et al. (2003), fondé sur un réseau de neurones *feedforward* de trois couches (voir la figure 2.4), avec un modèle probabiliste de langue.

L'idée principale de ce modèle est d'utiliser un réseau de neurones pour apprendre à estimer la probabilité d'observer le mot courant en s'appuyant sur les mots précédents.

La fonction d'activation de la couche de sortie est basée sur le principe de vraisemblance maximale (*maximum likelihood*). Il s'agit de maximiser la probabilité conditionnelle d'appartenance d'un mot donné w à un contexte donné c en utilisant la fonction softmax qui garantit des probabilités positives et leur somme égale à 1:

$$P(w|c) = softmax(y_w) = \frac{e^{y_w}}{\sum_{v \in Vocab} e^{y_v}} \tag{2.13}$$

Les y_v sont les probabilités logarithmiques non normalisées de chaque mot v de sortie, elles dépendent de plusieurs paramètres des deux premières couches (la couche d'entrée et la couche cachée).

La mise à jour des pondérations se fait avec la rétropropagation du gradient. À la fin de l'apprentissage les lignes de la matrice de pondérations entre la couche

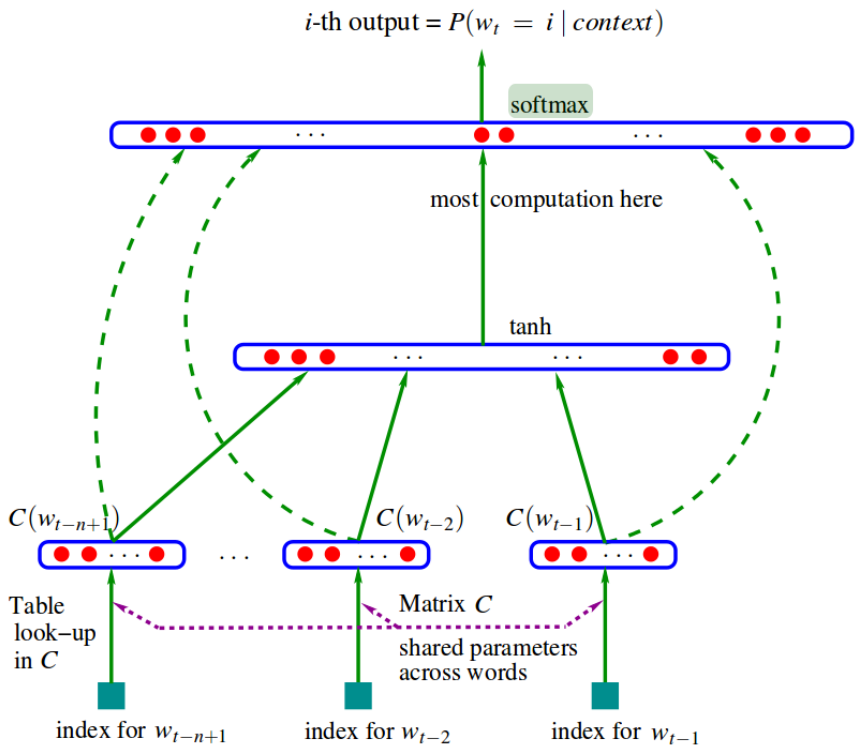

FIG. 2.4: Extrait de BENGIO et al. (2003) : le premier *word embedder*

d'entrée et la couche cachée représentent les vecteurs de mots.

Ce modèle effectue une prédiction presque parfaite du mot suivant étant donné un contexte, en tout cas en anglais, ce qui est un indice de qualité. Cependant, il demande énormément de temps de calcul, car nous devons calculer et normaliser chaque probabilité en utilisant le score de tous les autres mots du contexte actuel, et cela à chaque étape d'apprentissage. Son utilisation est donc restée très expérimentale.

2.3.2 Modèle Collobert et Weston (C&W)

Après les premiers pas de Bengio et al. (2003) dans les modèles de langue neuronaux, la recherche sur le *word embedding* stagnait, à cause de la demande excessive de temps de calcul qui ne permettait pas la formation des représentations vectorielles de mots d'un vocabulaire étendu. Collobert et Weston (2008) ont proposé un nouveau modèle neuronal capable de produire des vecteurs à partir d'un ensemble de données suffisamment volumineux. Les auteurs indiquent qu'en plus, ces vecteurs comportent une signification syntaxique et sémantique qui améliore les performances des tâches en NLP (Collobert, Weston et al., 2011).

Leur solution pour éviter le temps de calcul excessif consiste à remplacer la fonction softmax par une autre fonction d'activation moins coûteuse : au lieu de maximiser la probabilité du mot suivant étant donné les mots précédents, Collobert et Weston (2008) entraînent le réseau à discriminer entre des séquences de mots existant dans le corpus et des séquences de mots bruitées (le mot du milieu est remplacé par un autre mot du vocabulaire). La nouvelle fonction d'activation a pour objectif de maximiser les scores $f_\theta(x)$ pour les vraies séquences de mots x et minimiser les scores $f_\theta(\hat{x})$ pour les séquences bruitées $\hat{x}$. Cela revient à minimiser la fonction de coût donnée par l'équation 2.14) sur l'ensemble des fenêtres X possibles de n mot dans le corpus :

$$J_\theta = \sum_{x \in X} \sum_{\hat{x} \in V} \max\left(0, 1 - f_\theta(x) + f_\theta(\hat{x})\right) \tag{2.14}$$

Pour chaque fenêtre x, ils remplacent son mot central par un mot tiré aléatoirement du vocabulaire V, afin de produire une séquence de n mots bruitée $\hat{x}$. La figure 2.5 illustre l'architecture du modèle.

Le réseau est composé d'une couche d'entrée, trois couches cachées : une couche de projection (Lookup Table) suivie d'une couche linéaire et une non linéaire (Tanh), et une couche linéaire de sortie qui calcule un score pour une séquence de mots donnée.

Bien que leur nouvelle fonction d'activation élimine la complexité de softmax,

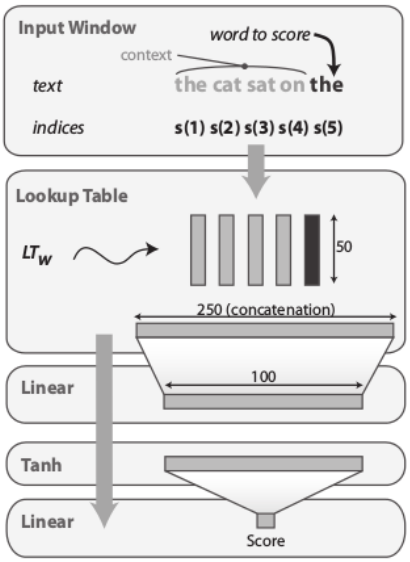

FIG. 2.5: Extrait de COLLOBERT et WESTON (2008) : architecture de C&W

ils conservent tout de même la couche cachée intermédiaire entièrement connectée, qui constitue une autre source de calcul coûteux. En conséquence, leur modèle s'entraîne pendant sept semaines avec un vocabulaire de 130 000 mots.

Ce modèle est ensuite revisité par TURIAN, RATINOV et BENGIO (2010). Au lieu de bruiter le mot au centre de la fenêtre, ils bruitent le dernier mot, et proposent d'utiliser d'autres paramètres d'apprentissage pour entraîner le réseau de neurones.

2.3.3 Word2Vec

MIKOLOV, CHEN et al. (2013) et MIKOLOV, SUTSKEVER et al. (2013) proposent de nouveaux modèles neuronaux probabilistes de langue plus efficaces, devenus très populaires sous le nom de *Word2Vec*. Word2Vec est beaucoup moins coûteux en temps de calcul que le modèle de Bengio et que C&W. Nous décrivons ici les principales innovations pour accélérer le calcul et améliorer la qualité des vecteurs. Pour plus de détail, voir RONG (2014).

La première innovation concerne la couche cachée, responsable d'une grande partie de la complexité du calcul dans le modèle de Bengio. Ils ont remplacé l'activation non-linéaire par une activation linéaire, et ils la considèrent donc comme une simple couche de projection (*projection layer*). La rétro-propagation du gradient est toujours utilisée pour la correction des pondérations d'apprentissage.

La deuxième innovation concerne l'activation de la couche de sortie, pour remplacer Softmax. Mikolov disait à propos de la formule 2.13:

> This formulation is impractical because the cost of computing $\bigtriangledown \log p(w|c)$ is proportional to the number of words in the vocabulary, which is often large ($10^5 - 10^7$ terms).
>
> (Mikolov, Sutskever et al., 2013)

Ils ont d'abord remplacé Softmax par Softmax Hiérarchique (*Hierarchical Softmax*) puis ils ont proposé (au choix) un classifieur log-linéaire et l'échantillonnage négatif (Mikolov, Sutskever et al., 2013).

La troisième innovation concerne la fonction utilisée pour entraîner le réseau (ci-dessus désignée comme $f(\vec{X} = \vec{Y})$; Bengio utilisait le contexte antérieur pour prédire les mots, Word2Vec présente deux fonctions à approcher, qui seront implémentées dans deux modèles légèrement différents, même s'ils ont en commun les principales innovations décrites (Mikolov, Chen et al., 2013) : CBOW (*Continuous Bag-Of-Words*) et SkipGram.

CBOW prédit un mot à partir de son contexte, et SkipGram prédit le contexte à partir d'un mot ; nous donnons les détails dans les sous-sections suivantes. Mikolov, Chen et al. (2013) recommandent d'utiliser CBOW pour les petits corpus et pour capturer les relations syntaxiques, et d'utiliser SkipGram pour les grands corpus et pour capturer les relations sémantiques. CBOW consomme moins de temps en entraînement que SkipGram.

Enfin, ils ont introduit le sous-échantillonnage des mots fréquents au cours de l'apprentissage.

Détaillons ces points.

2.3.3.1 Activation de la couche de sortie

Softmax hiérarchique est une fonction approximant la fonction Softmax, introduite par Morin et Bengio (2005). L'avantage principal de Softmax hiérarchique est qu'au lieu d'évaluer N nœuds de sortie (N correspondant au nombre de mots) pour obtenir la distribution de probabilité, il n'est nécessaire d'évaluer que $\log_2(N)$ nœuds. Pour ce faire, elle utilise un arbre binaire de recherche dans la couche de sortie. Les mots sont représentés par ses feuilles et chaque nœud dans l'arbre représente les probabilités relatives de ses nœuds enfants.

Soit $n(w, j)$ le j-ième nœud sur le chemin entre la racine et le mot w, et soit $L(w)$ la longueur de ce chemin (par exemple, si $n(w, 1) = racine$, $n(w, L(w)) = w$). Pour tout noeud interne n, on choisi aléatoirement un enfant $ch(n)$ parmi ses

enfants. Softmax hiérarchique définit la probabilité conditionnelle du mot w par rapport à un autre mot w' comme suit :

$$P(w|w') = \prod_{j=1}^{L(w)-1} \sigma(f(n(w,j+1) = ch(n(w,j))) \times V_{n(w,j)} \cdot V_{w'} \tag{2.15}$$

$$\text{Avec : } f(x) = \begin{cases} 1 & \textit{Si } x \textit{ est vrai} \\ -1 & \textit{Si } x \textit{ est faux} \end{cases}$$

$V_{n(w,j)} \cdot V_{w'}$ est le produit scalaire entre les vecteurs mots, $\sigma(x) = 1/(1 + \exp(-x))$, et bien entendu $\sum_{i=1}^{T} P(w_i|w_I) = 1$.

L'**échantillonnage négatif** (*Negative Sampling*) est une version simplifiée de l'estimation du contraste de bruit (*Noise Contrastive Estimation*) présenté par Gutmann et Hyvärinen (2012), qui estime qu'un bon modèle devrait pouvoir différencier les données du bruit au moyen d'une régression logistique.

L'idée derrière l'échantillonnage négatif est de représenter chaque mot w_i et chaque mot de son contexte, $w_j \in C_{w_i}$ par des vecteurs de dimension de manière à ce que le produit scalaire de $\vec{w_i}$ et $\vec{w_j}$ soit :

- maximum pour les couples (w_i, w_j) qui apparaissent ensemble dans le corpus (w_i et w_j sont liés sémantiquement),
- minimum pour les exemples négatifs (w_i, w_{Neg}), qui ne sont pas observés dans le corpus (w_i et w_j n'ont pas de lien sémantique).

Les exemples négatifs sont générés aléatoirement à partir du corpus. L'échantillonnage négatif a comme paramètre k, le nombre d'échantillons ; k est choisi entre 5 et 20 pour un petit corpus et entre 2 et 5 pour un grand corpus.

2.3.3.2 Sous-échantillonnage des mots fréquents

Le sous-échantillonnage des mots fréquents au cours de l'apprentissage entraîne une accélération significative du calcul et améliore la qualité des représentations vectorielles des mots les moins fréquents. Dans les très grands corpus, les mots les plus fréquents peuvent facilement apparaître des centaines de millions de fois (par exemple les mots vides). Il est convenu de dire que de tels mots fournissent moins d'informations que les mots rares.

Cette technique permet au modèle de renforcer l'importance de cooccurrences comme («france», «paris») et de diminuer celles de cooccurrences comme («france»,

«la»). Attention, ceci concerne uniquement l'apprentissage, au final, tout les mots du vocabulaire,mêmes les plus fréquents, auront des représentations vectorielles.

Ils ont utilisé une approche simple de sous-échantillonnage : chaque mot w_i de l'ensemble d'apprentissage est éliminé avec une probabilité calculée par la formule :

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \tag{2.16}$$

Où $f(w_i)$ est la fréquence du mot w_i et t est un seuil choisi, typiquement autour de 10^{-5}.

2.3.3.3 Fonctions à estimer

CBOW a pour objectif de prédire un mot donné du vocabulaire, à partir de ses contextes (figure 2.6).

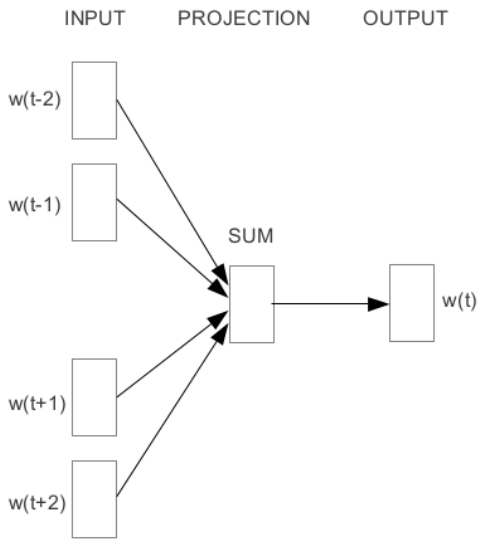

Fig. 2.6: Extrait de Mikolov, Chen et al. (2013) : architecture de CBOW

Dans ce modèle la couche de sortie est partagée pour tous les mots (tous les mots sont projetés dans la même position pour obtenir la moyenne de leurs vecteurs. L'ordre des mots dans le contexte n'est pas pris en compte (*Bag-Of-Words*).

La principale idée de CBOW est de calculer la somme des vecteurs mots des contextes, puis de donner le vecteur résultant à un classifieur log-linéaire pour prédire le mot cible. Le modèle compare sa prédiction avec le résultat souhaité

et ajuste les paramètres du réseau (les représentations vectorielles de mots) par la méthode de rétro-propagation du gradient.

Ce modèle cherche à maximiser la valeur de la formule suivante :

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t \mid w_{t-n}, ..., w_{t-1}, w_{t+1}, ..., w_{t+1}) \tag{2.17}$$

Où T est le nombre de mots dans le vocabulaire et n la taille de la fenêtre de contexte autour du mot cible w_t.

SkipGram vise à prédire les mots qui se produisent avant et après le mot cible (figure 2.7). Il introduit pour ce faire la notion de skipgrams, qui sont des ngrammes "à trous".

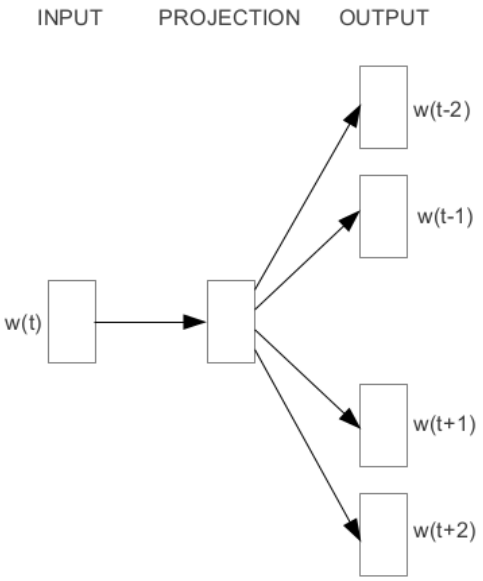

FIG. 2.7: Extrait de (MIKOLOV, CHEN et al., 2013) : architecture de Skip-gram

Étant donnée une séquence de mots d'apprentissage $w_1, w_2, w_3, \ldots, w_T$, l'objectif du modèle SkipGram est de maximiser la moyenne de log-probabilité :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{t+j} \mid w_j) \tag{2.18}$$

Où T est le nombre de mots du vocabulaire et c la taille de la fenêtre de contexte.

2.3.4 FastText

Bojanowsk et al. (2016) ont proposé une nouvelle méthode de représentation vectorielles de mots basée sur le modèle SkipGram (Mikolov, Chen et al., 2013). Cette méthode a deux particularités : elle donne aussi des représentations vectorielles à des mots qui n'apparaissent pas dans le corpus d'apprentissage, d'une part, et de l'autre elle intègre de l'information sur la morphologie du mot, en attribuant donc un vecteur distinct à plusieurs mots qui se trouvent dans le même contexte.

> En effet, les autres modèles, comme Word2Vec, ignorent en particulier la structure interne des mots, ce qui constitue une limite importante pour les langues riches sur le plan morphologique, telles que le turc ou le finnois. Par exemple, en français ou en espagnol, la plupart des verbes ont plus de quarante formes fléchies différentes.
>
> (Bojanowsk et al., 2016, ma traduction).

Dans FastText, les mots du corpus sont découpés en ngrammes de caractères et ce sont ces ngrammes qui sont traités par l'algorithme SkipGram pour générer des vecteurs. Les mots sont ensuite représentés comme la somme des vecteurs des ngrammes qu'ils contiennent. Après la génération du modèle à partir d'un grand corpus, nous pouvons extraire des vecteurs mots pour n'importe quel mot, même ceux qui ne sont pas inclus dans le corpus d'apprentissage, et même des mots rares (normalement supprimés par le seuillage). En effet la probabilité qu'un nouveau mot contienne des ngrammes qui ne figurent pas dans le corpus d'apprentissage est extrêmement faible.

L'algorithme étant récent, nous n'avons pas trouvé d'évaluation des performances comparées sur des applications concrètes en dehors des tests faits par les auteurs, selon lesquels FastText réussit mieux que Word2Vec (les deux modèles), sur Wikipedia en huit langues (dont le français, l'anglais et le russe), avec différents datasets comme mesure de comparaison. Nos propres tests (voir dernière partie) ne confirment pas ces résultats. Le temps d'apprentissage reste proche de celui de SkipGram standard.

2.3.5 GloVe

GloVe (Global Vectors for word representation) est un modèle proposé par l'équipe NLP de l'université de Stanford (Pennington, Socher et Manning, 2014). Cette méthode combine les avantages de la factorisation de la matrice de co-occurrences et les techniques de l'apprentissage automatique des paramètres. Ce

modèle est considéré comme modèle statistique à base de comptage par Arora et al. (2016), bien que la plupart des chercheurs du domaine le considèrent comme un modèle prédictif (Levy, Goldberg et Dagan, 2015 ; Li et Jurafsky, 2015).

Ce modèle repose sur la construction d'une matrice X de cooccurrences $mots \times mots$. Chaque valeur X_{ij} de cette matrice représente le nombre de fois où le mot w_j apparaît dans le contexte du mot w_i (le contexte est une fenêtre de longueur fixe d'éléments lexicaux centrés sur le mot w_i).

GloVe est un modèle d'apprentissage non supervisé qui prend en compte toute l'information portée par le corpus et non pas la seule information portée par une fenêtre de mots, d'où le nom GloVe, pour Global Vectors.

Chaque mot w du vecabulaire est représenté par deux vecteurs : vecteur ligne $\vec{w}$ et vecteur colonne $\vec{\tilde{w}}$, le vecteur final du mot w est la somme des deux vecteurs $(\vec{w} + \vec{\tilde{w}})$. On cherche à représenter chaque mot w_i et chaque mot w_j apparaissant dans le même contexte par des vecteurs tels que :

$$\vec{w_i} \cdot \vec{\tilde{w}_j} + b_i + \tilde{b_j} = log(X_{ij}) \tag{2.19}$$

Où b_i et $\tilde{b_j}$ sont des biais scalaires associés aux mots w_i et w_j respectivement.

En effet, l'objectif de GloVe est d'apprendre les vecteurs mots en minimisant la fonction de coût, qui calcule la somme des erreurs au carré :

$$E = \sum_{i,j=1}^{|V|} f(X_{ij})(\vec{w_i} \cdot \vec{\tilde{w}_j} + b_i + \tilde{b_j} - log(X_{ij}))^2 \tag{2.20}$$

Où $|V|$ est la taille du vocabulaire V, $f(X_{ij})$ est une fonction qui pondère le coût en fonction de la valeur X_{ij} définie comme suit :

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{X_{max}})^\alpha & Si\ X_{ij} < X_{max} \\ 1 & Sinon \end{cases} \tag{2.21}$$

Dans la pratique : $X_{max} = 100$ et $\alpha = 3/4$. Lorsque la valeur de X_{ij} est supérieure à X_{max}, la fonction de coût ne prend pas en compte cette valeur.

En temps de calcul GloVe est comparable à CBOW. En qualité des représentations, certains tests favorisent GloVe et d'autres Word2Vec. En fait, la controverse sur la meilleure méthode est très active, cf Levy, Goldberg et Dagan (2015), Schnabel et al. (2015) et Baroni, Bernard et Kruszewski (2014).

2.4 Conclusion

Nous nous sommes efforcé de décrire dans ce chapitre, sans prétendre à l'exhaustivité, un panel assez représentatif des divers types de méthodes de représentation vectorielle des mots, en n'oubliant ni les méthodes les plus populaires, comme LSA ou Word2Vec, ni les méthodes présentant une originalité particulière.

Deux éléments ressortent de notre étude.

Tout d'abord, on constate que d'une manière globale, à côté de la division statistiques / prédictives, il y a ici deux sortes de méthodes ; des méthodes qui associent directement aux mots un vecteur et des méthodes indirectes, qui associent aux mots des thèmes qui sont associés à des vecteurs (la famille des algorithmes *latents*).

Avec la famille *latente*, plusieurs mots auront donc la même représentation ; avec les méthodes directes, des mots qui ont la même distribution auront la même représentation, même si en réalité, avec un grand corpus, la représentation n'aura pratiquement aucune chance d'être identique. Avec la méthode WebSOM, l'identité est impossible, puisque chaque vecteur contient une partie exclusivement liée au mot. FastText est un cas un peu à part, car il associe aux mots des ngrammes associés à des vecteurs, mais il aboutit à une représentation différente pour chaque mot, donc on peut le regrouper avec les méthodes directes.

Pour notre propos, nous prendrons en compte les méthodes directes, car les méthodes indirectes posent des problèmes pour l'évaluation ; en effet, comme on le verra, les évaluations extérieures reposent sur les mots et non sur les thèmes.

Par conséquent nous utiliserons dans la deuxième partie (système et expérimentations) quatre méthodes directes : la méthode WebSOM comme baseline, les deux méthodes neuronales de Word2Vec (SkipGram et CBOW), deux méthodes non neuronales récentes (GloVe et GraPaVec), et FastText.

D'autre part, la publication de Word2Vec a eu une grande publicité ; peut-être le fait que ce soit une équipe du géant Google qui ait fait cette avancée y est-il pour quelque chose. En tout état de cause, elle a été un facteur important dans l'élan nouveau dans la recherche en NLP sur la représentation des mots, à l'origine de nombreuses propositions. Certaines sont élaborées à partir de Word2Vec, comme FastText, Levy et Goldberg (2014), d'autres proposent d'autres modèles prédictifs comme Lebret et Collobert (2014) (adaptant un modèle plus ancien de Collobert et Weston (2008)) ou GloVe (devenu pratiquement aussi populaire que Word2Vec), d'autres enfin renouvellent des modèles statistiques, comme GraPaVec, Subramanian et Vora (2016).

Dans l'opinion générale (par exemple Bernard et Lebboss (2017)), les modèles prédictifs ont gagné la course contre les méthodes statistiques, surtout dans

les grands corpus. La difficulté des modèles statistiques avec les grands corpus réside dans la gestion de la mémoire et du temps de calcul, en raison de la grande matrice de cooccurrences.

Mais Levy, Goldberg et Dagan (2015) indiquent qu'il sufit de jouer avec les paramètres des méthodes statistiques pour les rendre bien meilleures que les méthodes prédictives. D'autre part, les méthodes statistiques renouvelées utilisent des techniques pour diminuer le temps de calcul. Les auteurs de GraPaVec utilisent une base de données pour stocker les informations collectées du corpus, qui permet d'ajouter de nouveaux fichiers sans refaire tout le processus de calcul. Ceux de GloVe utilisent des fichiers intermédiaires sur disque pour calculer la matrice de co-occurrences. Les techniques de réduction de dimensionalité restent aussi d'actualité, et certains modèles prédictifs en incorporent.

Relevons par ailleurs que les méthodes statistiques restent les meilleures pour la représentation vectorielle de documents ; les modèles prédictifs représentent un document généralement par la moyenne des vecteurs mots qui le composent, ce qui donne des vecteurs de qualité médiocre (Joulin et al., 2016).

La recherche très active en ce moment rend cruciale la réponse à la question au coeur de notre thèse : quelle méthode donne la meilleure représentation vectorielle pour les mots ? Cette question en soulève d'autres :

- quelles sont les meilleures techniques d'évaluation des représentations ?
- peut-on mesurer la qualité de ces représentations sans prendre en compte leur contexte d'usage ?

Pour nous guider dans ces questions, les chapitres suivants de cette partie abordent l'état de l'art des méthodes d'évaluation et des méthodes de catégorisation sémantique des mots, que nous avons adopté comme contexte d'usage (voir chapitre 1, page 26).

Chapitre 3

Catégorisation sémantique de mots

Sommaire

3.1 Introduction

Dans ce chapitre nous allons passer en revue les modèles effectuant la catégorisation sémantique de mots. Notre objectif étant de préciser les contextes d'usage des représentations de mots présentées au chapitre précédent, nous ne cherchons pas l'exhaustivité ni même une couverture large.

Dans la première section, nous présentons les principes généraux communs à tous les modèles ; puis nous montrons qu'il existe deux sortes de modèles qui se différencient par la manière d'évaluer la qualité du clustering produit, Bag of Clusters et Self Organizing Map.

Dans la deuxième section, nous présentons la manière d'évaluer les algorithmes de la famille Bag of Clusters, et quelques modèles ; dans la troisième section, nous présentons la famille Self Organizing Map et ses techniques d'évaluation, avant de conclure.

3.2 Principes de la catégorisation

La catégorisation sémantique de mots est un cas particulier de clustering (regroupement non supervisé). L'objectif d'une méthode de clustering est de regrouper les objets qui se ressemblent. Dans notre cas d'étude les objets sont des mots et la ressemblance est une relation sémantique entre les mots.

Dans tous les cas examinés ici (il y a quelques exceptions), il faut partir d'une représentation vectorielle des mots d'un corpus. Le processus de clustering utilise une mesure de distance ou de similitude entre les vecteurs pour construire les clusters. Le choix de la mesure et de la représentation vectorielle influera sur le type de clusters construit.

Les types de représentation vectorielle fondés sur l'hypothèse de Harris (ou la variante proposée par Lundt à la base de HAL) sont compatibles avec une mesure sémantique, selon ces hypothèses mêmes. En effet, l'hypothèse de Harris dit que les mots sémantiquement proches auront la même distribution et celle de Lundt dit que les mots sémantiquement proches seront voisins (et donc dans les mêmes contextes).

Par conséquent, selon ces hypothèses, la mesure de l'écart entre les distributions de deux mots sera un indicateur de la similitude sémantique de ces deux mots. Nous pouvons donc aborder le problème de la similitude sémantique entre mots par le biais de la notion de distance ou de similitude entre les vecteurs associés. Pour abus de langage, nous parlerons souvent de clusters de mots dans la suite de ce chapitre, mais dans la pratique, un algorithme de clustering manipule les vecteurs associés.

3.2.1 Mesures de l'écart sémantique

Soit l'ensemble de vecteurs associés aux mots d'un corpus, $X = \{\vec{x_1}, \vec{x_2}, ..., \vec{x_n}\}$.

Une **distance** sur l'ensemble X est une application $d : X^2 \to [0, \infty[$ telle que :

- $\forall(\vec{x}, \vec{y}) \in X^2, d(\vec{x}, \vec{y}) = 0$ si et seulement si $\vec{x} = \vec{y}$,
- $\forall(\vec{x}, \vec{y}) \in X^2, d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x})$,
- $\forall(\vec{x}, \vec{y}, \vec{z}) \in X^3, d(\vec{x}, \vec{y}) \leq d(\vec{x}, \vec{z}) + d(\vec{z}, \vec{y})$.

Une **similitude** sur l'ensemble X est une application $s : X^2 \to [0, \infty[$ telle que :

- $\forall(\vec{x}, \vec{y}) \in X^2, s(\vec{x}, \vec{y}) = s(\vec{y}, \vec{x})$,
- $\forall(\vec{x}, \vec{y}) \in X^2, s(\vec{x}, \vec{y}) \leq s(\vec{x}, \vec{x})$.

En règle générale, plus la similitude entre deux vecteurs est élevée, plus ils sont proches, et plus la distance entre eux est faible. Puisque $s(\vec{x}, \vec{x})$ est la similitude maximale pour un $\vec{x} \in X$, nous pouvons convertir une similitude s en une distance d par la formule :

$$d(\vec{x}, \vec{y}) = s_{max} - s(\vec{x}, \vec{y}) \tag{3.1}$$

Où s_{max} est la similitude maximale dans X.

Dans le clustering, pour comparer deux vecteurs, le plus courant est d'utiliser la distance euclidienne ou la similitude cosinus. Mais on peut utiliser toute autre distance ou similitude.

Soient les vecteurs $\vec{x} = (x_1, x_2, ..., x_p) \in \mathbb{R}^p$ et $\vec{y} = (y_1, y_2, ..., y_p) \in \mathbb{R}^p$, $p \in \mathbb{N}^*$.

La **distance euclidienne** entre $\vec{x}$ et $\vec{y}$ est définie comme :

$$de(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2} \tag{3.2}$$

La **similitude cosinus** entre $\vec{x}$ et $\vec{y}$ est définie comme :

$$sc(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \times \|\vec{y}\|} \tag{3.3}$$

Où $\vec{x} \cdot \vec{y}$ est le produit scalaire entre les vecteurs $\vec{x}$ et $\vec{y}$, $\|\vec{x}\|$ et $\|\vec{y}\|$ sont respectivement les normes de $\vec{x}$ et $\vec{y}$. La similitude cosinus maximale est égale à 1.

On définira selon la formule 3.3 la **distance cosinus** entre $\vec{x}$ et $\vec{y}$ comme :

$$dc(\vec{x}, \vec{y}) = (1 - \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \times \|\vec{y}\|}) \tag{3.4}$$

3.2.2 Types de partitions

Il y a un grand nombre de méthodes de clustering. Une partie des méthodes que nous avons examinées dans le premier chapitre sont d'ailleurs des méthodes de clustering (de documents et/ou de mots) ou peuvent être considérées comme telles. Mais du point de vue de l'évaluation de la qualité du résultat (l'évaluation indirecte dont nous avons parlé dans l'introduction), on peut repérer deux types d'algorithmes, présentés ici.

Une partition P de V est un ensemble de parties de V, les clusters, non vides et disjointes deux à deux, dont l'union est V. Si w est un élément de V, il existe donc un unique cluster de P comprenant w.

Un algorithme de clustering cherche à faire une partition P de V. Les mots de chaque cluster de P sont sémantiquement reliés. Dans une bonne partition, les clusters doivent donc être le plus homogène possible.

Concernant les liens entre clusters, deux types d'algorithmes se distinguent suivant qu'il y a ou non une topologie des clusters.

S'il n'y a pas de topologie des clusters, l'algorithme construit un sac de clusters (au sens de *bag of words*) : il n'y a pas de relation d'ordre entre les clusters, et chaque cluster a la même relation à tous les autres. Dans ce cas, la seule relation que doivent entretenir les clusters dans une bonne partition c'est d'être maximalement distincts les uns des autres.

S'il y a une topologie des clusters, dans une bonne partition la distance entre les clusters doit refléter la distance entre les ensembles de mots qu'ils contiennent. Autrement dit, certains clusters se ressemblent plus que d'autres, et on retrouve donc, au niveau des clusters, le même type de relations qu'entre les mots.

Par exemple, les noms de pays pourraient être groupés dans un cluster, les noms de villes dans un autre, et les verbes désignant une modalité (vouloir, pouvoir…) dans un troisième ; il est évident qu'il y aura plus de proximité entre les deux premiers clusters qu'avec le troisième. Un cluster rempli de noms comme « volonté, puissance, potentialité » serait en revanche plus proche du troisième.

Il est très vraisemblable que les relations sémantiques entre les mots doivent obéir à une telle topologie globale, qu'il y ait, en plus des relations sémantiques

entre les mots, des relations sémantiques entre les catégories de mots. En revanche, jusqu'à aujourd'hui, il n'existe qu'une seule famille de modèles (dérivée de Self Organizing Map) présentant une topologie fine sur les clusters. Tous les autres modèles construisent un sac de clusters.

L'évaluation de la qualité de la partition est radicalement différente dans chacun des deux types de clustering. Aussi nous présentons séparément les techniques d'évaluation des uns et des autres dans les sections suivantes.

3.3 Bag of Clusters

Nous présentons d'abord les mesures d'évaluation interne de la qualité de la partition, comme elles sont communes à tous ces algorithmes, puis trois modèles représentatifs : un modèle basique, le clustering ascendant hiérarchique, le modèle sans doute le plus utilisé, K-means (avec ses variantes), et un modèle probabiliste, Expectation Maximization.

3.3.1 Évaluation de la qualité

Nous considérons la partition P de k clusters de l'ensemble X des vecteurs, $P = (C_1, C_2, ..., C_k)$. Pour évaluer la qualité de la partition, on utilise la notion d'inertie, fonction de la distance des vecteurs au centre de gravité des clusters ; nous allons définir ces éléments.

Centre de gravité : Le centre de gravité d'un ensemble de vecteurs est le vecteur $\vec{g}$ de coordonnées $(\bar{x_1}, \bar{x_2}, ..., \bar{x_p})$ tel que, pour tout $j \in \{1, ..., p\}$,

$$\bar{x_j} = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \tag{3.5}$$

Inertie totale : L' inertie totale de X par rapport à son centre de gravité $\vec{g}$ est le scalaire :

$$I_{tot} = \frac{1}{n} \sum_{i=1}^{n} d^2(\vec{x_i}, \vec{g}) \tag{3.6}$$

Inertie d'un cluster : L'inertie d'un cluster C_i est définie de la même façon sur les $\vec{x_j} \in C_i$, à partir de son centre de gravité $\vec{g_i}$:

$$I(C_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} d^2(\vec{x_j}, \vec{g_i}) \tag{3.7}$$

Où n_i est le nombre d'éléments de C_i.

Inertie intra-clusters : L'inertie intra-clusters, qui mesure l'homogénéité de l'ensemble des clusters, est définie comme le scalaire :

$$I_{intra}(P) = \frac{1}{n} \sum_{i=1}^{k} n_i I(C_i) \tag{3.8}$$

Inertie inter-clusters : L'inertie inter-clusters, qui mesure la qualité de la répartition entre les clusters, est définie comme le scalaire :

$$I_{inter}(P) = \frac{1}{n} \sum_{i=1}^{k} n_i d^2(g_i, g) \tag{3.9}$$

Décomposition de Huygens : Cette décomposition permet de mesurer globalement la qualité de la partition. Pour toute partition P de X , elle se définit comme le scalaire :

$$I_{tot} = I_{intra}(P) + I_{inter}(P) \tag{3.10}$$

Cette décomposition est illustrée par la figure 3.1.

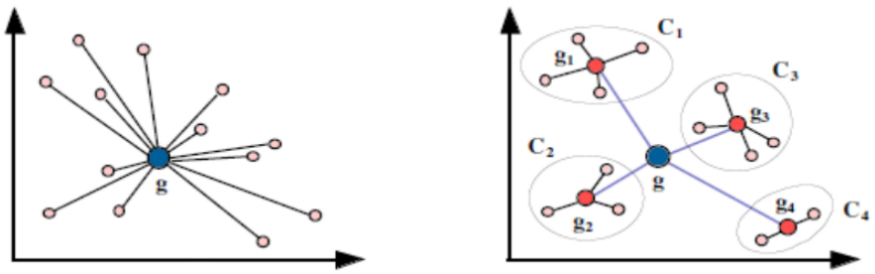

FIG. 3.1: Extrait de CORNUÉJOLS et MICLET (2010) : décomposition de Huygens

Nous constatons que minimiser l'inertie intra-clusters est équivalent à maximiser l'inertie inter-clusters.

3.3.2 Clustering Ascendant hiérarchique (CAH)

Les techniques hiérarchiques de clustering sont des techniques itératives qui hiérarchisent les données dans des arbres binaires appelés dendrogrammes, et comme toutes les autres méthodes géométriques, ces techniques se servent de la distance inter-classes pour rapprocher les données et mettre en évidence leur appartenance à des clusters.

Au départ l'algorithme CAH considère chaque élément de l'ensemble X des vecteurs comme étant un cluster. Pendant la deuxième étape il parcourt tous les clusters et les compare deux à deux ; il les réunit s'ils sont suffisamment proches et les laisse isolés sinon. En récursant sur cette prodédure l'algorithme produit un arbre dans lequel on voit apparaître différents niveaux de clustering.

La racine de cette arbre regroupe l'ensemble des observations. La figure 3.2 illustre un dendrogramme produit par la méthode CAH.

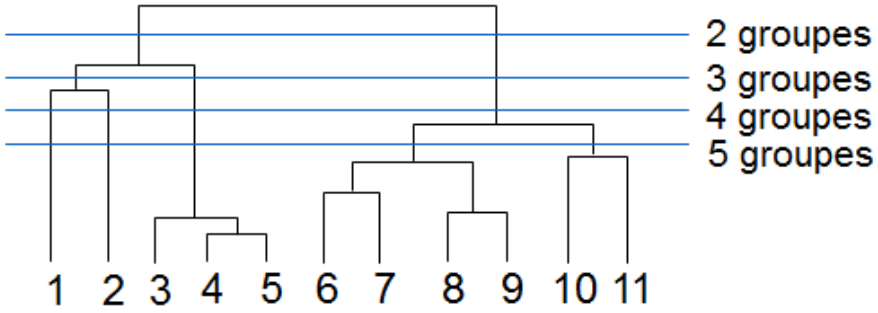

Fig. 3.2: CAH : dendrogramme

Chaque niveau d'un dendogramme constitue une partition sur les données. Cet algorithme produit donc plusieurs partitions.

3.3.3 K-means

La technique de clustering des K-means, ou centres mobiles, a été introduite par McQueen (1967). Elle est réputée pour sa facilité d'utilisation et sa rapidité d'exécution. Elle est utilisée dans de très nombreux travaux, dans l'analyse combinatoire ou le clustering.

Voici une description d'ensemble de l'algorithme, en utilisant la méthode d'initialisation de Lloyd (1982), qui est la plus standard.

Soit l'ensemble E des individus à classer (ici, les mots).

- On choisit k individus au hasard dans l'ensemble E. Ces k individus sont appelés centres.
- On calcule les distances entre tous les individus et les k centres.
- On forme alors k groupes de la manière suivante : chaque groupe est constitué d'un centre et des individus plus proches de ce centre que des autres. On obtient une partition P_1 de E.
- On redéfinit les centres en calculant les centres de gravité des k groupes.
- On calcule les distances entre tous les individus et les nouveaux k centres.
- On forme alors à nouveau k groupes, chaque groupe étant constitué d'un centre et des individus les plus proches de ce centre, ce qui donne une nouvelle partition P_2 de E.
- On itère la procédure précédente jusqu'à ce que deux itérations conduisent à la même partition.

C'est la rapidité et la simplicité de la méthode K-means qui la rendent attrayante, et non sa précision. Cet algorithme présente une imperfection majeure, qui oblige l'utilisateur à définir au préalable le nombre k de clusters. Si c'est supportable pour une application de classification, cela s'avère très désavantageux pour du clustering, où on ne connaît pas apriori le nombre de clusters. De plus K-means est très sensible au choix des centres initiaux. Il existe de nombreux exemples naturels pour lesquels l'algorithme génère de mauvais clusters. Plusieurs méthodes existent pour choisir judicieusement ces centres, comme par exemple, la méthode de Forgy (1965) ou encore celle des nuées dynamiques (Diday, 1971).

Arthur et Vassilvitskii (2007) ont proposé Kmeans++, intégrant dans l'algorithme initial une technique performante pour initialiser les centres. Cette technique consiste à choisir des centres de manière aléatoire à partir des points de données, mais en prenant en compte leur distance au carré au centre le plus proche déjà sélectionné. Cette technique est non seulement efficace, mais de plus elle permet d'accélérer la convergence de l'algorithme de K-means.

3.3.4 Expectation Maximisation (EM)

L'algorithme Expectation-Maximization (EM) proposé par Dempster, Laird et Rubin (1977) est une méthode qui indique si un individus est plus susceptible d'appartenir à l'un des k groupes en produisant une distribution de probabilité.

Nous pouvons voir sur la figure 3.3, que certains cas peuvent être si simples qu'il est possible de produire un clustering en utilisant K-means ou CAH (figure de droite). Dans d'autres cas (figure de gauche), le clustering n'est pas aussi clair et nous ne produisons qu'une probabilité d'appartenir à l'une des classes.

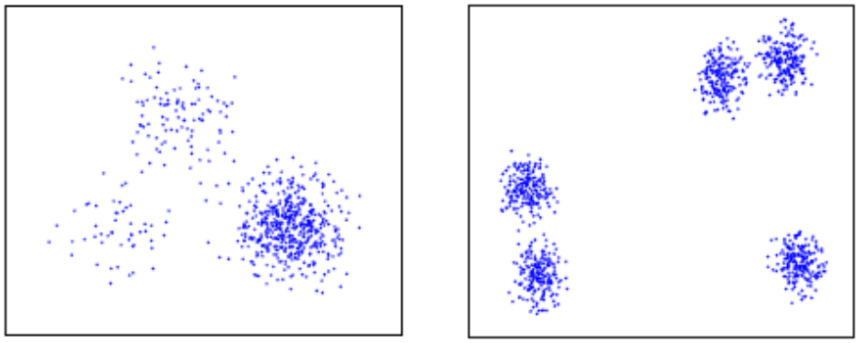

FIG. 3.3: Extraît de LIU et al. (2010) : clustering simple / complexe

Pour bien comprendre le fonctionnement de l'algorithme EM, nous introduisons le modèle de mélange :

Modèle de mélange : Un modèle de clustering probabiliste ou modèle de mélange (*mixture model*) modélise l'hypothèse que, étant donné un point x dans $\mathbb{R}^p$, il est plus susceptible d'appartenir à un cluster (en comparaison avec les autres).

Cette hypothèse peut être quantifiée par une probabilité conditionnelle d'être membre de chaque cluster. Pour être plus précis, considérons une famille $(f_\theta), \theta \in \Theta$ de densités sur $\mathbb{R}^p$ et une distribution de probabilité sur Θ. La densité d'observation de l'individu x peut s'écrire de la manière suivante :

$$\forall x \in \mathbb{R}^p, f(x) = \sum_{k=1}^{K} \pi_k f_{\theta_k}(x) \tag{3.11}$$

Où $\forall x \in \{1, 2, \ldots, K\} \pi_k \geq 0$ et $\sum_{k=1}^{K} \pi_k = 1$, π_k est appelé la proportion du mélange. Un exemple typique de modèle de mélange est le modèle gaussien $(\theta_k = (\mu_k, \sigma_k))$.

Une fois que les paramètres π et θ sont déterminés, le modèle de mélange permet de donner automatiquement le degré d'appartenance de n'importe quel point de $\mathbb{R}^p$ grâce à la règle de Bayes :

$$\mathbb{P}(C_k|x) = \frac{\mathbb{P}(C_k) \times \mathbb{P}(x|C_k)}{\mathbb{P}(C_k)} = \frac{\pi_k f_{\theta_k}(x)}{\sum_{j=1}^{k} \pi_j f_{\theta_j}(x)} \tag{3.12}$$

L'estimation du maximum de vraisemblance (*maximum likelihood estimation*) permet de faire une estimation efficace pour l'équation 3.12. Supposons que les individus de l'ensemble E, $X_1, ..., X_n$ sont générés selon une distribution inconnue $\mathbb{P}_\theta$, où $\theta \in \Theta$ doit être estimé. Pour ce faire nous calculons :

$$Likelihood(\theta|E) = \mathbb{P}_\theta(E) = \prod_{i=1}^{n} \mathbb{P}_\theta(X_i) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_{\theta_k}(X_i) \tag{3.13}$$

Pour calculer le maximum de la fonction *Likelihood* dans l'équation 3.13, on doit calculer sa dérivée. Pour simplifier les calculs, on prend le logarithme de cette fonction ; la nouvelle fonction, appelée *log likelihood*, lh, est calculée comme suit :

$$lh((\pi, \theta)|E) = \sum_{i=1}^{n} \log(\sum_{k=1}^{K} \pi_k f_{\theta_k}(X_i)) \tag{3.14}$$

Les estimateurs $\hat{\pi}_n$ et $\hat{\theta}_n$ sur l'ensemble E sont calculés par la formule suivante :

$$(\hat{\pi}_n, \hat{\theta}_n) = ArgMax_{(\pi,\theta)}(lh((\pi, \theta)|E)) \tag{3.15}$$

L'algorithme EM se déroule en 5 étapes pour estimer les paramètres π_k et θ_k; nous utilisons le temps en exposant, comme dans pi^0; le nombre total d'itérations est noté par T_{max}.

1. **Initialisation :** θ^0 et π^0 sont initialisés aléatoirement.
2. **Log likelihood :** à partir de l'ensemble E on calcule $lh((\pi, \theta)|E)$.
3. **Expectation :** calcul des probabilités conditionnelles π_k pour la valeur courante de θ: $\pi_{k,i}^{(t+1)} = \frac{\pi_{k,i}^{(t)} f_{\theta_k^{(t)}}(x_i)}{\sum_{k=1}^{K} \pi_{k,i}^{(t)} f_{\theta_k^{(t)}}(x_i)}$
4. **Maximisation :** actualisation de l'estimation de θ avec l'équation 3.15.
5. **Test d'arrêt :** si θ ne change plus ou si $t = T_{max}$ (le nombre total d'itérations), on arrête le calcul, sinon on retourne en 2.

Une fois l'apprentissage terminé, il ne reste plus qu'à appliquer la fonction *Likelihood* comme approximation de la règle de Bayes sur le corpus de test.

Cet algorithme n'a pas été élaboré pour le clustering, il a de nombreux autres usages ; par exemple, il sert, en combinaison avec Latent Dirichlet Allocation (voir chapitre précédent) pour faire apparaître les structures sous-jacentes, ou en imagerie médicale. Par rapport au clustering, il n'a d'intérêt que si le clustering est complexe, sinon Kmeans fera mieux l'affaire et sera beaucoup plus rapide.

3.4 Self Organizing Map (SOM)

Il s'agit d'un modèle neuromimétique proposé par Kohonen (1982). Il s'inspire de la structure du cortex et de ses relations avec les signaux envoyés par les capteurs sensoriels. Certaines zones du cortex présentent la même topologie que les capteurs sensoriels ; par exemple deux zones proches dans le cortex visuel correspondent à deux zones proches dans la rétine ; les zones du cortex qui traitent les signaux envoyés par les jambes sont entre celles qui traitent les signaux envoyés par les pieds et celles qui traitent les signaux envoyés par les cuisses (*cortical homunculus*), voir figure 3.4[1]. Les neurobiologistes parlent de projection (*brain mapping*).

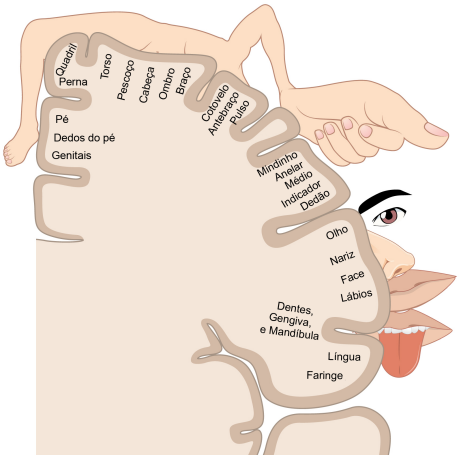

Fig. 3.4: Extrait de Wikipedia (en), Sensory homunculus

Ce modèle, appelé Self Organizing Map (SOM ; il s'appelait autrefois aussi Topology Preserving Map), projette les données d'entrée sur une carte à deux dimensions, avec un apprentissage compétitif non supervisé, pour reproduire la distribution globale des données d'entrée et pas seulement les grouper par catégories. Le principe de base consiste à regrouper les observations qui se ressemblent dans des clusters voisins sur la carte s'ils ne se ressemblent pas assez pour aller sur un même cluster.

La réduction dimensionnelle permet de visualiser des relations de similitude implicites dans les données, difficiles à visualiser dans leur espace d'origine (Mariage, 2001). L'algorithme de Kohonen est une généralisation de l'algorithme d'apprentissage compétitif, y rajoutant la notion de voisinage entre les neurones (Bennani, 2006). Sa topologie des clusters présente un avantage secondaire, comme le remarque Lebboss (2016) : il minimise l'effet d'un mauvais classement, car une

1. `https://en.wikipedia.org/wiki/Cortical\_homunculus`, visité le 2/3/2019.

donnée qui n'est pas classée correctement se retrouvera normalement dans une classe voisine.

Un intérêt particulier de ce modèle, c'est qu'il construit des clusters même pour les données qu'il n'a pas vues pendant l'apprentissage (dont il n'a vu aucun exemple). En effet, à supposer que par exemple les données en question se situent entre deux clusters, avec lesquels elles partagent une partie de leurs propriétés, une fois l'apprentissage terminé elles seront groupées dans la zone intermédiaire entre les deux clusters (même si cette zone n'a jamais attiré de données pendant l'apprentissage).

De nombreux modèles dérivés existent : Hierarchical Feature Map de Miikkulainen (1990), Growing Hierarchical SOM de Dittenbach, Merkl et Rauber (2000), Neural Gas de Martinetz et Schulten (1991)…

3.4.1 Architecture

Le modèle de Kohonen est composé de deux couches : une couche d'entrée de P capteurs, et une couche de sortie de Q neurones disposés en général sur une grille rectangulaire ; on peut les disposer en ligne dite en ficelle, sur une boule ou sur un tore. La couche de sortie constitue la carte topologique auto-organisatrice illustrée par la figure 3.5.

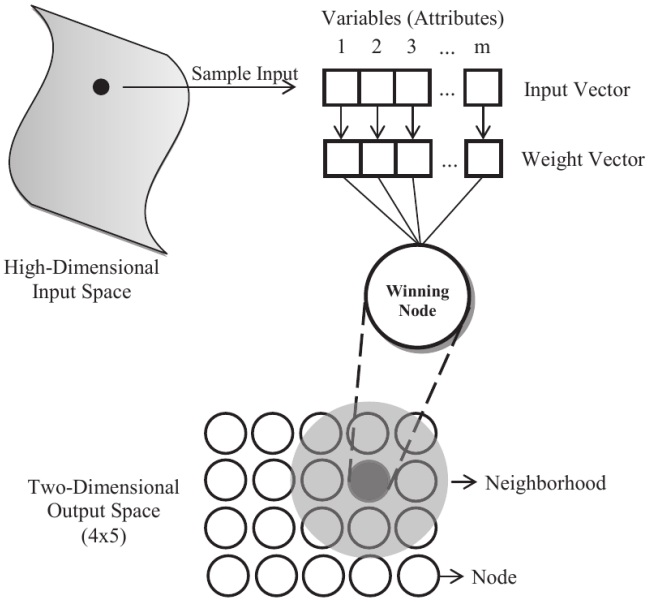

Fig. 3.5: Extrait de Asan et Ercan (2012) : architecture de SOM

Chaque neurone j de la couche de sortie est connecté avec tout les capteurs de la couche d'entrée. W_j est un vecteur de pondérations entre les P capteurs et le neurone j, W_j est la mémoire du neurone j, il est de dimension P d'où :

$W_j = (w_{1j}, w_{2j}, \ldots, w_{pj})$.

On définit un voisinage autour du neurone j de rayon r noté $V_r^t(j)$ qui contient l'ensemble des neurones situés sur la grille à une distance inférieure ou égale à r à l'instant t.

Plusieurs topologies sont possibles, et donc plusieurs voisinages. Le voisinage hexagonal est celui que Kohonen recommande pour la plupart des applications (figure 3.6).

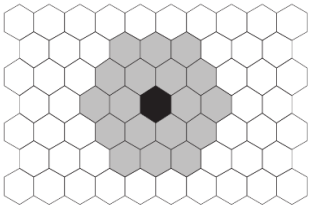

FIG. 3.6: Extrait de ASAN et ERCAN (2012) : voisinage hexagonal

3.4.2 Fonctionnement

Le déroulement de l'algorithme d'apprentissage standard de Kohonen se fait suivant les étapes ci-dessous :

1. On initialise de manière aléatoire les vecteurs mémoire de chaque neurone j de la carte topologique SOM.
2. À chaque itération t, on choisit au hasard un mot, on compare son vecteur X^t à tous les vecteurs de mémoire W_j^t, et on détermine le neurone gagnant (BMU : best matching unit) j^*, celui dont le vecteur mémoire est le plus proche du vecteur X^t (généralement on utilise la distance euclidienne) :

$$d(X^t, W_{j^*}^t) = \min_{\forall j:1 \to N}(d(X^t, W_j^t)) \tag{3.16}$$

   Où N est le nombre de neurones.
3. Ensuite, on rapproche du vecteur X^t le vecteur mémoire du neurone gagnant j^*, ainsi que ceux de ces voisins, selon la règle d'apprentissage décrite ci-dessous :

$$W_j^{t+1} = \begin{cases} W_j^t + \alpha^t V_r^t(j, j^*)(X^t - W_j^t) & pour\ tout\ j \in V_r^t(j^*) \\ W_j^t & pour\ tout\ j \notin V_r^t(j^*) \end{cases} \quad (3.17)$$

$V_r^t(j^*)$ est le voisinage de j^*; il décroit au cours de l'apprentissage. α^t est le taux d'apprentissage ; il décroit lui aussi au cours de l'apprentissage, selon la formule :

$$\alpha^t = \alpha_0(1 - \frac{t}{T_{max}}) \quad (3.18)$$

Où α_0 est le taux d'apprentissage initial, initialisé entre 0.5 et 1.0. T_{max} est le nombre total d'itérations, fixé avant apprentissage. En pratique T_{max} est égale à 10 fois le nombre de vecteurs d'apprentissage.

$V_r^t(j, j^*)$ est la fonction de voisinage qui dépend de la distance dans la carte $d_M(j, j^*)$, r (rayon de voisinage) et du temps t (étape d'itération). La fonction de voisinage la plus adaptée à l'algorithme de Kohonen est une gaussienne, de formule :

$$V_r^t(j, j^*) = \exp(-\frac{d_M^2(j, j^*)}{2\sigma^{t\,2}}) \quad (3.19)$$

Telle que :

$$\sigma^t = \sigma_i(\frac{\sigma_f}{\sigma_i})^{\frac{t}{T_{max}}} \quad (3.20)$$

σ_i et σ_f sont des paramètres initiaux vérifiant $\sigma_i > \sigma_f$. σ^t diminue avec le temps pendant l'apprentissage.

$d_M(j, j^*)$ représente le nombre minimal de connexions topologiques entre j et j^*, par exemple cette distance est égale à 1 entre un neurone donné et ses voisins direct (les plus proches sur la carte).

Remarque : Lorsque $d_M(j, j^*)$ augmente, $V_r^t(j, j^*)$ diminue et quand t tend vers $+\infty$, $V_r^t(j, j^*)$ tend vers 0.

4. Si la condition d'arrêt, $t \geq T_{max}$, n'est pas vérifiée, on reprend à l'étape (2).

Après le processus d'apprentissage, le vecteur mémoire de chaque neurone a une position optimale dans l'espace des vecteurs d'entrée. Le fonctionnement après apprentissage consiste simplement à retrouver le vecteur mémoire le plus proche de la donnée d'entrée.

Comme on l'a vu, même si un neurone est resté vide pendant l'apprentissage, il peut très bien recevoir des données pendant le fonctionnement ultérieur.

3.4.3 Version Batch de SOM

Au lieu de choisir au hasard un seul vecteur d'entrée à chaque itération t et mettre à jour les vecteurs mémoires, comme dans l'algorithme standard (section précédente), dans la version batch (Kohonen, 1993 ; Mulier et Cherkassky, 1995), on présente à chaque itération (epoch) tous les vecteurs d'entrée, et on détermine le neurone gagnant pour chacun d'eux avec l'équation 3.16.

La mise à jour des vecteurs mémoire des neurones se fait à la fin de chaque itération t avec l'équation 3.21.

$$W_j^{(t)} = \frac{\sum_i V_r^{(t)}(j, j^*_{X_i^{(t)}}) \cdot X_i^{(t)}}{\sum_i V_r^{(t)}(j, j^*_{X_i^{(t)}})} \tag{3.21}$$

$W_j^{(t)}$ est le centre de gravité d'un ensemble formé par les vecteurs qui déclenchent le neurone j et les vecteurs qui déclenchent ses voisins, pondérés par la fonction de voisinage $V_r^{(t)}$. Sans la notion de voisinage, l'algorithme Batch de SOM est le même que l'algorithme de Kmeans.

Une étude récente de Melka et Mariage (2017), montre l'efficacité de la version standard de SOM par rapport à la version Batch sur plusieurs données de tests.

On peut ajouter que SOM standard dépend peu de l'initialisation des vecteurs mémoire, alors que SOM Batch y est très sensible.

Afin d'affiner la topologie de la carte SOM, Cabanes et Bennani (2011) ont proposée DDR-SOM (*Data Driven Relaxation - SOM*). Au lieu d'utiliser la distance de Manhattan dans le calcul du voisinage comme dans la version standard de Kohonen, DDR-SOM utilise la distance de Manhattan pondérée entre les neurones sur la carte SOM. Cette pondération prend en compte le degré d'activation des neurones voisins du neurone gagnant (BMU) pour chaque vecteur d'entrée, considéré sur l'ensemble des vecteurs d'entrée. Plus un neurone est activé, plus il sera voisin du neurone gagnant.

3.4.4 Évaluation de la qualité

Comme indiqué dans la section 3.2.2, page 64, l'algorithme de Kohonen construisant une partition avec une topologie, la séparabilité n'y est pas un critère. L'inertie ne nous est donc pas utile, puisque l'inertie intra-classe est liée à l'inertie inter-classe. Ce qui est important pour une évaluation de la qualité c'est de s'assurer que la topologie des clusters reflète bien la topologie des données d'entrée.

Erreur de quantification : Pour mesurer l'homogénéité des clusters, on calcule en général l'erreur de quantification Qe, introduite par KOHONEN, SCHROEDER et HUANG (2001). Il s'agit de la moyenne de la somme des carrés des distances de chaque vecteur d'entrée au vecteur de mémoire du neurone correspondant. Elle est définie par la formule suivante :

$$Qe = \frac{1}{n}\sum_{i=1}^{n} d^2(x_i, w(x_i)) \tag{3.22}$$

Il faudra donc minimiser l'erreur de quantification pour obtenir une meilleure homogénéité des clusters. C'est ce qui remplace dans ce type de partition l'inertie intra-classe.

Qualité de projection ou erreur topologique Te: Pour s'assurer de la qualité de la projection, deux neurones voisins sur la carte doivent avoir des vecteurs mémoires semblables. FLEXER (2001) propose d'utiliser une mesure de corrélation entre la matrice de distance des vecteurs mémoires, d_W, et la matrice de distance des neurones sur la carte, d_N, pour calculer l'erreur topologique[2] (*topological error*).

Pour toute matrice A de taille $k \times k$, notons $\Sigma A = \sum_{i,j} A_{ij}$. Nous calculons la corrélation entre d_W et d_N par la formule suivante :

$$cor(d_W, d_N) = \frac{\Sigma d_W d_N - \frac{1}{k^2}\Sigma d_W \Sigma d_N}{\sqrt{(\Sigma d_W^2 - (\Sigma d_W/k)^2)(\Sigma d_N^2 - (\Sigma d_N/k)^2)}} \in [-1, +1] \tag{3.23}$$

Avec k le nombre de neurones sur la carte.

Une alternative pour calculer l'erreur topologique, proposée par KIVILUOTO (1996) et Kohonen lui-même (KOHONEN, SCHROEDER et HUANG, 2001), prend en compte les vecteurs d'entrée pour mesurer la préservation de la topologie. L'erreur topologique mesure la proportion de vecteurs d'entrée pour lesquels les deux neurones ayant la plus forte activation ne sont pas adjacents sur la carte. Ainsi, plus l'erreur topologique est faible, mieux la topologie de la carte SOM est conservée. Elle est calculée par l'équation 3.24.

$$Te = \frac{1}{n}\sum_{i=1}^{n} u(\vec{X}_i) \tag{3.24}$$

$u(\vec{X}_i)$ est égal à 1, si les deux neurones ayant la plus forte activation pour le vecteur d'entrée $\vec{X}_i$ ne sont pas adjacents et 0 sinon ; n est le nombre de vecteurs d'entrée.

2. Dans la littérature, nous trouvons aussi le terme l'erreur topographique (CABANES et BENNANI, 2011, par exemple) au lieu d'erreur topologique.

U-matrix : U-matrix (*unified distance matrix* en anglais) (Ultsch et Siemon, 1990), est une des techniques d'évaluation visuelle de la topologie des clusters. Elle permet entre autres de visualiser les frontières entre les différents clusters sur la carte topologique.

U-matrix est fondée sur le calcul des distances entre les vecteurs mémoires de neurones adjacents. Ces distances sont présentées avec des couleurs différentes (voir la figure 3.7) ; une coloration sombre entre les neurones correspond à une grande distance et donc à un écart entre les vecteurs mémoires dans l'espace d'entrée. Une légère coloration entre les neurones signifie que les vecteurs mémoire sont proches les uns des autres dans l'espace d'entrée. Les zones claires peuvent être considérées comme des méta-clusters et les zones sombres comme des séparateurs entre méta-clusters. Cela peut être une présentation utile lorsque l'on essaie de regrouper des données sans avoir d'informations a priori sur leur structure.

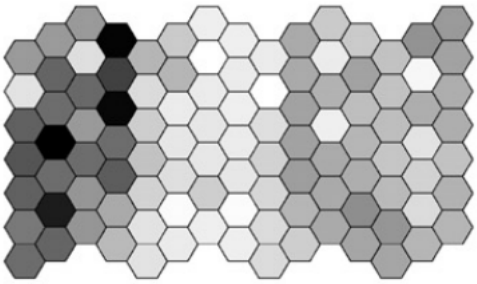

Fig. 3.7: Extrait de O'Malley et al. (2012) : exemple de U-matrix

3.5 Conclusion

Nous avons choisi, pour diminuer les risques de biais dans l'évaluation semi-directe, de prendre plusieurs algorithmes de clustering de types suffisamment différents, pour ensuite pouvoir confronter leurs résultats.

Nous avons donc retenu, pour les expérimentations de la partie suivante, des modèles de chacune des deux catégories ; pour les modèles topologiques, le modèle de Kohonen, et pour les *Bags of Clusters*, K-means, dans sa variante K-means++, et EM. EM a été inclu ici à titre de comparaison surtout parce que c'est l'un des rares algorithmes qui ne soit pas basé sur la distance entre distributions mais sur l'analyse probabiliste des distributions.

Chapitre 4

Évaluation des représentations vectorielles de mots

Sommaire

4.1 Introduction

Dans ce chapitre, nous allons présenter les différentes méthodes d'évaluation des représentations vectorielles des mots dans le cadre de la problématique définie dans notre introduction.

Nous allons établir un panorama des méthodes d'évaluation directe, aussi bien attributionnelle que relationnelle, dans nos deux premières sections. Puis dans la troisième section nous établirons un panorama des méthodes d'évaluation semi-directes à l'aide de la catégorisation sémantique de mots. Comme nous avons déjà vu, dans le chapitre précédent, les méthodes d'évaluation semi-directes internes, nous nous concentrerons ici sur les méthodes externes.

Par conséquent dans ce chapitre toutes les méthodes d'évaluation passées en revue utiliseront des jeux de données de référence, qui seront donc au coeur de notre propos.

Ensuite, dans la quatrième section, nous présenterons les travaux qui portent spécifiquement sur l'évaluation comparée de représentations vectorielles. Nous finirons par une conclusion, incluant une critique des gold standards.

4.2 Évaluation directe attributionnelle

L'évaluation directe attributionnelle consiste à comparer, à partir d'un gold standard constitué de paires de mots établies et reliées manuellement, les relations entre les vecteurs correspondants, et à calculer leurs corrélations.

Le gold standard, ou jeu de données de référence, peut être établi par des experts, mais ici, il repose surtout sur l'intuition d'un ensemble de participants sélectionnés par différentes méthodes, selon la procédure suivante.

1. Un ensemble de paires de mots est établi, soit par un expert, soit par une procédure automatique (choix aléatoire sur des mots fréquents par exemple).

2. Une grille de notation est établie en amont pour noter les similitudes sur une échelle numérique. par exemple entre 0 (mots totalement indépendants) et 10 (mots synonymes).

3. Chaque participant humain reçoit ces paires de mots et il est invité à évaluer le degré de similitude ou de parenté sémantique sur l'échelle constituée. Par exemple, sur une échelle de 0 à 10, il notera 8 la similitude entre «tasse» et «bol», puisque ces mots sont de sens très proches mais pas complètement synonymes.

4. La moyenne des notes données par les participants est calculée et affectée à chaque paire de mot : le gold standard est constitué.

La procédure d'évaluation passe par les étapes suivantes :

1. On prend l'ensemble des vecteurs correspondant à des mots inclus dans le gold standard ;
2. On calcule, pour chaque paire de mot du gold standard, la similitude de leurs vecteurs avec l'une des mesures de distance ou de similitude disponible (voir section 3.2.1) ;
3. On calcule la corrélation, à l'aide d'une des mesures disponibles, entre les notes du gold standard et la similitude des vecteurs.

Parmi les problèmes que rencontre cette évaluation, on a le fait qu'un mot du gold standard ne se retrouve pas dans l'ensemble des vecteurs. C'est ce qu'on appelle *Out Of Vocabulary*, et qu'on abrège le plus souvent par OOV dans les tableaux de corrélation.

La méthode de sélection des participants est malheureusement rarement précisée. Dans certains cas, les participations se sont faites au travers d'une plateforme internet, sans sélection. Dans d'autres, il est indiqué que les participants avaient une maîtrise quasi-parfaite de la langue, ou une maîtrise de la langue. Parfois il n'est rien indiqué.

La langue, dans les gold standards originaux, est toujours l'anglais. Des efforts ont été faits pour en adapter quelques uns à d'autres langues.

Dans la section suivante, nous allons faire un panorama des gold standards originaux couramment utilisés par la communauté des chercheurs du domaine. La section d'après sera consacrée aux adaptations à d'autres langues.

4.2.1 Gold standards originaux

Rappelons que ces gold standards sont tous en anglais. Nous les présentons par ordre chronologique de première publication.

4.2.1.1 RG-65 et MC-30

RG-65 (Rubenstein et Goodenough, 1965) contient 65 paires de mots. La relation sémantique est notée selon une échelle de 0 (indépendance) à 4 (synonymie).

Il y avait 51 participants ayant une maîtrise de l'anglais. Un sous-ensemble de 30 paires, MC-30, a été extrait et soumis à une nouvelle évaluation par 38 nouveaux participants, avec la même échelle(Miller et Charles, 1991).

4.2.1.2 TOEFL Synonyms

Ce jeu de données, introduit par Landauer et Dutnais (1997), est un jeu de données extrait essentiellement de la partie *Synonyms* du test d'anglais TOEFL (Test of English as a Foreign Language). Cette partie est constituée de 80 questions à choix multiples ; pour chacun des 80 mots, quatre mots candidats parmi lesquels un synonyme du mot doit être identifié. Le tableau 4.1 donne des exemples de question, avec le mot à identifier en gras.

Mots	Candidats synonymes
levied	**imposed**, believed, requested, correlated
enormously	appropriately, uniquely, **tremendously**, decidedly
prominent	battered, ancient, mysterious, **conspicuous**
zenith	completion, **pinnacle**, outset, decline

Tab. 4.1: Quelques questions de TOEFL Synonyms

Il n'y a pas à proprement parler de notation, on peut simplement parler d'une échelle binaire (synonyme ou non). Le gold standard est constitué de quintuplets, et celui des quatre derniers mots qui est le synonyme est marqué. Tout se passe comme si nous avions trois paires de mots notées 0 et une paire de mots notée 1. Par conséquent le fonctionnement de l'évaluation des représentations vectorielles reste le même que pour les autres jeux de données.

4.2.1.3 WordSim

WordSim (Finkelstein et al., 2001) contient 353 paires de mots, incluant les paires de RG-65. La relation sémantique est notée selon une échelle de 0 (indépendance) à 10 (synonymie). Il y avait 13 participants ayant une maîtrise quasi-native de l'anglais.

Pour répondre aux critiques sur la similitude (voir section 1.2, page 22), Agirre et al. (2009) ont séparé ce jeu de données en deux parties, en distinguant les paires similaires et les paires apparentées (voir section 1.3 pour les définitions des termes utilisés).

1. WordSim Similarity : contient les paires de mots classées comme synonymes, antonymes, hyponymes et hyperonymes.

2. WordSim Relatedness : contient les paires de mots reliées par une parenté sémantique : méronymes, holonymes, etc. Elle contient également les paires de mots sans aucune relation sémantique.

Certaines paires de mots, dont la relation sémantique était difficile à déteminer, sont inclues dans les deux ensembles à la fois.

4.2.1.4 MTurk

MTurk a donné lieu à une première version, MTurk-287 (Radinsky et al., 2011), puis à une version étendue, MTurk-771, constituées de la même manière.

MTurk-771 (Halawi et al., 2012) contient 771 paires de mots (287 dans la version précédente). La relation sémantique est notée selon une échelle de 0 (indépendance) à 5 (synonymie). Il y avait 20 participants (23 dans la version précédente), des internautes connectés à la plateforme Amazon Mechanical Turk[1]. Pour les sélectionner, ils ont introduit dans le test 10 paires de mots extraites de WordNet (ici page 91) contenant de vrais synonymes ; les internautes ne parvenant pas à les repérer étaient éliminés (98% sur plus d'un millier).

4.2.1.5 Rare-Word

Rare-Word (Luong, Socher et Manning, 2013) contient 2034 paires de mots. Les premiers mots dans chaque paire sont sélectionnées parmi les mots de faible occurrence dans le dump de Wikipédia et qui sont indexés dans WordNet (ici page 91) ; les deuxièmes mots dans chaque paire ne sont pas nécessairement rares). La relation sémantique est notée selon une échelle de 0 (indépendance) à 10 (synonymie).

Les participants étaient également des internautes connectés à la plateforme Amazon Mechanical Turk. Il leur était demandé d'indiquer pour chaque paire s'ils connaissaient le premier mot, le deuxième mot ou les deux. Ils utilisent ces informations pour collecter des notations fiables pour chaque paire. Au final, 10 notations ont été retenues par paire de mots.

Ce gold standard est disponible sur `https://nlp.stanford.edu/~lmthang/morphoNLM/`.

4.2.1.6 Verb-143

Verb-143 (Baker et Reichart, 2014) contient 143 paires de verbes, constituées à partir des 122 lemmes de verbes apparaissant au moins 10 fois dans un cor-

1. `https://fr.wikipedia.org/wiki/Amazon\_Mechanical\_Turk`

pus qui regroupe des documents de la législation du travail et des documents sur l'environnement, en ne conservant que les paires dont la note était supérieure à 0. La relation sémantique est notée selon une échelle de 0 (indépendance) à 10 (synonymie). Il y avait 10 participants anglophones.

4.2.1.7 MEN

MEN, nommé par les acronymes des prénoms des auteurs du gold standard (Bruni, Tran et Baroni, 2014), est constitué de 3000 paires de mots, choisies aléatoirement parmi des mots apparaissant au moins 700 fois dans les corpus *ukWaC* et *Wackypedia corpora combined*[2], et au moins 50 fois dans le jeu de données *ESP game dataset*[3]. Ils ont ensuite échantillonné les paires de manière à ce qu'elles représentent une distribution équilibrée de relations sémantiques. La relation sémantique est notée selon une échelle de 0 (indépendance) à 50 (synonymie).

Les participants étaient également des internautes connectés à la plateforme Amazon Mechanical Turk (via l'interface CrowdFlower[4]). Comme dans MTurk, un test de contrôle a été mis en place pour sélectionner les participants les plus fiables. Ce test consiste à proposer des paires de mots, extraites de WordSim, alternativement une paire à score de similitude élevée et une paire à score faible. On ne garde que les participants capables d'identifier correctement au moins 70% des paires à score élevé.

4.2.1.8 SimLex-999

Comme nous l'avons expliqué section 1.2, page 22, Hill, Reichart et Korhonen (2015) présentent SimLex-999 comme un gold standard différent des précédents. En effet, les consignes données aux participants pour la construction de SimLex-999, insistent sur la relation d'équivalence entre les mots pour bien la distinguer des autres relations sémantiques (hiérarchie, association). Ceci explique la différence entre les scores dans SimLex-999 et dans WordSim. Par exemple, la note moyenne normalisée donnée pour la paire « clothes - closet » (en français : « vêtement - placard ») est de 0.38 dans SimLex-999 et de 0.8 dans WordSim-353. En effet il y a une faible similitude sémantique entre les mots « vêtements » et « placard », mais une relation d'association forte entre eux.

SimLex-999 contient 999 paires de mots. Ces paires se répartissent en trois catégories grammaticales : 666 paires de noms, 222 paires de verbes et 111 paires d'adjectifs. Elles contiennent une sélection équilibrée de concepts concrets («dog»,

2. `http://wacky.sslmit.unibo.it/doku.php`
3. `https://en.wikipedia.org/wiki/ESP\_game`
4. `http://crowdflower.com/`

«hen», «bed»…) et de concepts abstraits («rhythm», «depth», «fever»). Cette répartition variée permet une analyse fine de la performance des modèles de représentation vectorielle, et, selon les auteurs, elle encourage le développement de modèles avec une gamme d'applications différente, et sans doute plus large que celles qui reflètent une association conceptuelle.

4.2.2 Autres langues

Hassan et Mihalcea (2009) ont demandé à des locuteurs natifs de l'espagnol, du roumain et de l'arabe, qui maîtrisaient également très bien l'anglais, de traduire les mots des gold standards MC-30 et WordSim-353.

Les participants ont été invités à traduire chaque mot en tenant compte de son appartenance à la paire de mots en anglais, pour lever l'ambiguïté des mots lorsque plusieurs traductions étaient possibles. Les traducteurs ont également été autorisés à utiliser des mots de remplacement pour surmonter les termes argotiques ou à préjugés culturels. Par exemple, le mot «buck» dans la paire de mots «dollar-buck» a été traduit en arabe par le mot pour «dinar».

Une fois les nouvelles paires constituées, elles ont été soumises pour notation à des participants maîtrisant parfaitement la langue considérée.

Barzegar et al. (2018), afin de soutenir le développement de ressources linguistiques et de représentations vectorielles pour les langues autres que l'anglais, ont traduit quatre gold standards en onze langues, indiquées dans le tableau 4.2. Ils ont ensuite demandé à des participants maîtrisant la langue considérée de noter ces paires, indépendamment des notations de Hassan et Mihalcea (2009). Les gold standards résultants sont disponibles sur le web[5].

Le tableau 4.3 montre des exemples de traductions en français pour cinq paires de mots de SimLex-999; le nombre à droite est la notation moyenne des paires.

5. `https://github.com/Lambda-3/Gold-Standards/tree/master/SemR-11`

Langues	MC-30	RG-65	WordSim-353	SimLex-999
Allemand	✓	✓	✓	✓
Français	✓	✓	✓	✓
Russe	✓	✓	✓	
Italien	✓	✓	✓	✓
Néerlandais	✓	✓	✓	✓
Chinois	✓	✓	✓	
Portugais	✓	✓	✓	✓
Suédois	✓	✓	✓	✓
Espagnol	✓	✓	✓	✓
Arabe	✓	✓	✓	
Persan	✓	✓	✓	

TAB. 4.2: Gold standards traduits

anglais	français
smart - intelligent 9.2	intelligent - intelligent 10.0
hard - difficult 8.77	dur - difficile 9.69
woman - wife 5.72	femme - femme 10.0
wood - paper 2.88	bois - papier 1.15
lover - companion 5.97	amoureux - compagnon 3.62

TAB. 4.3: Comparaison anglais / français sur SimLex-999

4.3 Évaluation directe relationnelle

Bien que l'idée de l'importance des similitudes relationnelles dans l'évaluation remonte au moins à TURNEY (2006), c'est surtout avec l'apparition de Word2Vec (MIKOLOV, SUTSKEVER et al., 2013), que le besoin de disposer de gold standards pour l'évaluation relationnelle est devenu crucial. En effet, les auteurs de Word2Vec ont constaté que leurs vecteurs résultants présentaient des parallélismes relationnels et ils ont voulu les mettre à l'épreuve à grande échelle, à l'aide d'un gold standard ad hoc (il n'en existait pas encore).

Depuis, ce type d'évaluation, en général appelée évaluation par analogies entre mots, est devenu un standard. L'objectif est de mesurer la capacité du modèle à trouver un mot étant donné trois autres mots, en se basant sur la proportion analogique entre deux paires de mots (A est à B comme C est à D)

Un gold standard dans cette catégorie est une base dans laquelle deux couples de mots sont liés entre eux. Le tableau 4.4 en donne quelques exemples.

Le fonctionnement de l'évaluation par analogies entre mots est le suivant :

Catégorie	Quadruplet	
capitale	Athens : Greece	Baghdad : Iraq
genre	uncle : aunt	he : she
comparatif	bad : worse	good : better
gérondif	look : looking	discover : discovering
nombre	banana : bananas	bird : birds

TAB. 4.4: Exemples d'analogies

1. Parmi les quadruplets du gold standard, on choisit ceux dont au moins trois mots possèdent des représentations vectorielles produites par le système à évaluer.

2. Soit le quadruplet analogique A : B ‖ C : D. On devrait avoir $\vec{V}_A - \vec{V}_B \simeq \vec{V}_C - \vec{V}_D$.

3. Supposons que nous voulions vérifier si le modèle est capable de trouver A étant donné B, C et D. A doit vérifier $\vec{V}_A \simeq \vec{V}_C - \vec{V}_D + \vec{V}_B$.

4. Il ne reste plus qu'à trouver le vecteur le plus proche de $\vec{V}_C - \vec{V}_D + \vec{V}_B$; si c'est bien $\vec{V}_A$, le modèle a réussi le test.

Par exemple, Paris est à la France ce que Tokyo est au Japon ; étant donné la relation Paris : France et le mot Tokyo, on doit pouvoir déduire Japon (quatrième proportionnelle).

Le principal gold standard du domaine (le seul à notre connaissance à cette échelle) a été réalisé par Mikolov et son équipe, en deux versions, MSR Analogy (du temps où ils étaient chez Microsoft) et Google Analogy (du temps où ils étaient chez Google). Le second reprend la totalité des données du premier, et le complète. Ils sont en anglais.

4.3.1 MSR Analogy

Le gold standard MSR Analogy (MIKOLOV, YIH et ZWEIG, 2013) contient 8000 analogies, construites à partir des 100 mots les plus fréquents dans chacune des catégories grammaticales noms, verbes et adjectifs possédant un comparatif. Chaque analogie a la forme de deux paires de mots «A : B, C : D». Le tableau 4.5 en donne quelques exemples.

Ce gold standard est disponible sur le site de Microsoft[6].

6. `http://research.microsoft.com/en-us/projects/rnn/`

Catégorie	Quadruplet	
noms	everyone : everyone's year : years	agency : agency's law : laws
verbes	be : was make : makes	return : returned support : supports
adjectifs	good : better high : higher	rough : rougher bad : worse

TAB. 4.5: MSR Analogy : Exemples d'analogies

4.3.2 Google Analogy

Basé sur la même structure, ce gold standard comporte 19 544 analogies[7], avec 8869 analogies grammaticales comme «write : writes ; eat : eats» et 10.675 analogies sémantiques comme «boy : girl , brother : sister».

MIKOLOV, SUTSKEVER et al. (2013) et PENNINGTON, SOCHER et MANNING (2014) ont utilisé ce gold standard pour évaluer leurs représentations vectorielles.

4.4 Évaluation semi-directe externe

Comme nous l'avons indiqué à la section 1.4, l'évaluation semi-directe externe consiste d'abord à appliquer un algorithme de clustering sur les représentations vectorielles de mots, puis à évaluer les regroupements de mots du vocabulaire en en mesurant la cohérence sémantique à l'aide d'un gold standard.

Les gold standards ici doivent représenter une cohérence sémantique sur chaque groupe de mots. La ressource idéale serait un ensemble conséquent de mots classés par catégories sémantiques pour servir à étalonner les clusters produits.

Il existe assez peu, comparativement, de gold standards de ce type ; certains chercheurs utilisent leurs propres jeux de données, ou ils évaluent avec une indirection supplémentaire (il existe beaucoup de gold standards concernant les documents ou les pages web). On trouvera les trois principaux gold standards utilisés dans ce contexte (pour l'anglais) dans les prochaines sous-sections.

Une partie des chercheurs utilisent pour l'évaluation des ressources sémantiques de type thésaurus, que nous présenterons dans la dernière sous-section.

7. https ://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art)

4.4.1 BM

L'un des premiers gold standards, BM, désigné par l'acronyme de ses premiers auteurs, a été introduit par F. Battig et E. Montague (1969). Il a été étendu par Van Overschelde, Rawson et Dunlosky (2004). BM contient maintenant 5321 mots répartis en 56 catégories sémantiques. Le tableau 4.6 montre un extrait de deux catégories sémantiques. Ce gold standard est par exemple utilisé par Baroni, Murphy et al. (2010).

Catégorie	Mots
precious stone	diamond, ruby, emerald, sapphire, pearl, opal, jade, topaz, amethyst, onyx, garnet, turquoise, gem, aquamarine, gold, silver…
unit of time	hour, minute, second, year, day, century, month, decade, week, millisecond, eon, era, score, millenium, fortnight….

Tab. 4.6: BM : extrait de deux catégories sémantiques

4.4.2 AP

AP (Almuhareb et Poesio, 2005), désigné par l'acronyme de ses auteurs, contient 402 mots répartis en 21 catégories sémantiques. Le tableau 4.7 montre un extrait de deux catégories sémantiques.

Catégorie	Mots
motivation	compulsion, conscience, deterrence, disincentive, dynamic, ethics, impulse, incentive, incitement, inducement, life, mania, morality, motivator, obsession, occasion, possession, superego, urge, wanderlust
solid	concavity, corner, crinkle, cube, cuboid, cylinder, dodecahedron, dome, droop, fluting, icosahedron, indentation, jag, knob, octahedron, ovoid, ring, salient, taper, tetrahedron,

Tab. 4.7: AP : deux catégories sémantiques

4.4.3 BLESS

BLESS (Baroni et Lenci, 2011), acronyme de ***B****aroni and* ***L****enci* ***E****valuation of* ***S****emantic* ***S****paces*, contient 200 mots répartis en 27 catégories sémantiques. Bien

que *BLESS* ait été conçu au départ pour un autre type d'évaluation, Jastrzebski, Lesniak et Czarnecki (2017) par exemple l'ont utilisé pour évaluer la catégorisation de mots. Le tableau 4.8 montre un extrait de deux catégories sémantiques.

Catégorie	Mots
water animal	carp, catfish, cod, dolphin, goldfish, herring, mackerel, salmon, trout, tuna, whale,
vegetable	beet, broccoli, cabbage, carrot, cauliflower, celery, corn, cucumber, garlic, lettuce, onion, parsley, potato, radish, spinach, turnip

Tab. 4.8: BLESS : extrait de deux catégories sémantiques

4.4.4 Méthodes à base de thésaurus

Les thésaurus, au sens large (incluant les ontologies et les dictionnaires), sont des bases de connaissances construites manuellement par des linguistes professionnels et des ingénieurs ontologues. Ils sont censés représenter la connaissance dans un domaine particulier ou en général. Les connaissances y sont hiérarchisées en arbres, avec des liens non arborescents, à la manière des réseaux sémantiques ou des graphes conceptuels. Les noeuds de ces réseaux sont des concepts, représentés par des mots.

Dans ces thésaurus on peut définir des mesures de similitude ou de distance entre paires de mots, à partir de la longueur du chemin entre les concepts correspondants dans le graphe. Grâce à ces mesures, on peut les utiliser pour évaluer la qualité d'une représentation vectorielle de mots (Agirre et al., 2009).

Tsvetkov et al. (2015) ont proposé la méthode QVEC[8] qui repose essentiellement sur la quantification de la similitude entre une représentation vectorielle et une ressource linguistique de type thésaurus.

L'idée de base est de faire un alignement de la matrice des représentations vectorielles de mots avec une autre matrice construite à partir du thésaurus. Cette dernière est une matrice de cooccurrences *mot* × *catégorie*, la catégorie étant un concept et les mots étant ceux qui sont reliés à des concepts en relation avec cette catégorie. Par exemple, le concept étant *animal* (avec les mots «animal, bête»), les concepts reliés peuvent être des sous-catégories, ce qui donne les mots «oiseau, mammifère, poisson».

8. https ://github.com/ytsvetko/qvec

Ács et Kornai (2016) ont proposé une autre méthode basée sur des dictionnaires : on construit une matrice carré de cooccurrences $mot \times mot$. Chaque composante c_{ij} dans cette matrice représente le nombre de fois où le mot j apparait dans la définition du mot i. Puis on calcule la corrélation entre les vecteurs mots dans cette matrice et les vecteurs mots donnés par une méthode de représentation vectorielle.

La seule ressource ici qu'on retrouve de manière générale dans les évaluations existantes, c'est WordNet, que nous présentons brièvement ci-dessous.

4.4.4.1 WordNet

WordNet (Miller, 1995) est une grande base de données lexicale de l'anglais, développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. Les mots (noms, verbes, adjectifs et adverbes) sont regroupés en ensembles de synonymes cognitifs (synsets). Un synset représente un sens ou un concept particulier. Les synsets sont interconnectés au moyen de relations conceptuelles – sémantiques et lexicales.

Il s'agit donc d'un thésaurus qu'on peut considérer comme une ontologie de haut niveau (pas une ontologie de domaine), et il lui est associé SUMO (*Suggested Upper Merger Ontology*), une méta-ontologie qui relie de nombreuses ontologies de domaine. Selon Chaumartin (2007), «[WordNet] est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991». La version 2.1 (Fellbaum, 1998) répertorie plus de 200 000 mots, ainsi que plus de 115 000 synsets ; sa dernière version (3.1)[9] est actuellement disponible uniquement en ligne.

WordNet est devenu une des ressources les plus utilisées dans les applications de compréhension ou d'interprétation automatique, les moteurs de question-réponse, le résumé automatique, et tout spécialement pour les tâches impliquant de la désambiguïsation sémantique. WordNet était un composant clé du système informatique Watson (Ferrucci, 2012) de Jeopardy[10], utilisé par IBM. Il est à couverture large.

Avec EuroWordNet (Vossen, 1998), la ressource est vraiment devenue multilingue, assurant la production de WordNets pour plusieurs langues européennes (néerlandais, italien, espagnol, allemand, français, tchèque et estonien). Le mot wordnet est pratiquement devenu un nom commun, et le WordNet original est maintenant appelé Princeton WordNet. Les wordnets de EuroWordNet ont été construits sur la structure de Princeton WordNet version 1.5, avec la même no-

9. https ://wordnet.princeton.edu/download/current-version
10. Watson est devenu en 2011 champion du jeu télévisé américain Jeopardy

tion de synset. Un mécanisme de liaison, appelé InterLingual Index, permet de trouver des synonymes dans n'importe quelle langue, et a ouvert la porte à des applications de traduction. Le projet EuroWordNet a fourni un cadre sémantique commun à toutes les langues, tout en représentant les propriétés spécifiques de chaque langue.

Le projet EuroWordNet s'est achevé à l'été 1999. Néanmoins, de nombreux autres instituts et groupes de recherche développent des WordNets similaires dans d'autres langues (européennes et non européennes), par exemple Balkan WordNet. En raison de notre choix de langue, nous nous intéressons ici particulièrement aux ressources pour le français et l'arabe.

Pour le français, on peut regretter que le WordNet français de EuroWordNet n'ait pas été mis sous licence libre. De ce fait, il est resté figé et inutilisé. Heureusement, d'autres travaux ont permis de combler ce vide.

En particulier, Sagot et Fišer (2008) ont introduit WOLF (Wordnet Libre du Français), construit sur le modèle de Princeton WordNet avec diverses ressources multilingues. Wolf est, selon le terme consacré, un *silver standard*, car il a été produit automatiquement à partir de Princeton WordNet et de ressources en français, et qu'il n'a pas été validé par des humains dans la version actuellement disponible. La première version de WOLF contient 32 251 synsets, alors que le WordNet français du projet EuroWordNet n'en contenait que 22 121 (Sagot, 2017).

La construction automatique de WOLF dans sa première version (0.1.4) a donné des synsets erronés. Dans l'exemple donné par Sagot (2017), le mot «chien» appartient à huit synsets différents (parallèlement au mot anglais «dog»), mais certains sont erronés. Pour résoudre ce problème, Sagot et Fišer (2011) et Sagot et Fišer (2012) ont appliqué une technique de désambiguïsation pour la version (0.2) de WOLF. Cette technique est basée sur un classifieur qui valide ou non un synset pour un mot donné. Pour plus de détail se référer à Sagot (2017).

On peut aussi citer le cas de GLAWI, proposé par Sajous et Hathout (2015) à partir de Wiktionnaire, avec des relations de type WordNet. GLAWI contient dans sa première version 1 341 410 articles et est publié sous licence libre.

Pour l'arabe, Arabic WordNet (Black et al., 2006 ; Elkateb et al., 2006 ; Rodríguez et al., 2008), en abrégé AWN, a été créé sur la base de la version 2.0 de Princeton WordNet. Il est encore très loin d'atteindre la couverture de Wolf, mais s'agrandit lentement.

Différentes associations, comme Global WordNet, promeuvent la création de WordNet libres. Un nouveau projet, BabelNet (Navigli et Ponzetto, 2012), prévoit d'intégrer les ressources de type WordNet et les ressources encyclopédiques comme Wikipedia. Les babelsets sont liés aux synsets et aux wikipages, ou bien à

la fusion des deux en cas de correspondance.

Nous rentrerons plus dans les détails techniques de WordNet au chapitre suivant.

4.5 Travaux d'évaluation comparée

Comme nous l'avons vu (section 1.1), la première comparaison des performances de plusieurs représentations vectorielles apparaît en 2011 (McNamara, 2011). Auparavant, l'attention des chercheurs était surtout préoccupée par le classement des tâches de NLP (reconnaissance de la parole, indexation documentaire, résumé automatique, classification de textes), et la représentation vectorielle apparaissait comme un point de passage obligé mais pas souvent envisagée comme un sujet d'étude en soi.

Mais c'est surtout avec l'apparition de Word2Vec (ici, chapitre 2, section 2.3.3) et la vogue des méthodes prédictives de représentation vectorielle que ce sujet commence à attirer des chercheurs qui s'intéressent à savoir si les représentations vectorielles prédictives sont de meilleure qualité que les représentations statistiques.

Nous proposons dans cette section un aperçu des travaux les plus influents dans le domaine.

4.5.1 Baroni et al 2014

Baroni, Bernard et Kruszewski (2014) conduisent un ensemble d'expériences en comparant CBOW (ici section 2.3.3, page 53) aux méthodes statistiques dans plusieurs tâches d'évaluation.

Pour CBOW, ils ont testé :

- activation : softmax hiérarchique vs échantillonnage négatif avec les valeurs 5 et 10 pour k;
- dimension des vecteurs entre 200 et 500 par un pas de 100;
- taille de fenêtre de contexte entre 2 et 5 de chaque côté du mot central.

Au total, ils ont ainsi évalué 48 paramétrages de CBOW.

Pour les méthodes statistiques, ils ont formé les vecteurs mots à l'aide de la boîte à outils DISSECT[11]. Pour calculer les poids de la matrice de cooccurrences, ils ont utilisé PPMI (ici chapitre 2, section 2.2.3, page 36).

11. `http://clic.cimec.unitn.it/composes/toolkit/`

Ils ont utilisé les vecteurs d'origine (vecteurs creux de grande dimension) et des vecteurs compressés, en appliquant la décomposition en valeurs singulières (voir SVD au chapitre 2, section 2.2.6, page 39).

Au total, 36 paramétrages de méthodes statistiques ont été évalués.

Ils ont constaté que CBOW surpasse systématiquement les approches statistiques dans la plupart des tâches. Le tableau 4.9 montre les résultats obtenus.

	similitude			Catégorisation		Analogie	
Méthode	RG-65	WordSim-353	MEN	AP	BM	TOEFL	Google analogie
PPMI	0.74	0.62	0.72	0.66	0.98	0.76	0.49
CBOW	**0.84**	**0.75**	**0.80**	**0.75**	**0.99**	**0.91**	**0.68**

TAB. 4.9: Extrait de BARONI, BERNARD et KRUSZEWSKI (2014) : performances (meilleures configurations)

4.5.2 Levy et al 2015

LEVY, GOLDBERG et DAGAN (2015) ont cherché les meilleurs paramètres pour chaque approche et ont montré que les hyperparamètres utilisés dans les approches prédictives peuvent être adaptés et appliqués dans les méthodes statistiques, pour que ces dernières donnent de meilleurs résultats.

Pour évaluer la contribution de chaque hyperparamètre aux performances des algorithmes, ils ont réalisé un ensemble d'expériences et ont comparé quatre méthodes de représentation différentes, tout en contrôlant les différents hyperparamètres. Ils ont constaté que lorsque toutes les méthodes sont autorisées à ajuster un ensemble similaire d'hyperparamètres, leurs performances sont largement comparables. Le tableau 4.10 compare les quatre méthodes en calculant les meilleurs hyperparamètres pour chacune, avec une fenêtre de contexte de taille 2.

	similitude						Analogie	
Méthode	WordSim Similarity	WordSim Relatedness	MEN	MTurk-771	Rare-Word	SimLex-999	Google	MSR
PPMI	0.755	**0.697**	0.745	0.686	0.462	0.393	0.553	0.306
SVD	**0.793**	0.691	**0.778**	0.666	**0.514**	0.432	0.554	0.408
SkipGram	**0.793**	0.685	0.774	**0.693**	0.470	**0.438**	**0.676**	**0.618**
GloVe	0.725	0.604	0.729	0.632	0.403	0.398	0.569	0.533

TAB. 4.10: Extrait de LEVY, GOLDBERG et DAGAN (2015) : performances (meilleures configurations)

Les méthodes prédictives testées sont SkipGram avec échantillonnage négatif (ici chapitre 2, section 2.3.3, page 53), et GloVe (section 2.3.5, page 57), les méthodes statistiques sont SVD (section 2.2.6 page 39) et PPMI (section 2.2.3 page 36).

Ils concluent qu'il n'existe pas de méthode unique qui donne systématiquement de meilleurs résultats que les autres. Ils ont constaté aussi que l'application des méthodes statistiques sur des grands corpus s'est avérée techniquement difficile, car elles utilisaient trop de mémoire pour être manipulées efficacement. Selon l'évaluation par analogies, SkipGram et GloVe fonctionnent effectivement mieux que PPMI et SVD.

Pennington, Socher et Manning (2014) montrent diverses expériences dans lesquelles GloVe surpasse SkipGram. Cependant, les résultats trouvés par Levy, Goldberg et Dagan (2015) montrent que SkipGram avec échantillonage négatif surpasse GloVe dans toutes les tâches (tableau 4.10).

4.5.3 Schnabel et al 2015

Schnabel et al. (2015) ont mené à leur tour une étude comparative entre six méthodes de representations vectorielles de mots ; deux approches prédictives, CBOW et C&W (ici section 2.3.2, page 51), et quatre approches statistiques, *Hellinger PCA* (ici section 2.2.4, page 37; dans Rémi et Ronan (2013), ce modèle donnait de meilleurs résultats que C&W), GloVe, TSCCA (ici section 2.2.8, page 40) et Sparse Random Projection (ici section 2.2.5, page 38).

Ils considèrent GloVe comme une méthode statistique et non prédictive. Le tableau 4.11 présente les résultats d'évaluations sur plusieurs gold standards pour les six méthodes de représentations. CBOW surpasse les autres méthodes sur 6 des 7 jeux de données.

	Similitude			Catégorisation		Analogie	
Méthode	RG-65	WordSim-353	MEN	AP	BM	TOEFL	Google analogy
CBOW	**0.74**	**0.64**	**0.70**	**0.65**	**0.85**	0.66	**0.52**
GloVe	0.63	0.54	0.64	0.64	0.77	**0.69**	0.42
TSCCA	0.57	0.54	0.56	0.57	0.64	0.58	0.15
C&W	0.48	0.49	0.57	0.60	0.80	0.66	0.10
H-PCA	0.19	0.32	0.21	0.34	0.42	0.54	0.03
Rand Proj	0.17	0.19	0.11	0.21	0.29	0.51	0.01

Tab. 4.11: Extrait de Schnabel et al. (2015) : meilleurs résultats en gras

Les auteurs ont également suggéré qu'un bonne représentation vectorielle devrait définir des voisinages cohérents pour chaque mot. ils ont proposé un test qui

consiste à choisir les deux mots voisins les plus proches d'un mot cible et à ajouter un mot aléatoire. Un évaluateur humain devrait être capable d'identifier l'intrus (le mot ajouté aléatoirement).

Par exemple, dans l'ensemble {*professeur, enseignant, université, national*}, les plus proches voisins du mot cible *professeur* sont *enseignant* et*université*. En effet, ils sont plus sémantiquement liés les uns aux autres qu'au mot *national*, ajouté aléatoirement. Ce test a montré que toutes les méthodes de représentations vectorielles fonctionnent mieux qu'une estimation aléatoire (une estimation aléatoire permettrait d'atteindre une précision moyenne de 0,25 (un mot sur quatre)), ce qui indique qu'il existe au moins une structure cohérente dans chacune d'elles. La figure 4.1 montre la précision de chaque méthode de représentation sur ce test de recherche de l'intrus. La précision est définie comme le nombre d'annotateurs ayant découvert l'intrus divisé par le nombre total des annotateurs

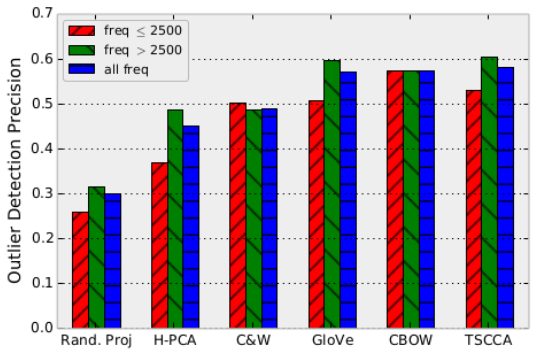

FIG. 4.1: Extrait de SCHNABEL et al. (2015) : Test sur l'intrus

4.6 Conclusion

4.6.1 Limites des gold standards

Les données de référence des gold standards utilisés dans la plupart des travaux posent un certain nombre de problèmes non résolus, que nous précisons ici.

4.6.1.1 Constitution

Dans la constitution des jeux de données, les chercheurs se fondent sur des définitions floues. Les premiers jeux de données sont fondés sur la «similitude sémantique», mais cette notion finit par recouvrir toutes sortes de relations sémantiques. Nous ne savons pas si ces différentes sortes de relations seront capturées par les mêmes mesures de distance / similitude dans l'espace des vecteurs correspondants.

Même la distinction similitude sémantique / parenté sémantique reste encore très floue et difficile à distinguer ; on a vu qu'en divisant WordSim selon ces deux catégories, les chercheurs se sont retrouvés avec des paires qu'ils ne savaient pas classer.

Le mieux en ce domaine serait de se reporter à une différenciation fine, du type de celle indiquée page 22, bien que même cette différenciation fine ne soit pas exempte de difficultés.

Pour ce qui est des analogies, il y a également un flou ; les jeux de données mêlent analogie sémantique et analogie grammaticale (que les linguistes ne traitent pas comme analogie). Mais il est peu probable que ces analogies soient de même nature et qu'elles servent à évaluer les mêmes qualités du modèle. Par ailleurs, même dans les analogies sémantiques, rien ne garantit qu'elles recouvrent toutes la même notion d'analogie.

En fait le seul jeu de données qui de ce point de vue est relativement fiable, ayant été constitué en plusieurs décennies, par des chercheurs en linguistique et en psychologie avertis de ces difficultés, c'est WordNet.

4.6.1.2 Évaluation humaine

Dans les gold standards attributionnels, on fait intervenir des participants pour évaluer les paires de mots (les analogies ne font intervenir pour l'instant les humains que lors de la constitution).

C'est toujours difficile de demander à une personne de représenter son avis ou son sentiment par un nombre dans un intervalle bien précis. Les notes données par des humains sont subjectives, sinon aléatoires, et pourraient refléter la fréquence et l'ordre de présentation des mots plutôt que les facteurs purement sémantiques à évaluer (Avraham et Goldberg, 2016).

L'utilisation d'échelles d'évaluation rend les annotations vulnérables à divers biais (Friedman et Amoo, 1999). Par exemple, le fait que dans WordSim la paire «money, dollar» soit évaluée à 8,42 et la paire «tiger, mammal» à 6.85 est-il dû à une différence sémantique ou à la plus grande fréquence dans l'usage quotidien de ces mots ? Nous ne disposons d'aucune base théorique expliquant la plus grande force dans les relations sémantiques.

Hill, Reichart et Korhonen (2015) signalent par exemple un désaccord entre annotateurs (faible accord inter-examinateur sur les scores de similitude), qui pourrait avoir été causé par des instructions peu claires pour la tâche d'annotation. Bruni, Tran et Baroni (2014) mentionnent le facteur de fatigue des évaluateurs lors de l'annotation d'un grand nombre de paires.

Enfin, nous manquons dans beaucoup de cas d'information sur la fiabilité des annotateurs ; quand un test est indiqué, nous n'avons aucune étude pour savoir quels effets ce test a sur cette fiabilité.

4.6.1.3 Taille et représentativité

La taille des gold standards en nombre de mots reste très faible, par rapport au nombre de mots à évaluer. RG-65 contient seulement 65 paires de mots avec un vocabulaire d'une centaine de mots, et le plus utilisé, WordSim, contient seulement 353 paires de mots avec un vocabulaire de 400 mots.

Même les analogies, qui peuvent apparaître conséquentes en nombre de quadruplets, ne le sont que parce qu'il s'agit justement de quadruplets. Quand on regarde le nombre de mots, on a 300 lemmes pour MSR et à peine le double pour Google Analogy. Rien que pour l'analogie singulier / pluriel, comme il y a environ 10^5 noms en anglais, l'ordre de grandeur de l'inventaire complet serait environ 5×10^9. Quelle serait la taille d'un échantillon représentatif ?

Et du coup la question de la représentativité se pose (Jastrzebski, Lesniak et Czarnecki, 2017). Les mots choisis ne sont pas échantillonnés selon une procédure explicite. Comme l'indiquent Claveau et Kijak (2015), «L'évaluation directe séduit par sa simplicité, mais pose la question de l'adéquation des lexiques utilisés comme références».

De ce fait, les chercheurs essaient de capturer non pas les relations réelles entre les mots dans le corpus d'apprentissage, mais les relations existant dans les données de références, qui ne sont très probablement pas représentatives.

4.6.1.4 Multilinguisme

La quasi-totalité des gold standards est en anglais ; ceux qui ont été portés dans d'autres langues par traduction posent des problèmes, qu'on peut illustrer par la paire «intelligent - intelligent», page 86, qui n'apporte rien à l'évaluation (à part augmenter le score de succès de tous les algorithmes). Ce ne sont certainement pas les gold standards qui auraient été élaborés dans les autres langues si leur construction s'était faite indépendamment.

Par conséquent on a là un véritable obstacle à la mesure des performances d'un

système à travers plusieurs langues, ou même d'une seule langue qui ne soit pas l'anglais.

4.6.1.5 Corrélation entre les gold standards

On ne connait pas la corrélation (s'il y en a une) entre les scores définis par les différents gold standards. Cela rend difficile l'interprétation d'une variation dans l'évaluation. Comment comparer une représentation qui a de bonnes performances avec un gold standard et une mauvaise avec un autre, et une autre représentation qui donne des résultats inverses ?

4.6.2 Choix

Parmi les multiples gold standards actuellement utilisés dans l'évaluation que nous avons passés en revue, WordNet s'impose comme étant le seul à réunir les propriétés suivantes :

- une couverture large ;
- un fondement sémantique scientifique ;
- une différenciation fine des relations sémantiques ;
- des versions en d'autres langues que l'anglais.

Ajoutons à cela que c'est une ressource dont l'ambition n'est pas de sélectionner certains mots, avec le degré d'arbitraire inévitable qui va avec la sélection, mais d'intégrer la totalité des mots. Même si la couverture totale des langues ne sera probablement jamais atteinte, elle progresse et les méthodes pour l'améliorer existent.

Enfin, il ne s'agit pas d'une ressource ad-hoc ; elle a de nombreux usages et aucun d'entre eux n'est à l'origine de sa création, ce qui limite les biais introduits. En effet, les autres gold standards ont été construits pour l'évaluation directe et la seule manière de les évaluer c'est par l'évaluation directe, d'où une circularité dont il faut sortir.

Pour ces raisons, notre choix a été de centrer la suite de notre travail sur l'évaluation semi-directe avec WordNet, l'évaluation directe avec WordNet, sans écarter les autres types d'évaluation, bien entendu.

Deuxième partie

Système réalisé

Sommaire

Chapitre 5

Description du système

Sommaire

Nous présentons ici le système, que nous appellerons *EvalRep*, et les éléments du protocole que nous avons réalisé. Nous commençons par une description générale et technique d'EvalRep, ainsi que les définitions et notations dont nous aurons besoin, puis nous décrivons les méthodes d'évaluation qu'il intègre ; nous introduisons six évaluations inédites.

Les méthodes que nous avons introduites présentent deux caractéristiques liées. D'une part, pour la plupart d'entre elles, sauf la dernière, elles sont pensées par rapport à la déficience de gold standards dans les langues autres que l'anglais. D'autre part, pour la plupart sauf les deux premières, elles mettent à contribution WordNet et ses propriétés particulières, y compris sa topologie.

5.1 Contexte

EvalRep s'intègre dans un grand projet du groupe Data du laboratoire LIAD initié depuis 2014. Ce projet consiste à réunir l'ensemble des outils nécessaires à faire émerger, à partir de grands corpus, les structures de n'importe quelle langue et les propriétés des objets (mots, syntagmes, phrases, documents, voire caractères) dont elle est constituée.

Il y a ainsi eu des développements autour de :

- la détection des propriétés des caractères (Bernard, Aliane et Manad, 2015 ; Manad, Aliane et Bernard, 2016) ;
- la catégorisation sémantique des mots en arabe (Lebboss, 2016) ;
- la détection de contenu dans des pages web (Manad, 2018) ;
- la représentation vectorielle de mots (Lebboss et al., 2017 ; Melka et Bernard, 2017) ;
- le clustering topologique (Melka et Mariage, 2017) ;

D'autres développements sont en cours, comme sur la segmentation et l'analyse syntaxique en arabe (Zaki, 2019). la catégorisation sémantique fine en arabe.

L'ensemble de ces développements s'est effectué avec en vue la réutilisabilité des fonctionnalités communes, et cette exigence, si elle présente pour chacun des chercheurs du groupe une contrainte et un travail supplémentaire, permet aussi de confronter les différents usages et de voir l'utilisation concrète des outils que chacun de nous réalise.

C'est dans ce cadre que nous avons participé à la réalisation de plusieurs de ces composants en dehors de ceux qui rentrent dans le cadre strict de cette thèse et

du système présenté ici, et avons participé à leur exploitation, comme on pourra le voir dans notre liste de publications (juste avant la bibliographie générale à la fin de la thèse).

L'architecture générale des projets de notre groupe de recherche est caractérisée par une librairie Interface qui connecte chaque projet à une base de données PostgresQL (sachant qu'il est possible d'en changer) et qui permet de réunir les différents composants d'un projet, y compris des librairies externes. Les onglets de l'interface utilisateur héritent ainsi de méthodes pour gérer les accès à la base de données, aux url, aux corpus et aux fichiers inclus dans les corpus.

Cette librairie est disponible en open-source sur `https://gitlab.com/Data-Liasd/Interface`.

5.2 Architecture générale

EvalRep inclut ainsi des composants réalisés par d'autres chercheurs, en particulier concernant l'interface général et la construction de corpus. Il est principalement écrit en C++, avec une partie en Python ; il utilise la bibliothèque Qt5, la base de données PostgreSQL 9.4, la bibliothèque spécifique à notre groupe de recherche, et Cmake comme outil de construction. Il tourne sur Linux.

Nous avons réalisé la plupart de nos expérimentations sur les machines de notre laboratoire :

- Thales, 72 processeurs et 125 GiB de mémoire,
- Or, 48 processeurs et 503 GiB de mémoire.

EvalRep est constitué de quatre composants principaux, correspondant chacun à une étape du protocole :

- construction des corpus,
- construction des représentations vectorielles,
- clustering avec les trois méthodes choisies,
- évaluations.

Le composant Construction de corpus (Build Corpus) (figure 5.1) permet de construire le corpus comme un ensemble d'url stocké dans la base de données.

Une partie des corpus pour l'arabe ont été constitués par le constructeur de corpus élaboré par Lebboss (2016). Nous avons constitué les autres corpus à

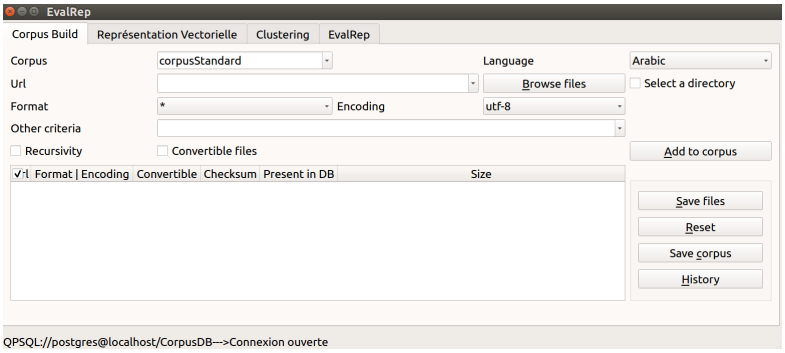

Fig. 5.1: Construction du corpus

partir de Wikipedia, en téléchargeant le dump de Wikipedia[1]. Nous avons utilisé sur ce dump *WikiExtractor*[2], de l'Université de Pise, pour le transformer en fichiers texte sans balises, et *Gensim Extractor*[3] pour le transformer en un seul fichier texte, sans balises, ponctuations et caractères spéciaux (pour Word2Vec et GloVe) ; Gensim Extractor remplace également les nombres par leurs équivalents en chiffres en toutes lettres (ex. 12 devient un deux).

Aussi bien les méthodes de Word2Vec que Gensim Extractor convertissent les majuscules en minuscules ; nous avons donc généralisé ce prétraitement à tous les corpus, quel que soit l'outil de production, pour homogénéiser les données. Pour la même raison, nous avons généralisé la conversion de Gensim Extractor pour les nombres et les caractères spéciaux.

En l'état actuel, le processus complet de construction des corpus à partir de Wikipedia prend environ quatre jours (pour les grands corpus).

Le composant génération des représentations vectorielles. Nous pouvons appeler les méthodes de représentations vectorielles à partir de notre interface graphique (figure 5.2), en choisissant le corpus et les paramètres adaptés à chaque méthode (dimension de vecteurs, fenêtre de contexte, seuil minimum de nombre d'occurrences de mots, etc.). Pour la méthode WebSOM, nous utilisons notre propre implémentation, pour GraPaVec nous n'avons eu accès qu'aux vecteurs produits par Georges Lebboss. Pour les autres méthodes, nous utilisons les implémentations fournies par les auteurs : word2vec (CBOW et SkipGram)[4], GloVe[5] et

1. https ://dumps.wikimedia.org
2. https ://github.com/attardi/wikiextractor
3. https ://radimrehurek.com/gensim/corpora/wikicorpus.html
4. https ://code.google.com/archive/p/word2vec/
5. https ://nlp.stanford.edu/projects/glove/

FastText[6].

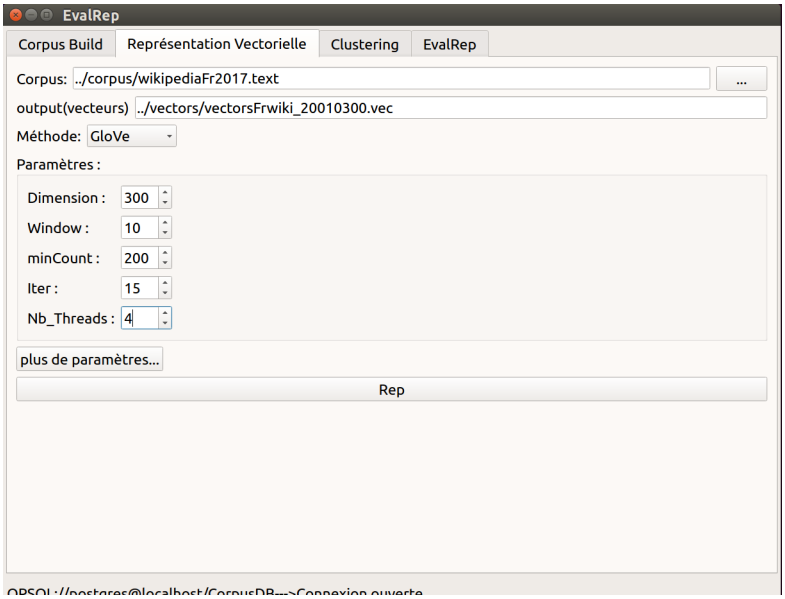

FIG. 5.2: Représentation vectorielle

À la fin de ce processus, nous avons un fichier texte contenant les vecteurs mots. Chaque ligne contient un mot et les composantes de son vecteur séparées par des espaces.

Le composant de catégorisation. La fenêtre du clustering (figure 5.3) nous permet de choisir la méthode de clustering (SOM, Kmeans ou EM). Ensuite, nous chargeons le fichier qui contient les vecteurs de mots et choisissons les paramétres pour chaque méthode. Pour les détails des paramètres et de leurs valeurs, voir page 126.

Nous avons implémenté toutes les mesures d'évaluation internes de la qualité de clustering. L'inertie pour Kmeans et EM, et pour SOM, l'erreur de quantification, l'erreur topologique et U-matrix (pour plus de détail sur ces mesures voir chapitre 3).

Pour les vecteurs de mots creux (notamment produits par la méthode GraPa-Vec), nous utilisons l'algorithme *Sparse_SOM*[7], proposé par MELKA et MARIAGE

6. https ://fasttext.cc/
7. Dans la fenêtre de clustering, il suffit de choisir "sparse" dans le champ de "type de données", par défaut c'est "dense".

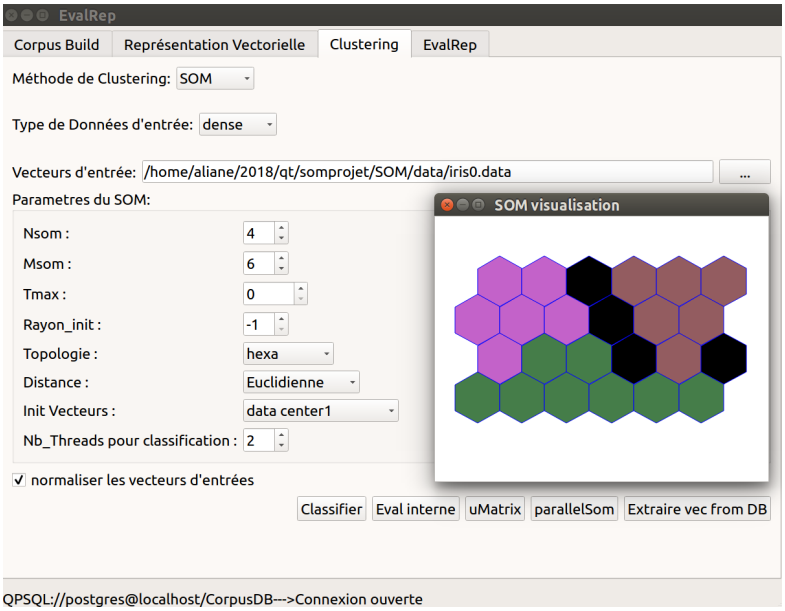

FIG. 5.3: Clustering - Exemple avec SOM et U-matrix sur iris data

(2017). Cet algorithme est très rapide et adapté pour les vecteurs creux (sparse vectors), comme ceux de GraPaVec.

Pour les vecteurs denses, nous utilisons notre implémentation de l'algorithme standard de SOM. La mise à jour des vecteurs mémoire des neurones (codebook) s'effectue en multi-thread.

Pour l'implémentation de Kmeans et EM, nous utilisons Scikit-Learn (PEDREGOSA et al., 2011). Scikit-Learn est un outil libre dédié à l'apprentissage automatique, écrit en Python. Il est simple et efficace pour l'exploration et l'analyse de données.

L'interface du clustering nous permet également de charger les vecteurs de mots à partir de la base de données et d'y sauvegarder les vecteurs mémoire aprés l'apprentissage. Ils peuvent aussi être sauvegardés comme des fichiers.

Le composant d'évaluation. À partir de l'interface d'évaluation (figure 5.4), nous chargeons les vecteurs de mots produits lors des étapes antérieures et choisissons la méthode d'évaluation et ses paramètres.

Pour répondre au problème des vecteurs creux (grande dimension avec une majorité de valeurs nulles), nous avons créé une structure de donnée optimisée, dans laquelle on stocke seulement les éléments non nuls avec leurs indices. Nous avons implémenté également plusieurs fonctions basées sur cette structure et qui

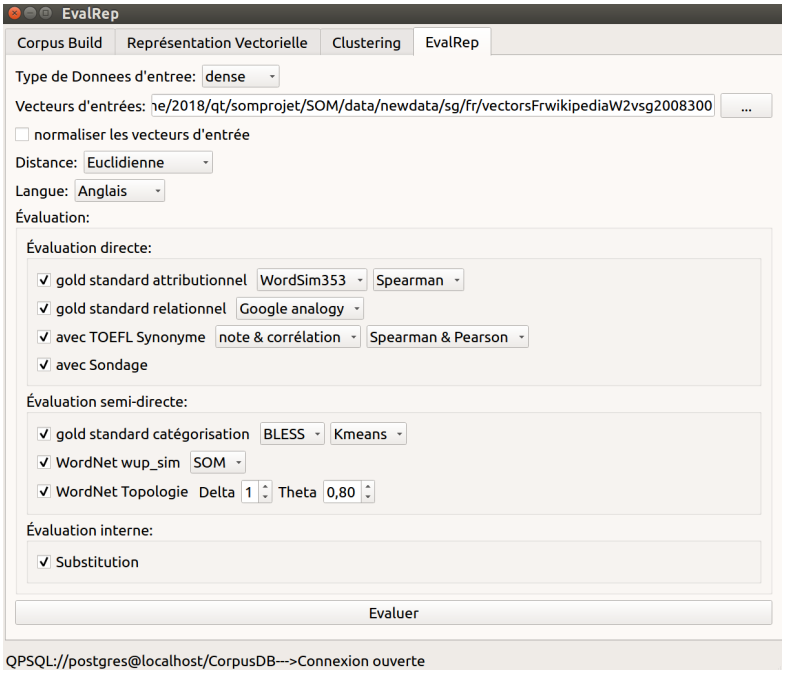

FIG. 5.4: Interface pour le composant d'évaluation

traitent ces vecteurs creux, notamment : la distance euclidienne ou le produit scalaire entre deux vecteurs, la somme ou la multiplication de deux vecteurs, la norme d'un vecteur, etc. parce que les fonctionnalités offertes par Boost (SCHLING, 2011) par exemple étaient trop lentes.

Le composant d'évaluation est le coeur de notre programme. Dans les sections suivantes nous donnons le détail du fonctionnement de chacune des méthodes.

EvalRep permet de cumuler de nombreuses procédures d'évaluation des représentations vectorielles de mots. Sans prétendre à une exhaustivité totale, une large palette de procédures (directes, semi-directes, externes, internes) ont été intégrées dans notre protocole, et nous avons introduit quelques procédures nouvelles. Comme on le verra dans la conclusion de ce chapitre, ces évaluations permettent de générer plus d'une centaine de mesures.

Voici la liste des méthodes d'évaluation que nous avons élaborées, en plus des méthodes reprises de la littérature :

- évaluation externe par sondage, destinée à pallier le manque de gold standards en français et en arabe ;
- évaluation interne par substitution, destinée à permettre une évaluation sans

gold standard ;

- évaluation directe basée sur un gold standard attributionnel tiré de WordNet, destiné à compléter les gold standards existants ;
- évaluation semi-directe basée sur un gold standard de catégorisation tiré de WordNet ;
- évaluation semi-directe sans gold standard, basée sur la mesure *wup_similarity* de WordNet ;
- évaluation topologique, basée sur la comparaison entre la topologie des clusters et la topologie du graphe de WordNet.

Le fonctionnement global d'EvalRep est illustré dans la figure 5.5, qui regroupe toutes les évaluations. Cette figure est complexe et sera détaillée dans la section de chaque type d'évaluation. Pour l'instant, on relèvera les éléments suivants :

- **l'évaluation directe** suit les flèches 1, 2, 8 et 11;
- **l'évaluation semi-directe** suit les flèches 1, 2, 5, 7, 9 et 11;
- **l'évaluation avec WordNet** suit les flèches 1, 2, 3, 6, 4, 7, 9 et 10;
- **l'évaluation directe par sondage** suit les flèches 1, 2, 8 et 12;
- **l'évaluation par substitution** suit les flèches 0, 1, 2 et 8.

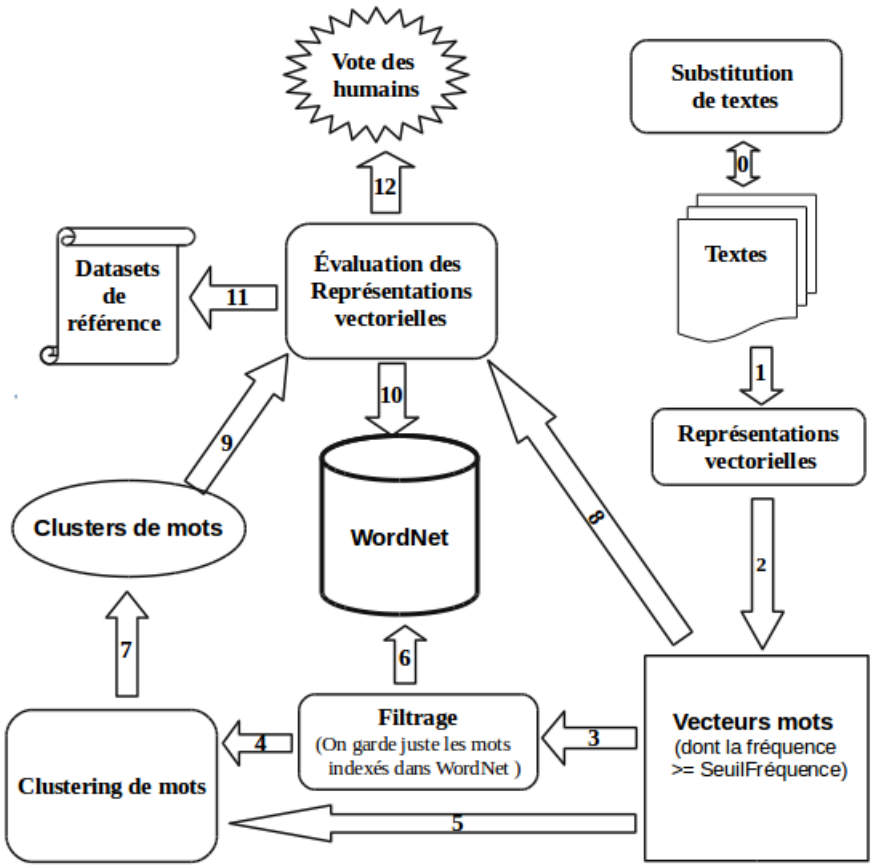

FIG. 5.5: Fonctionnement global d'EvalRep

5.3 Définitions communes

Un certain nombre de définitions sont communes à plusieurs évaluations ; nous les présentons ici.

Nous appellerons **V** le vocabulaire du corpus *Corp* en cours, défini à partir des mots uniques de *Corp* et du seuil **Min** par l'équation 5.1.

$$V = \{mot \in Corp,\ Min \in \mathbb{N} \mid \#(mot) \geq Min\} \tag{5.1}$$

Où #(mot) est le nombre d'occurrences de *mot*, et Min un seuil permettant d'écarter les mots qui n'occurrent pas assez souvent pour que leur vecteur soit significatif.

Si au moins un mot manque dans une paire de mots (pour les gold standards attributionnels) ou dans un quadruplet de mots (pour les gold standards relationnels), cette paire ou ce quadruplet ne peut être utilisé dans l'évaluation. De même pour les mots qui ne figurent pas dans les gold standards de catégorisation.

Nous appellerons **support** d'un gold standard le sous-ensemble du gold standard dont les mots appartiennent tous à V. Le support peut donc être une liste de mots (gold standards de catégorisation) ou une liste de n-uples de mots (autres gold standards). Le support d'un gold standard est un indice de la fiabilité de ce gold standard par rapport à *Corp*.

5.3.1 Distance et similitude

Un paramètre important de notre système dans la plupart des évaluations est **Dist**. Il désigne la méthode utilisée pour calculer la ressemblance entre deux mots, à partir de leurs vecteurs.

Ce paramètre prend deux valeurs : le cosinus de l'angle entre les vecteurs, défini par l'équation 5.2 ou la distance euclidienne entre eux, définie par l'équation 5.3. Ce sont les valeurs les plus couramment utilisées dans les évaluations en général, même si on pourrait tester d'autres types de distances.

Soient les mots m_1 et m_2 et leurs vecteurs $\vec{m_1}$ et $\vec{m_2}$ de dimension $d \in \mathbb{N}^*$:

$$Dist(m_1, m_2) = \frac{\vec{m_1} \cdot \vec{m_2}}{\|\vec{m_1}\|.\|\vec{m_2}\|} \tag{5.2}$$

Dans ce cas, plus $Dist(m_1, m_2)$ est petite, plus les mots sont éloignés.

$$Dist(m_1, m_2) = \sqrt{\sum_{i=1}^{d}(m_{1i} - m_{2i})^2} \tag{5.3}$$

Dans ce cas, plus $Dist(m_1, m_2)$ est petite, plus les mots sont proches.

À partir de **Dist**, on définit le **voisinage** d'un mot m en triant l'ensemble des autres mots d'après $Dist(m)$ (tri décroissant pour le cosinus et tri croissant sinon), puis en définissant un seuil.

5.3.2 Définitions liées au gold standard

Un gold standard peut être défini formellement comme l'ensemble des association de chaque n-uple de mots qu'il contient à une cible, supposée représenter la vérité de terrain (*ground truth*). Le n-uple peut être réduit à un mot (gold standard de catégorisation), à une paire de mots (gold standard attributionnel), etc. La cible peut être un coefficient de similitude, un label, ou un mot.

Soient un gold standard et un algorithme prédisant automatiquement pour chaque n-uple de mots du support de ce gold standard une cible. L'évaluation diffère suivant la nature de la cible.

Si la cible est une valeur continue, nous aurons deux tableaux de valeurs continues, celui du gold standard et celui prédit par l'algorithme. L'évaluation dans ce cas consiste à calculer une corrélation entre ces tableaux.

Si la cible prend ses valeurs dans un ensemble de valeurs prédéfinies, appelées *label*, le résultat sera un classement des éléments du support dans chaque label. L'évaluation dans ce cas consiste à calculer une matrice de confusion et une mesure basée sur cette matrice.

Certaines cibles peuvent être considérées comme l'un ou l'autre, et dans ce cas deux types d'évaluation peuvent être produits (par exemple cas du TOEFL plus loin).

Nous allons examiner chaque cas ci-dessous.

5.3.3 Corrélation

Appelons nos deux tableaux X et Y; tous deux ont N valeurs. Nous devons calculer un coefficient de corrélation entre X et Y, qui nous renseigne sur leur similitude. Les coefficients de corrélation les plus largement utilisés sont ceux de Pearson et de Spearman.

- **Pearson :** La formule la plus simple pour calculer le coefficient π de Pearson est donnée par l'équation 5.4.

$$\pi = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{5.4}$$

Où $\bar{x}$ est la moyenne de X et $\bar{y}$ est la moyenne de Y.

- **Spearman** (Spearman, 1904) : Elle consiste à trouver un coefficient de corrélation, non pas entre les valeurs prises par les deux variables mais entre les rangs de ces valeurs. Elle mesure la relation d'association non paramétrique entre les deux vecteurs.

  La formule pour calculer le coefficient σ de Spearman est donnée par l'équation 5.5.

$$\sigma = 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N^3 - N} \tag{5.5}$$

  Où d_i est la différence entre les rangs de x_i et de y_i dans leurs tableaux respectifs.

5.3.4 Matrice de confusion

Dans le cas où les cibles sont des labels, appelons L_i^{gs} un label du gold standard pour un n-uple donné, et L_j^{aut} un label prédit automatiquement par l'algorithme pour le même n-uple. Deux situations peuvent se produire : soit il existe une correspondance entre l'ensemble L^{aut} et L^{gs} (et cette correspondance est connue au départ), soit non. À chacune de ces deux situations correspond une évaluation différente.

Dans le premier cas, il suffit de vérifier que L_j^{aut} correspond bien à L_i^{gs}. Dans le deuxième cas, il faut vérifier si les n-uples du support pris deux à deux ont le même L_i^{gs} et le même L_j^{aut}. Les évaluations dans chaque cas utilisent une **matrice de confusion**, calculée différemment.

5.3.4.1 Matrice de confusion avec correspondance

Au départ les matrices de confusion ont été élaborées sur des gold standards binaires, c'est à dire à deux classes, avec les labels P (*positive*) et N (*negative*), puis étendues aux gold standards multiclasses. Avec un gold standard binaire et une correspondance connue, la matrice de confusion est définie comme suit, en fonction du label produit par l'algorithme pour un mot donné (P^{aut} ou N^{aut}) :

- TP (true positive) : nombre de cas de P^{aut} avec P en vérité de terrain ;
- TN (true negative) : nombre de cas de N^{aut} avec N en vérité de terrain ;
- FP (false positive) : nombre de cas de P^{aut} avec N en vérité de terrain ;

- FN (false negative) : nombre de cas de N^{aut} avec P en vérité de terrain.

Pour étendre cette définition au multiclasse, il faut d'abord calculer la matrice de confusion pour chaque label, en considérant le label comme P et tous les autres labels comme N. Il suffira ensuite de faire la moyenne des classes dans le calcul de l'indice produit.

On peut définir plusieurs indices à partir de la matrice de confusion : l'exactitude (*accuracy*), équation 5.6, la sensibilité (*sensitivity*), équation 5.7, le rappel (*recall*), équation 5.8, la précision (*precision*), équation 5.9, la F-mesure (Rijsbergen, 1979).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.6}$$

On utilise aussi l'exactitude dans des cas où il n'y a qu'une seule classe (pas de possibilité d'avoir de TN, de vrais négatifs). Dans ce cas l'exactitude se confond avec la sensibilité (équation 5.7).

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.8}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.9}$$

$$F_\beta - mesure = \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{5.10}$$

Le facteur β permet de donner un poids différent aux différentes erreurs. Pour donner plus de poids à FP qu'à FN, on choisit une valeur $\beta > 1$. Quand $\beta = 1$, on parle de F1-mesure ou de F-mesure.

5.3.4.2 Matrice de confusion sans correspondance

Dans le cas où il n'y a pas de correspondance connue, la matrice de confusion est définie sur les couples de n-uples comme suit :

- TP (true positive) : nombre de cas où le couple est associé à un seul label à la fois par l'algorithme et le gold standard ;

- TN (true negative) : nombre de cas où le couple n'est pas associé au même label à la fois par l'algorithme et le gold standard ;
- FP (false positive) : nombre de cas où le couple est associé à un seul label par l'algorithme mais à deux par le gold standard ;
- FN (false negative) : nombre de cas où le couple n'est pas associé au même label par l'algorithme mais est associé au même par le gold standard.

Nous avons introduit une nouvelle manière de construire la matrice de confusion pour les clustering topologiques (ici 5.8.7)

On définit la F-mesure, le rappel et la précision de la même manière que précédemment (par contre la matrice de confusion n'est pas calculée de la même façon).

L'indice de Rand (Rand, 1971) est défini par l'équation 5.11:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.11}$$

On notera que sa définition est exactement la même que celle de l'exactitude (equation 5.6) ; c'est ici aussi la manière de calculer la matrice de confusion qui diffère.

5.4 Évaluation directe

La figure 5.6 montre le principe du fonctionnement de l'évaluation directe dans notre système. En vert, les paramètres qui jouent sur le résultat. Ils sont au nombre de trois : **GS**, le gold standard choisi ; **Dist**, la distance ou similitude (ici page 115) ; et **Corr** (ici page 116), la corrélation. Le *support* (ici page 115) est le nombre de paires ou de quadruplets dont tous les mots figurent dans le *vocabulaire* du corpus (ici page 115).

À partir de l'ensemble des mots extraits du corpus, d'une part, et du gold standard, de l'autre, on recherche le support. Puis on détermine les vecteurs correspondant à une paire ou à un quadruplet du support, et on calcule leur distance ; on effectue ensuite une mesure de corrélation. La corrélation s'applique différemment suivant qu'on effectue une évaluation attributionnelle ou relationnelle (voir section 1.2 page 21).

Par conséquent, pour une évaluation directe, il faut compter 2 valeurs pour **Dist**, 2 valeurs pour **Corr**; le nombre de valeurs pour **GS** dépend du type attributionnel ou relationnel de l'évaluation et de la langue considérée.

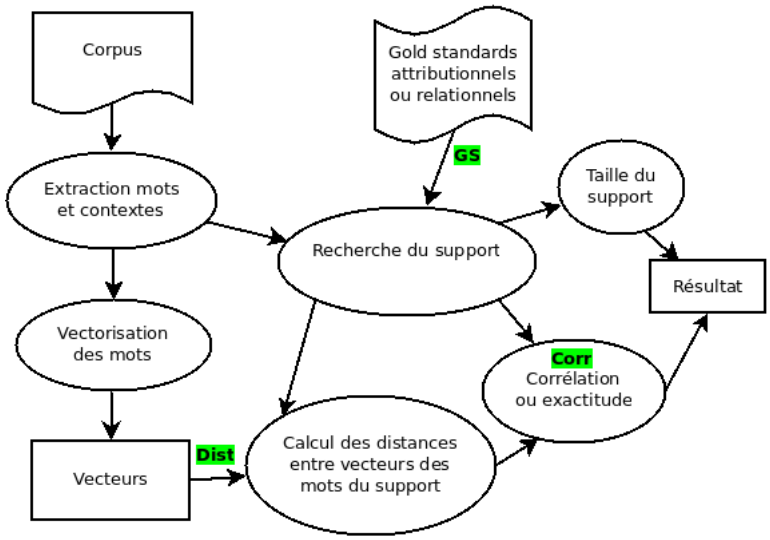

FIG. 5.6: Évaluation directe

Pour une évaluation directe attributionnelle en anglais, avec 9 gold standards, on comptera donc 36 résultats. Pour une évaluation directe relationnelle (en anglais seulement), 4 résultats. Pour une évaluation directe attributionnelle en français ou en arabe, respectivement 16 et 12 résultats.

Le tableau 5.1 présente la liste des gold standards que nous avons utilisé comme valeurs pour **GS** (pour plus de détail, voir ici chapitre 4).

datasets	**type d'évaluation**	**mesure**
MC30	attributionnel	corrélation
RG65	attributionnel	corrélation
WS353	attributionnel	corrélation
MTurk771	attributionnel	corrélation
Rare-Word	attributionnel	corrélation
Verb143	attributionnel	corrélation
MEN	attributionnel	corrélation
SimLex999	attributionnel	corrélation
TOEFL	attributionnel	corrélation / note
Google Analogy	relationnel	exactitude

TAB. 5.1: Gold standards utilisés

5.4.1 Évaluation attributionnelle

Les gold standards attributionnels sont constitués de triplets comme (m_1, m_2, s), avec une paire de mots et un scalaire pour la note de similitude. Sur ces triplets on applique l'algorithme 1 suivant :

Data :
GS$[(m_1, m_2, s)^{(i)},\ i = 1 \dots n]$**;**
V : Vocabulaire du corpus ;
vect(m_i**) : vecteur du mot** m_i**;**
sim[] : table des notes de similitude ;
dist[] : table des $Dist(vect(m_1), vect(m_2))$
Pour i **de** 1 **à** n **faire**
  Si $(\mathrm{GS}[i].m_1 \wedge GS[i].m_2 \in V)$ **Alors**
    $sm = Dist(vect(GS[i].m_1), vect(GS[i].m_2))$;
    ajouter $GS[i].s$ dans $sim[]$;
    ; ajouter sm dans $dist[]$;;
  Fin Si
Fin Pour
calculer la corrélation Pearson(sim, dist) ;
calculer la corrélation Spearman(sim, dist) ;
Résultat : **corrélations Spearman et Pearson**

Algorithme 1: évaluation directe attributionnelle

Une bonne représentation vectorielle est censée obtenir une corrélation élevée ($\gg 0.5$).

Le gold standard TOEFL est un cas un peu à part, car sa structure est différente. Il y a deux manières de le traiter : soit le ramener à la structure générale, puis calculer la corrélation (par ex. Schnabel et al. (2015), qui n'indique pas comment il traite TOEFL), soit noter la représentation vectorielle comme on note un humain passant le test. Nous avons effectué les deux types d'évaluation.

La transformation de TOEFL en triplets (m, m, s) s'effectue de la façon suivante :

- On parcourt les 80 questions de TOEFL ; chaque question est composée d'un mot cible m et de quatre mots proposés, $\{m_1, m_2, m_3, m_4\}$. Parmi ces mots se trouve le mot réponse m^*.
- On associe à chaque paire de type (m, m^*) la note 1, et pour les autres paires (m, m_i) la note 0.

- On obtient au final 320 paires de mots, 80 paires avec une note de 1, et 240 paires avec une note de 0, et il ne reste plus qu'à appliquer l'algorithme ci-dessus.

Sinon, on calcule chaque note sur le nombre de questions qui font partie du support pour lesquelles $Dist(m, m^*)$ donne le meilleur résultat, selon l'équation 5.12. Puis on somme ces notes.

$$Note = \begin{cases} 1 & si\ Dist(\vec{m_i}, \vec{m_i^*}) = \underset{m \in \{m_i^{(1)}, m_i^{(2)}, m_i^{(3)}, m_i^{(4)}\}}{Min} Dist(\vec{a_i}, \vec{m}) \\ 0 & sinon \end{cases} \tag{5.12}$$

Les deux approches ne donnent pas le même résultat, car les supports ne sont pas les mêmes. Dans la première, même si les cinq mots ne sont pas dans le corpus, il suffit que deux y figurent pour avoir une paire valide.

On verra à la section 5.8.7 que nous avons ajouté un gold standard attributionnel généré à partir de WordNet.

5.4.2 Évaluation relationnelle

Google Analogy a une structure en quadruplets de mots du type $\{m_1, m_2, m_3, m_4\}$. Le support est l'ensemble des quadruplets dont tous les mots appartiennent au vocabulaire.

Pour chaque quadruplet i du support on calcule $\vec{v_i}$, le vecteur théorique de $m_4^{(i)}$, à partir des trois premiers, par l'équation 5.13.

$$\vec{v_i} = \vec{m_2}^{(i)} - \vec{m_1}^{(i)} + \vec{m_3}^{(i)} \tag{5.13}$$

Puis on parcourt l'ensemble des mots pour voir si le vecteur de m_4 est bien le plus proche de ce vecteur, selon l'équation 5.14.

$$q_i = \begin{cases} 1 & Si\ \ Dist(\vec{v_i}, \vec{m_4^{(i)}}) = \min\limits_{m \in V} Dist(\vec{v_i}, \vec{m}) \\ 0 & Sinon \end{cases} \tag{5.14}$$

Enfin on calcule le taux d'exactitude ; le gold standard ne possède que des analogies correctes, donc une seule classe, il n'y a pas de TN, par conséquent ce taux est en fait calculé avec l'équation 5.7, page 118), qui devient ici l'équation 5.15.

$$Accuracy = \frac{\sum_{i=1}^{|support|} q_i}{|support|} \tag{5.15}$$

5.5 Évaluation directe par sondage

En raison de la pauvreté des gold standards pour le français et l'arabe, nous avons introduit une autre sorte d'évaluation directe attributionnelle, par sondage auprès de participants français et arabes. Cette évaluation s'inspire de la «comparative intrinsic evaluation» proposée dans Schnabel et al. (2015), mais le protocole est différent. La procédure, schématisée dans la figure 5.7, est la suivante :

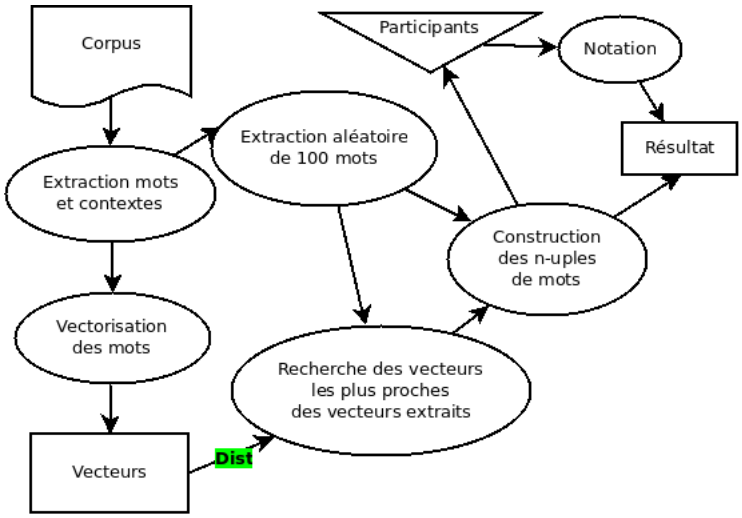

Fig. 5.7: Évaluation par sondage

1. 100 mots de longueur supérieure à 4 (pour éviter les mots trop simples) sont tirés au sort dans le corpus de départ ;
2. les vecteurs correspondant à ces mots pour chaque méthode de vectorisation sont sélectionnés ;
3. pour chacun de ces vecteurs, on recherche le vecteur le plus proche ; s'il y en plusieurs, le plus fréquent est sélectionné, et s'ils ont la même fréquence, le premier par ordre alphabétique ;
4. pour chaque mot on obtient donc un ensemble de mots (ici un triplet) ; on change aléatoirement l'ordre de présentation ;

5. pour éviter la fatigue, la liste de 100 items est divisée en quatre pour être proposée aux participants.

6. chaque participant indique quel est le mot qui lui semble le plus proche du mot cible ; s'il en sélectionne plusieurs ou aucun, on considère que c'est un vote blanc ;

7. on compte le nombre de points obtenus par chaque méthode.

Cette procédure ne possède qu'un seul paramètre, **Dist**. La table 5.2 montre en exemple un extrait d'une liste de mots tirés du sondage fait pour le français avec les propositions de mots faites par trois méthodes.

Mots	**CBOW**	**GloVe**	**WebSOM**
route	autoroute	autoroute	piste
professeur	enseignant	université	sénateur
propose	proposant	offre	offre
mondial	international	unesco	national
ouvrage	livre	livre	album
chapelle	couvent	sainte	chanson

TAB. 5.2: Propositions de mots

Pour le français, les 38 participants étaient des étudiants de l'université Paris 8 ayant le français pour langue maternelle, de différentes filières : licence de sciences de l'éducation, master de philosophie (analyse et critique des arts), licence de psychologie.

Pour l'arabe, les 43 participants étaient des étudiants du master d'informatique de l'université Ibn Khaldoun de Tiaret en Algérie.

5.6 Évaluation semi-directe par gold standard

Pour l'évaluation semi-directe à l'aide d'un gold standard, le fonctionnement est montré dans la figure 5.8. En vert, les paramètres qui jouent sur le résultat. Ils sont au nombre de trois : **GS**, le gold standard choisi ; **Mod**, le modèle de clustering choisi (parmi les trois dont nous disposons) ; et **QLT**, la mesure de la qualité du clustering, qui peut être Rand Index, F-mesure ou la pureté du clustering. Le *support* est l'ensemble de mots extraits du corpus qui figurent aussi dans le gold standard.

Le principe c'est de produire d'une part une catégorisation automatique, non supervisée, par un modèle de clustering **Mod**, puis de comparer pour les mots du

support la catégorisation du gold standard avec la catégorisation automatique, à l'aide de **QLT**. Le paramètre **Mod** est plus complexe que les autres, car il contient lui même les paramètres du modèle (k pour Kmeans, taux d'apprentissage pour SOM, par exemple), que nous détaillerons dans la sous-section suivante.

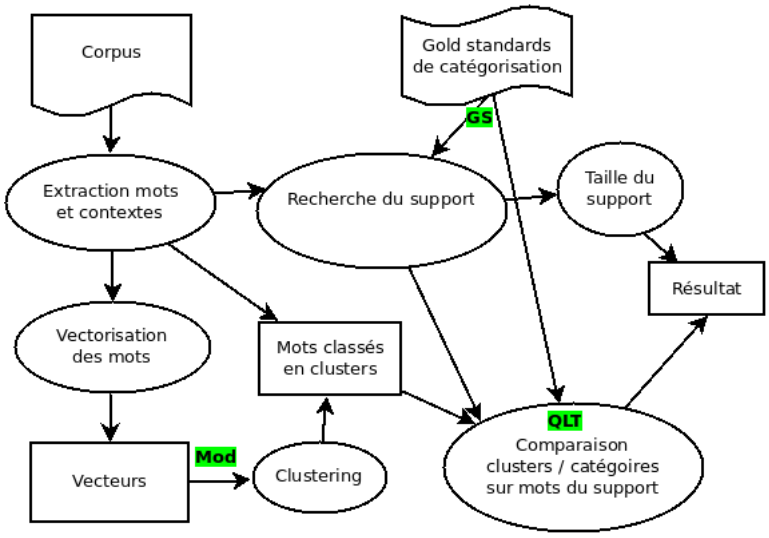

FIG. 5.8: Évaluation par catégorisation et gold standard

Ce type d'évaluation n'est disponible que pour l'anglais, avec 3 gold standards. On verra à la section 5.8.5 que nous avons ajouté un gold standard de catégorisation généré à partir de WordNet, pour les trois langues.

On dispose donc de 42 résultats différents pour l'anglais, et de 9 pour les autres langues, auxquels s'ajoutent les résultats produits par les variations des paramètres instrinsèques aux modèles.

5.6.1 Choix et paramètres des modèles de clustering

Comme indiqué, nous utilisons trois algorithmes :

- SOM, que nous avons réimplémenté avec une version optimisée (`https://gitlab.com/Data-Liasd/SOM`) ;
- Kmeans++, pour lequel nous utilisons la version implémentée dans Scikit-Learn (voir page 111).
- EM (Expectation-Maximization algorithm), pour lequel nous utilisons également l'implémentation de Scikit-Learn.

Nous présentons ici la liste des paramètres ajustables dans notre système, ainsi que les valeurs par défaut. Pour leur explication, voir ici le chapitre 3. L'interface présentée dans la figure 5.3 montre un exemple de paramètres qui peuvent être modifiés.

- **Kmeans :** k, le nombre de clusters, et *state_random*, le nombre de tirages aléatoires pour choisir les centres initiaux. La méthode d'initialisation des centres initiaux c'est celle de $Kmeans{+}{+}$ et le nombre maximum d'itérations est de 300 par défaut.
- **EM :** $n\_components$, le nombre de clusters ; les autres paramètres sont pris avec les valeurs par défaut de Scikit-Learn.
- **SOM :** Le choix entre *sparse* et *dense* pour le type de vecteur de mot, la taille de la carte X_{som} et Y_{som} ($X_{som} \times Y_{som}$ correspond au nombre de clusters), le nombre maximum d'itérations T_{max} par défaut est égale à 10 fois le nombre de données, la distance (euclidienne ou cosinus), la topologie de la carte (hexagonale ou carrée). L'initialisation des vecteurs mémoire peut se faire soit au centre de données soit aléatoirement. Le rayon initial de voisinage *Rayon_init* est initialisé par défaut avec la formule : $Rayon\_init = max(min(X_{som}, Y_{som})/2, 1)$. Les paramètres d'apprentissage sont par défaut : $\alpha_0 = 0.5$, $\sigma_i = 0.4$ et $\sigma_f = 0.1$.
- Les vecteurs d'entrée peuvent être normalisés ou non dans tous les modèles ; la valeur par défaut est sans normalisation.
- Le nombre de clusters est choisi partout en fonction du nombre de catégories du gold standard.

5.6.2 Qualité du clustering

Nous avons vu dans le chapitre 3 (section 3.3.1, page 65) que les méthodes du type *Bag of clusters* utilisent l'inertie inter et intra clusters pour évaluer leurs résultats. On cherche à baisser l'inertie intra-classes (les mots dans un cluster sont similaires) et d'augmenter l'inertie inter-classe (les mots de différents clusters sont dissemblables). L'inertie est un critère interne pour la qualité d'un clustering, elle prend en considération les vecteurs mais pas leurs étiquettes (les mots). Une bonne qualité interne ne se traduit pas forcément par une bonne qualité de clustering.

Pour évaluer la cohérence sémantique des clusters résultants, nous disposons de quatre mesures, à partir du gold standard choisi : la pureté du clustering, la F-mesure, l'Indice de Rand et l'Information Mutuelle Normalisée. Nous avons implémenté les trois premières.

Nous repartons pour définir ces mesures de qualité des conventions et définitions données page 117 et 5.3.4.2. L^{gs} désigne l'ensemble des labels du gold standard, et L^{aut} l'ensemble des clusters. Ici il n'existe pas de correspondance connue de L^{aut} sur L^{gs}.

La pureté de clustering (**PC**) est une mesure d'évaluation simple et transparente. Chaque cluster L_i^{aut} est assigné à la classe qui y est majoritaire, L_j^{gs}; soit N_i le nombre des mots de L_j^{gs} dans L_i^{aut}. On fait la même chose sur tous les clusters. On somme les N_i et on divise par la taille du support. La figure 5.9 illustre le calcul de la pureté dans un clustering à trois clusters. Le cluster 1 y est dominé par x (5 membres), le cluster 2 par o (4 membres) et le cluster 3 par $\diamond$ (3 membres) ; la pureté est de $(5 + 4 + 3)/17 \approx 0.71$..

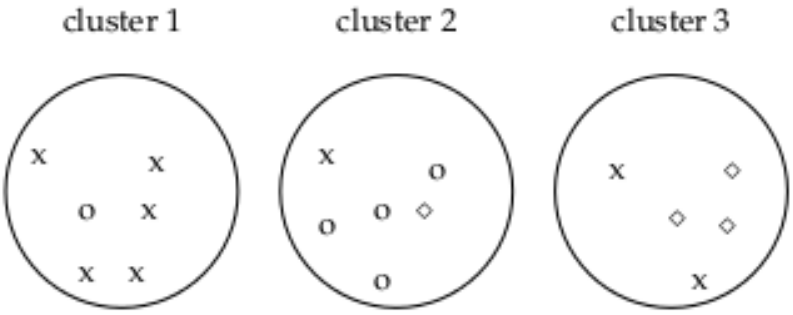

Fig. 5.9: Extrait de (Manning, Raghavan et Schütze, 2008)

PC varie entre 0 et 1; plus la valeur est élevée, meilleure est la classification. Elle est calculée par l'équation 5.16.

$$PC(L^{aut}, L^{gs}) = \frac{\sum_i \max_j \mid L_i^{aut} \cap L_j^{gs} \mid}{\mid support \mid} \tag{5.16}$$

L'inconvénient de cette mesure, c'est que lorsque le nombre de clusters est grand, une valeur élevée ne correspond pas forcément à une bonne classification. Par exemple s'il y a un mot par cluster, PC vaut 1. Une mesure qui nous permet de dépasser ce problème est l'information mutuelle normalisée, définie par l'équation 5.17.

$$NMI(L^{aut}, L^{gs}) = \frac{I(L^{aut}, L^{gs})}{[H(L^{aut}) + H(L^{gs})]/2} \tag{5.17}$$

I est l'information mutuelle :

$$I(L^{aut}, L^{gs}) = \sum_i \sum_j P(L_i^{aut} \cap L_j^{gs}) \log \frac{P(L_i^{aut} \cap L_j^{gs})}{P(L_i^{aut}) \cdot P(L_j^{gs})} \tag{5.18}$$

Où $P(X)$ est la probabilité qu'un mot appartienne à l'ensemble X.

H est l'entropie, elle se définit comme :

$$H(L^{aut}) = -\sum_i P(L_i^{aut}) \log P(L_i^{aut}) \tag{5.19}$$

Avec l'estimation du maximum de vraisemblance, les équations 5.18 et 5.19 deviennent :

$$I(L^{aut}, L^{gs}) = \sum_i \sum_j \frac{\mid L_i^{aut} \cap L_j^{gs} \mid}{\mid support \mid} \times \log \frac{\mid support \mid \times \mid L_i^{aut} \cap L_j^{gs} \mid}{\mid L_i^{aut} \mid \times \mid L_j^{gs} \mid} \tag{5.20}$$

$$H(L^{aut} = -\sum_i \frac{\mid L_i^{aut} \mid}{N} \log \frac{\mid L_i^{aut} \mid}{\mid support \mid} \tag{5.21}$$

Quand le nombre de clusters égale le nombre de mots, $H(L^{aut})$ atteint son maximum $\log(\mid support \mid)$, ce qui fait baisser la valeur de NMI. La valeur de NMI est entre 0 et 1, et plus elle est élevée meilleure est la classification. Dans l'exemple de la figure 5.9, $NMI \approx 0,36$.

L'indice de Rand a été présenté page 119, équation 5.11. Pour l'exemple de la figure 5.9: $TP = 20$, $TN = 72$, $FN = 24$ et $FP = 20$, donc $RI = 0,68$.

L'indice de Rand donne le même poids aux faux négatifs et aux faux positifs. Cependant, séparer deux mots similaires est parfois pire que de mettre des mots dissemblables dans le même cluster. F-mesure est une alternative pour surmonter ce problème ; on se reportera à sa définition page 118, équation 5.10. Dans l'exemple de la figure 5.9: $F_1 \approx 0.48$ et $F_5 \approx 0.45$.

5.7 Évaluation interne par substitution

Dans cette section, nous présentons une technique originale d'évaluation des représentations vectorielles de mots. Cette évaluation est interne car elle ne nécessite aucune donnée de référence issue de jugements humains. L'idée principale est de vérifier si deux mots ayant à peu près la même distribution vont avoir des vecteurs proches.

Pour cela, on prend un ensemble de mots tirés aléatoirement parmi les mots les plus fréquents du corpus. La moitié des occurrences de ces mots seront remplacées par des occurrences d'un autre mot inexistant dans le vocabulaire. La fréquence est importante parce que c'est la seule manière de s'assurer que la distribution du mot d'origine et du mot ajouté peuvent rester similaires. La figure 5.10 illustre le fonctionnement de cette méthode.

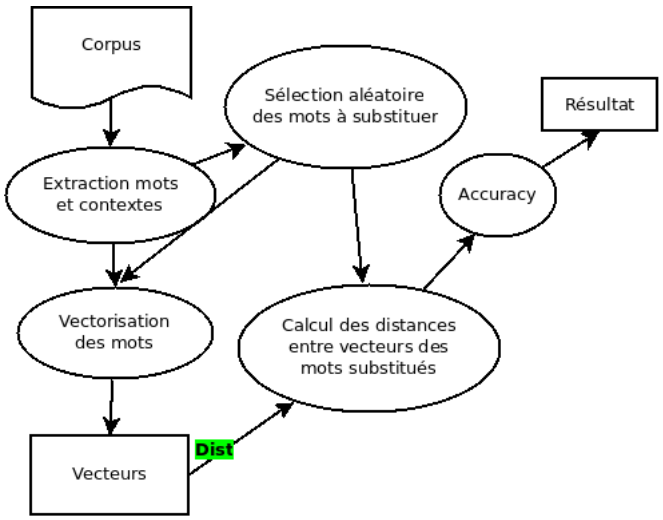

FIG. 5.10: Évaluation par substitution

Puis nous appliquons la méthode de représentation vectorielle sur le corpus modifié, et testons la similitude entre les mots d'origine et les mots modifiés. Pour simplifier, le mot modifié est simplement produit par redoublement du mot d'origine, ce qui permet de s'assurer facilement qu'il n'existait que marginalement dans le corpus.

Par exemple, si le mot «aime» apparaît 2000 fois dans le corpus, on va remplacer aléatoirement, avec une probabilité de 0,5, les occurrences de ce mot par le mot «aimeaime». Le nouveau corpus contiendra donc à peu près 1000 occurrences pour chacun de ces mots.

Voici l'algorithme détaillé :

1. prendre $Corp$, V et Min définis selon l'équation 5.1, page 115;
2. **Max :** définir un seuil maximal pour éviter d'avoir des mots outils pour lesquels le test risque d'être trivial.
3. **Mots de test :** choisir aléatoirement l'ensemble de test T selon l'équation

5.22 (pour s'assurer qu'une fois le nombre d'occurrences divisé par deux de m ni de mm ne passent au dessous de Min) ;

4. **Substitution :** pour chaque occurrence $o(m) \in Corp$ d'un mot $m \in T$, tirer aléatoirement un nombre réel a entre 0 et 1, si $a < 0.5$ on remplace le mot m par le mot mm dans $Corp$. Sinon on passe.

5. **Représentation :** appliquer la méthode de vectorisation sur le nouveau corpus ;

6. **Évaluation :** calculer le taux d'exactitude (on est dans le cas d'une seule classe, voir page 118, équation 5.7) ; concrètement cela donne les équations 5.23 et 5.24.

$$T = \{m \in V | Min * 2.5 < \#(m) < Max\} \tag{5.22}$$

$$Accuracy = \frac{\sum_{i=1}^{|T|} q(m_i)}{|T|} \tag{5.23}$$

$$q(m_i) = \begin{cases} 1 & si \;\; Dist(\vec{m_i}, m\vec{m}_i) = \min\limits_{\vec{v} \in V^* - \{m_i\}} Dist(\vec{m_i}, \vec{v}) \\ \\ 0 & sinon \end{cases} \tag{5.24}$$

Où V^* est le nouveau vocabulaire du corpus ($|V^*| = |V| + |T|$).

5.8 Évaluations basées sur WordNet

Nous avons déjà présenté Princeton WordNet en tant que thésaurus, ainsi que WOLF et Arabic WordNet (voir ici, chapitre 4, section 4.4.4.1, page 91). Nous allons dans la première sous-section décrire sa structure puis les mesures existantes ou proposées qui permettent de générer des méthodes d'évaluation nouvelles. Les méthodes d'évaluation que nous proposons sont décrites dans les sous-sections suivantes.

5.8.1 Structure de WordNet

WordNet est structuré comme un graphe sémantique dont les nœuds sont les synsets, et les arcs sont les liens sémantiques entre les synsets. Les mots ne sont pas des noeuds, et un mot qui a plusieurs sens se retrouvera dans plusieurs synsets.

La relation principale entre les mots dans WordNet est la synonymie, les mots qui dénotent le même concept et sont interchangeables dans de nombreux contextes sont regroupés dans des ensembles non ordonnés (synsets). Chacun des synsets de WordNet est lié à d'autres synsets au moyen d'un petit nombre de relations conceptuelles parmi celles citées dans le tableau de la page 22. De plus, chaque synset contient une brève définition, et dans la plupart des cas, un ou plusieurs exemples illustrant son utilisation. Les mots avec plusieurs significations distinctes sont représentées dans autant de synsets distincts. La figure 5.11 montre un exemple simple d'une hiérarchie de synsets dans WordNet.

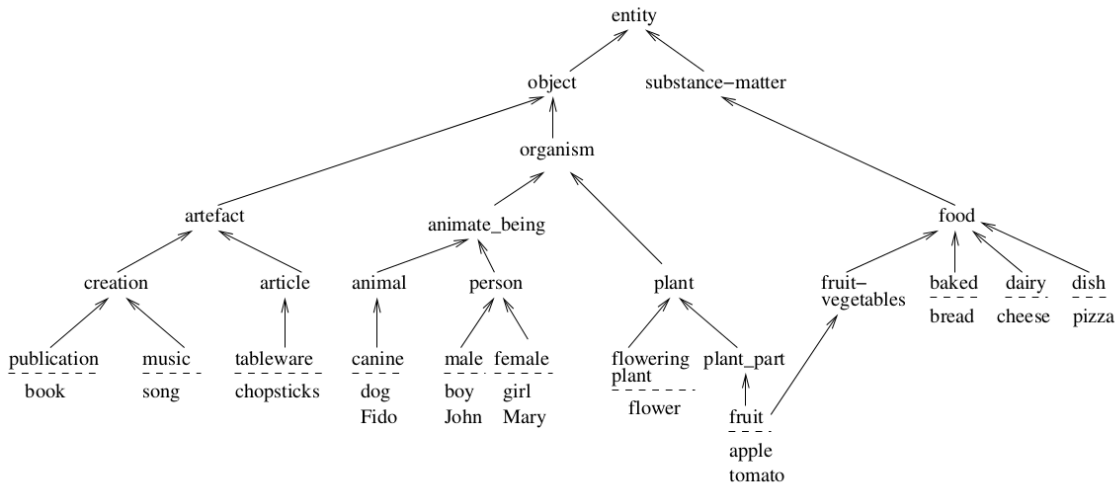

FIG. 5.11: Extrait de KAZAKOV et DOBNIK (2019) : hiérarchie de synsets

5.8.2 Interface de programmation pour WordNet

Depuis sa diffusion sur Internet, plusieurs outils ont été créés pour faciliter l'accès à la base de données de WordNet, à partir d'interfaces disponibles pour plusieurs langages de programmation (Java, Perl, PHP, Prolog, Python. etc).

Dans notre projet de thèse, pour intéroger WordNet, nous nous sommes basé sur la librairie NLTK (*Natural Language Toolkit*). NLTK a été développée au département d'informatique de l'université de Pennsylvanie par LOPER et BIRD (2002). C'est une suite de programmes Python open source qui couvre le traitement automatique de plusieurs langues naturelles et s'interface facilement avec les corpus annotés et les ontologies comme WordNet.

Nos trois WordNet de référence, Princeton WordNet (PWN) pour l'anglais, Arabic WordNet (AWN) pour l'arabe, et WOLF pour le français, sont tous distribués sous licence libre. De plus, ils sont disponibles pour NLTK.

PWN dans sa version (3.0 janvier 2007) répertorie plus de 117 597 synsets et 207 016 lemmes (Chaumartin, 2007). Arabic WordNet (AWN), dans sa version étendue (Abouenour, Bouzoubaa et Rosso, 2013), incluant plus d'entrées et plus de formes de pluriel irréguliers, contient 37 335 lemmes et WOLF contient 102 670 lemmes. Il s'agit ici des versions accessibles par NLTK, pas forcément des dernières versions.

5.8.3 Mesures de relation

Pedersen, Patwardhan et Michelizzi (2004) propose une implémentation en Perl de six mesures de similitude et trois mesures de parenté basées sur WordNet. Ces mesures prennent en entrée deux concepts (synsets) et leur associent une valeur numérique représentant le degré de similitude ou de relation qui les unit. Ces mesures sont aujourd'hui disponibles dans la librairie NLTK.

Parmi ces mesures, nous en avons retenu trois pour étude. Elles mesurent la similitude entre deux noeuds d'une taxonomie (un graphe sémantique), en se basant sur la profondeur des noeuds et la longueur des chemins entre eux. La figure 5.12 montre une taxonomie extraite de WordNet. La longueur d'un chemin entre noeuds est le nombre d'arrêtes qui joignent ces noeuds. La profondeur d'un noeud est la longueur du chemin entre la racine et le noeud plus 1.

Path_similarity. Cette mesure a été proposée par Rada et al. (1989). Elle retourne un score entre 0 et 1, en fonction du chemin le plus court entre les noeuds. L'équation 5.25 la définit pour WordNet.

$$path\_similarity(s_1, s_2) = \frac{1}{1 + chemin(s_1, s_2)} \tag{5.25}$$

Où $chemin(s_1, s_2)$ est la fonction qui donne le nombre d'arêtes du chemin le plus court entre s_1 et s_2.

Lch_similarity. Cette mesure a été proposée par Leacock, Miller et Chodorow (1998). Elle retourne des valeurs positives entre 0 et ∞, en fonction du chemin le plus court entre les noeuds. La valeur du chemin est ensuite normalisée en utilisant le double de la profondeur maximale D de la taxonomie. Elle est définie dans WordNet par l'équation 5.26.

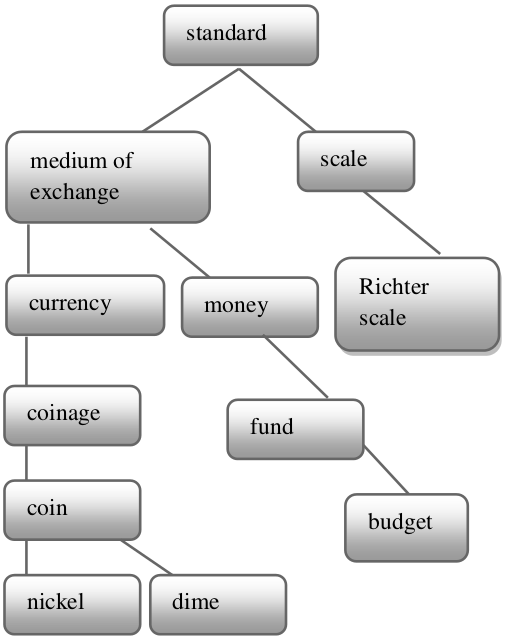

FIG. 5.12: Extrait de SLIMANI (2013) : Fragment d'une hiérarchie dans WordNet

$$lch\_similarity(s_1, s_2) = -\log(\frac{chemin(s_1, s_2)}{2 \times D}) \tag{5.26}$$

Wup_similarity. Cette méthode est définie par WU et PALMER (1994). Elle retourne une valeur entre 0 et 1, en fonction des profondeurs des deux noeuds et celle de leur dernier ancêtre commun. Elle est définie dans WordNet par l'équation 5.27.

$$wup\_similarity(s_1, s_2) = \frac{2 \times depth(lcs(s_1, s_2))}{depth(s_1) + depth(s_2)} \tag{5.27}$$

Où $lcs(s_1, s_2)$ est le dernier ancêtre commun entre s_1 et s_2 (anglais : *least common subsumer*), le dernier noeud à partir duquel divergent les branches de s_1 et s_2; $depth(s_i)$ est la profondeur de s_i.

La mesure *lch_similarity* est difficile à normaliser, en raison de la difficulté à déterminer la similitude maximale ; *lch_similarity* entre deux mots qui paratagent le même synset est égale à $+\infty$. Par conséquent elle s'avère pratiquement inutilisable pour notre propos.

En comparant les deux autres mesures, on constate que l'indice de similitude pour deux concepts frères est beaucoup plus élevé avec *wup_similarity*. En effet, *path_similarity* est trop lié au chemin (le chemin entre deux frères suppose de passer par le père, il est donc aussi long que le chemin entre un noeud et son oncle), ce qui nous a paru peu intuitif.

Pour illustrer les différences entre ces mesures, appliquons-les sur la figure 5.12, pour calculer le rapport entre les synsets désignés comme *nickel* et *dime*.

- $wup\_similarity(nickel, dime) = \frac{2\times4}{5+5} = 0.8$ ($lcs(nickel, dime) = coin$);
- $path\_similarity(nickel, dime) = \frac{1}{1+2} = 0.33$;
- $lch\_similarity(nickel, dime) = -\log(\frac{2}{2\times5}) = 0.69$.

5.8.3.1 Similitude entre les mots

Cependant, les mesures de similitude évoquées précédemment renvoient un degré de similitude entre synsets (entre concepts), et non entre mots. Il faut donc adapter ces mesures pour les utiliser dans notre contexte. Il y a peu de travaux sur la question. Notons cependant le programme WS4J[8] (*WordNet Similarity for Java*), pour lequel nous n'avons pas pu trouver de publication.

Notre proposition, pour calculer un degré de similitude entre deux mots m_1 et m_2, est de rechercher tous les synsets liés à chacun des deux mots, puis de calculer la mesure *wup_similarity*, choisie pour les raisons expliquées ci-dessus, entre tous ces synsets, enfin de sélectionner la valeur maximum. La procédure est exposée dans l'équation 5.28.

$$wup\_sim(m_1, m_2) = \max_{\substack{i\in synset(m_1),\\ j\in synset(m_2)}} wup\_similarity(i, j) \tag{5.28}$$

Où $synset(m_i)$ est une fonction qui retourne l'ensemble des synsets contenant le mot m_i.

5.8.4 Génération d'un gold standard attributionnel

L'idée ici c'est de générer automatiquement un gold standard attributionnel à partir de WordNet, **WNA**, qui permettra ensuite d'appliquer les évaluations définies dans la section 5.4.1, page 121, surtout pour les langues ayant peu ou pas de gold standard. La procédure est la suivante :

8. https ://code.google.com/archive/p/ws4j/

1. On divise la liste des mots du vocabulaire indexés dans WordNet en deux sous-listes A et B d'une manière aléatoire.

2. On tire aléatoirement un mot m_i de A et un mot m_j de B, pour construire la paire de mots (m_i, m_j).

3. On calcule la similitude entre les deux mots choisis avec $wup\_similarity$, et on obtient un triplet (m_i, m_j, s).

4. On recommence jusqu'à obtenir le nombre de paires souhaité.

Pour chaque langue, nous avons ainsi construit un dataset de 3000 paires de mots, qui représente par conséquent le plus grand des gold standards.

5.8.5 Génération d'un gold standard de catégorisation

Ce deuxième protocole d'évaluation est basé sur l'idée de transformer WordNet en un gold standard de catégorisation, **WNC**. Cette idée permet d'appliquer l'évaluation par catégorisation à beaucoup de langues (voir section 5.6, page 124 et suivantes), qui sinon n'est utilisable que pour l'anglais.

L'idée principale est d'utiliser les *synsets* comme catégories. Mais un gold standard de catégorisation doit avoir des catégories non vides et disjointes deux à deux. Or un mot peut avoir plusieurs sens, et donc se retrouver dans plusieurs synsets.

Pour contourner cette difficulté, nous avons choisi de n'utiliser que les mots du vocabulaire qui ont un seul sens dans WordNet. Le tableau 5.3 présente des exemples de catégories produites à partir de WordNet en anglais, en français et en arabe, après suppression des mots à plusieurs sens.

Langue	synsets
anglais	{idiot, idiots, cretin, imbeciles} {hark, harken, harkes, harking, harks, harkens, hearken } {wreath, coronal, wreaths, leis, chaplet} {swampy, marshy, boggy, waterlogged}
français	{diocèse, évêché, épiscopat, éparchie} {vignette, miniature, croquis } {mutuellement, réciproquement} {astronaute, cosmonaute, spationaute }
arabe	{اعتنق، تزوّج، صاهر} {أشهر، بث، علن} {تداول، تناقش، تباحث} {بهاء، أناقة، رشاقة}

TAB. 5.3: Synsets dont chaque mot a un sens unique dans le WordNet.

5.8.6 Évaluation semi-directe par wup

Cette méthode d'évaluation évalue une catégorisation non pas d'après un gold standard produit à partir de WordNet mais directement à partir de la similitude dans WordNet entre les synsets associés aux mots (ALIANE, MARIAGE et BERNARD, 2018b). La figure 5.13 montre le principe de fonctionnement de cette méthode. Le seul paramètre disponible ici est **Mod**, le modèle de clustering choisi (avec ses paramètres).

Comme pour la plupart des autres méthodes, on commence par rechercher *support* (mots du vocabulaire présents dans WordNet). Nous déterminons ensuite la liste des N clusters C_i qui contiennent des mots du support.

Puis nous calculons la cohérence de chaque C_i avec wup_sim (défini équation 5.28, page 134), sur $S = C_i \cap support$, avec l'équation 5.29.

$$coh(C_i) = \frac{\sum_{k=1}^{|S|-1} \sum_{j=k+1}^{|S|} wup\_sim(m_k, m_j)}{\mid S \mid .(\mid S \mid -1)/2} \tag{5.29}$$

Nous calculons enfin la moyenne de la cohérence des clusters, par l'équation 5.30.

$$Coh_{WN} = \frac{\sum_{i=1}^{K} coh(C_i)}{K} \tag{5.30}$$

Où K est le nombre de clusters.

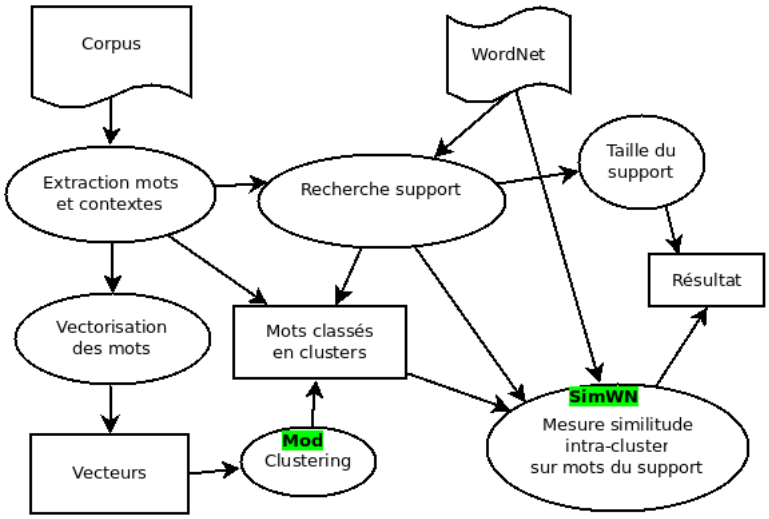

FIG. 5.13: Évaluation des catégories par wup

5.8.7 Évaluation semi-directe topologique

Dans cette dernière méthode d'évaluation, il s'agit non seulement d'évaluer la composition interne des clusters mais aussi d'évaluer la topologie produite. Cette méthode est proposée ici pour la première fois. Bien entendu, elle n'est compatible qu'avec un clustering topologique ; son principe c'est de comparer la topologie des clusters avec la topologie de WordNet. De la même façon que nous l'avons fait pour générer WNC (voir section 5.8.5, page 135) nous n'intègrerons dans le support que les mots n'appartenant qu'à un seul synset.

Son principe est fondé sur un calcul original de la matrice de confusion (voir section 5.3.4, page 117 et suivantes). La figure 5.14 illustre son fonctionnement.

Étant donné les mots m_i et m_j et un degré de similitude minimum θ, nous commençons par définir une relation de proximité dans WordNet par l'équation 5.31.

$$prox_{WN}(m_i, m_j) = \begin{cases} 1 & si \ \ wup\_sim(m_i, m_j) \geq \theta \\ \\ 0 & sinon \end{cases} \tag{5.31}$$

Avec une distance maximale δ sur la topologie des clusters, on définit également une proximité dans la carte topologique, en fonction des coordonnées sur cette carte des clusters C_i et C_j contenant respectivement les mots m_i et m_j, avec l'équation 5.32.

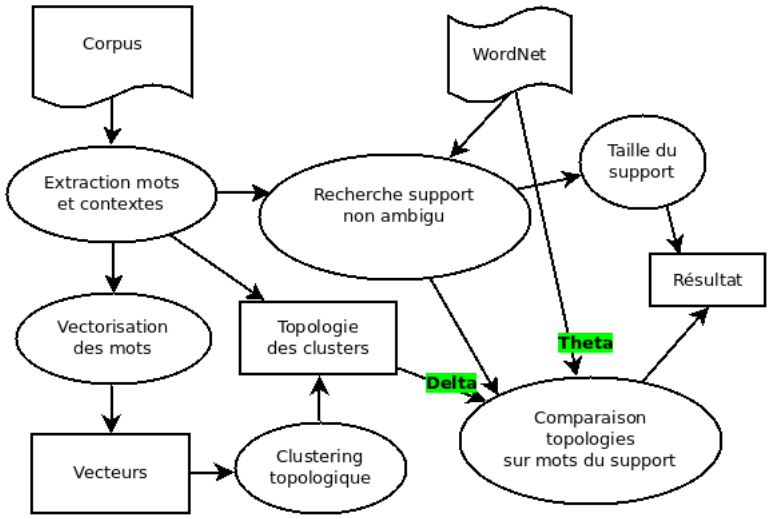

FIG. 5.14: Évaluation de la topologie

$$prox_{TC}(m_i, m_j) = \begin{cases} 1 & si\, dist_T(C_i, C_j) \leq \delta \\ \\ 0 & sinon \end{cases} \tag{5.32}$$

Où $dist_T$ est la distance dans la topologie des clusters.

Nous définissons la matrice de confusion comme suit :

- $TP = \#\{(m_i, m_j) \in V |\, prox_{WN}(m_i, m_j) \wedge prox_{TC}(m_i, m_j)\}$
- $TN = \#\{(m_i, m_j) \in V |\, \neg\, prox_{WN}(m_i, m_j) \wedge \neg\, prox_{TC}(m_i, m_j)\}$
- $FP = \#\{(m_i, m_j) \in V |\, prox_{WN}(m_i, m_j) \wedge \neg\, prox_{TC}(m_i, m_j)\}$
- $FN = \#\{(m_i, m_j) \in V |\, \neg\, prox_{WN}(m_i, m_j) \wedge prox_{TC}(m_i, m_j)\}$

Où $\#(m_i, m_j)$ est le nombre de paires de mots ($m_i \neq m_j$).

Sur la base de cette matrice de confusion, nous pouvons appliquer les différentes mesures d'évaluation standard (F-mesure, Rand index, etc.).

Pour le choix de la distance dans la topologie des clusters, on peut utiliser la distance euclidienne, la distance de Manhattan, la distance de Manhattan pondérée, la distance proposée par Bennani (ici section 3.4.3, page 75), entre autres.

Nous avons construit une distance qui donne la même valeur à tous les voisins de même rang quelle que soit la topologie du voisinage (Moore, en croix, Von

Neuman, en carré, ou hexagonale). En effet, la distance euclidienne ne remplit cette propriété qu'avec Moore ou hexagonale, la distance de Manhattan qu'avec Moore. Elle est calculée selon l'équation 5.33.

$$ds(C_i, C_j) = \max(|i_x - j_x|, |i_y - j_y|) \tag{5.33}$$

Où (i_x, i_y) et (j_x, j_y) sont les coordonnées respectives de C_i et C_j sur la carte topologique.

5.9 Conclusion

Nous récapitulons ici la liste des méthodes d'évaluation implémentées dans notre système, avec les abbréviations utilisées, qui nous serviront pour le chapitre suivant.

Les quatre méthodes suivantes sont exclusivement compatibles avec le clustering topologique :

1. internes
   1.1. **U-matrix**: visualisation topologique ;
   1.2. **QE**: erreur de quantification ;
   1.3. **TE**: erreur topologique ;
2. externe : **Topo**, l'évaluation topologique, qui se décline en plusieurs variantes suivant la distance choisie et les paramètres θ etδ.

Les trois méthodes externes suivantes évaluent le clustering en général ;

1. **RI**: Rand index ;
2. **PC**: pureté du clustering ;
3. **FM**: F-mesure ;

Le paramètre **Mod** (le modèle de clustering) prend trois valeurs possibles (SOM, Kmeans++ et EM) et **GS** en prend quatre en fonction du gold standard utilisé :

1. **AP**: RG-65;
2. **BM**: MC-30;

3. **BLESS**: WordSim-353;
4. **WNC**: WordNet de catégorisation.

Soit en tout 42 valeurs possibles, sans compter les paramètres propres aux modèles.

La méthode interne de l'inertie des données, abrégée en **ID**, peut s'appliquer sur le résultat du clustering ou directement sur les représentations vectorielles. Elle génère donc quatre valeurs. Cette méthode permet seulement d'évaluer la dispersion des données, et n'est compatible qu'avec les méthodes qui normalisent les vecteurs (sinon les espaces ne sont pas comparables).

On ajoutera dix tests fondés sur les gold standards attributionnels, qui sont multipliés par deux en fonction de la corrélation utilisée (Spearman ou Pearson) ; soit 20 valeurs possibles. Par défaut, nous n'indiquerons que la corrélation de Spearman, sauf quand la différence entre les deux est importante.

1. **RG**: RG-65;
2. **MC**: MC-30;
3. **WS**: WordSim-353;
4. **MEN**: MEN ;
5. **MTurk**: MTurk-771;
6. **RW**: Rare Word ;
7. **Verb**: Verb-143;
8. **SL**: SimLex-999;
9. **TOEFL**: TOEFL Synonyms ;
10. **WNA**: WordNet attributionnel.

On ajoutera aussi le gold standard relationnel Google Analogy, abrégé en **GA**.

Spécifiquement pour le français et l'arabe, pour compenser l'absence de gold standard, nous avons également introduit un test par sondage, abrégé en **TSF** et **TSA**.

Ajoutons encore la méthode interne aux représentations vectorielles que nous avons proposée, l'évaluation par substitution, abrégée en **Subst**.

Enfin pour terminer, il y a les deux valeurs du paramètre **Dist**.

Si nous faisons donc le total du nombre de scores d'évaluation que nous pouvons obtenir pour un modèle donné, nous arrivons à un total théorique de plusieurs centaines de mesures possibles. En réalité certaines combinaisons sont impossibles (certains scores ne sont pas disponibles pour toutes les langues ou tous les modèles).

On relèvera tout particulièrement, pour conclure, que parmi ces évaluations il y en a six que nous introduisons ici pour la première fois. En voici la liste :

- une méthode pour générer un gold standard attributionnel pour toute langue disposant d'un WordNet,
- une méthode pour générer un gold standard de catégorisation pour toute langue disposant d'un WordNet,
- une méthode de sondage auprès de participants humains testée sur deux langues,
- une méthode d'évaluation interne des représentations vectorielles par substitution,
- une méthode d'évaluation par catégorisation sans gold standard à partir de WordNet,
- une méthode d'évaluation de la topologie des clusters produite par une représentation vectorielle.

Chapitre 6

Expérimentations et résultats

Sommaire

6.1 Introduction

Dans ce chapitre nous présentons nos expérimentations et nos résultats.

EvalRep permet d'extraire et de comparer des centaines de résultats, en tout cas pour l'anglais ; certains de ces résultats, en particulier pour le clustering, dépendent de facteurs aléatoires et nécessitent plusieurs essais. La combinatoire est donc considérable, et nous ne l'avons pas explorée en totalité ; par exemple, nous avons réduit la part des méthodes internes au clustering. Même parmi celles que nous avons explorées, nous n'avons pas pu indiquer tous les résultats pour toutes les combinaisons.

Nous nous sommes efforcés d'avoir des résultats représentatifs dans suffisamment de combinaisons pour pouvoir en tirer des conclusions globales sur l'évaluation comparée et sur la validité des méthodes que nous avons introduites. Les méthodes de représentation vectorielle WebSOM, FastText et GraPaVec ont ainsi moins de résultats que CBOW, SkipGram et Glove, et nous servent surtout de témoins pour illustrer quelques points particuliers.

Dans la première section, nous présentons une étude de la corrélation entre gold standards et avec WordNet. Dans les sections suivantes, nous présentons pour chaque méthode de représentation vectorielle sélectionnée et pour chaque langue considérée les évaluations résultantes. Nous ne donnons, après la première section, que la corrélation de Spearman, conformément à l'usage dans la littérature, sauf quand les deux mesures divergent.

Puis nous présentons les résultats de l'évaluation comparée de ces méthodes de représentation vectorielle selon les différents types d'évaluation. Dans notre conclusion, nous chercherons à établir la validité des méthodes d'évaluation que nous avons introduite, et récapitulerons les éléments saillants de ces expérimentations.

6.2 Corrélations entre gold standards

Dans cette partie, nous donnons des éléments de réponse à la question que nous avons posée à la fin de l'état de l'art, en recherchant les corrélations entre les différents gold standards que nous avons cités. Nous commençons par étudier la corrélation entre les gold standards attributionnels, puis la corrélation entre eux et WordNet. Les corrélations utilisées sont celles de Spearman et de Pearson.

6.2.1 Comparaisons des gold standards attributionnels

Pour pouvoir étudier la corrélation, il faut des paires de mots à comparer. Par conséquent nous commençons par examiner la situation concernant les paires communes. Le tableau 6.1 donne le nombre de paires de mots en commun entre les différents gold standards attributionnels.

	WS353	MC30	RG65	MTurk771	SimLex999	MEN	RW	Verb143	TOEFL
WS353	353	29	29	0	9	6	0	0	0
MC30	29	30	30	0	1	1	0	0	0
RG65	29	30	65	0	1	5	0	0	1
MTurk771	0	0	0	771	15	14	0	0	0
SimLex999	9	1	1	15	999	45	1	1	2
MEN	6	1	5	14	45	3000	0	0	0
RW	0	0	0	0	1	0	2034	0	0
Verb143	0	0	0	0	1	0	0	143	17
TOEFL	0	0	1	0	2	0	0	17	400

TAB. 6.1: Paires de mots communes

On constate tout de suite que certains gold standards ne pourront être comparés faute de paires communes. Pour la suite, prenons un exemple.

Les 9 paires de mots communes à SimLex-999 et WordSim-353 sont données dans le tableau 6.2. Les notations ont été normalisées : chaque notation a été divisée par la norme euclidienne du vecteur des paires communes. Il reste à calculer les corrélations de Pearson et Spearman, qui donnent ici respectivement 0,53 et 0,36.

Paires de mots	SimLex-999	WordSim-353
science - psychology	0.492	0.671
day - dawn	0.547	0.753
clothes -closet	0.327	0.8
doctor - professor	0.465	0.662
student - professor	0.195	0.681
metal - aluminum	0.725	0.783
shore - coast	0.883	0.91
woman - man	0.333	0.83
moon - planet	0.587	0.808

TAB. 6.2: Comparaison entre SimLex et WordSim

Le tableau 6.3 représente les coefficients de corrélation de Pearson (dans la

moitié supérieure) et ceux de Spearman (en italique, dans la moitié inférieure) entre les gold standard attributionnels. Nous écartons du tableau les gold standards RareWord (RW), Verb-153 et TOEFL Synonyms, qui n'ont aucune intersection (ou réduite à une paire de mots, donc trop peu significative) avec les autres. De ces trois-là, TOEFL Synonyms et Verb-153 sont les seuls à avoir une intersection, que nous donnons dans le tableau 6.4. RareWord n'a aucune intersection significative (une paire en commun avec SimLex), ce qui n'est pas surprenant, s'agissant d'un jeu de données constitué de mots rares.

Datasets	WS353	MC30	RG65	MTurk771	SimLex999	MEN
WS353	1	0.94	0.91	-	0.53	0.79
MC30	*0.93*	1	0.96	-	-	-
RG65	*0.90*	*0.94*	1	-	-	0.90
MTurk771	-	-	-	1	0.82	0.76
SimLex999	*0.36*	-	-	*0.81*	1	0.43
MEN	*0.64*	-	*1*	*0.69*	*0.43*	1

Tab. 6.3: Corrélations entre les gold standards

	Pearson	Spearman
Verb143 vs TOEFL	0.89	0.84

Tab. 6.4: Corrélations Verb143 - TOEFL

Il ne serait pas très prudent de tirer des conclusions définitives de données aussi partielles ; on peut toutefois remarquer que la corrélation de Pearson donne un résultat supérieur à la corrélation de Spearman pour presque tous les calculs, sauf pour la paire MTurk - MEN, où c'est l'inverse (avec un support de 14 paires).

Ceci se produit quand il y a beaucoup de paires communes qui ont une valeur très proche avec une variation dans l'ordre des paires, puisque Spearman effectue son calcul sur les rangs des données. On relèvera également l'excellente corrélation entre SimLex et MTurk, et, à l'inverse, en dépit du fait qu'il s'agit du meilleur support (45 paires), la moins bonne corrélation entre SimLex et MEN, d'une part, et SimLex et WordSim, de l'autre. Pour ce dernier cas, étant donné la grande différence entre les protocoles et la manière d'interprêter la similitude, ce n'est pas surprenant.

6.2.2 Comparaison avec WordNet

Tous les mots dans WordNet se trouvent en relation sémantique avec tous les autres mots. Par conséquent il suffit de vérifier si les mots des gold standards

figurent dans WordNet. C'est le cas, à l'exception du mot «Maradona» qui figure dans WordSim353, ce qui écarte la paire «Maradona, football» de ce gold standard des comparaisons qui suivent, et du mot «halfheartedly», qui figure dans TOEFL Synonyms, ce qui écarte quatre paires (*halfheartedly + apathetically customarily bipartisanly unconventionally*).

Comme nous l'avons vu au chapitre précédent, il existe deux mesures de similitude définies sur WordNet, *wup* et *path*. Nous faisons ici l'étude des corrélations entre les notations des gold standards et ces deux distances. Ce serait redondant d'étudier les corrélations entre WNA et les autres gold standards, puisque WNA est fondé sur *wup*, l'une de ces deux distances.

Le tableau 6.5 représente le résultat de cette étude. La corrélation de Pearson est notée π, celle de Spearman σ.

	wup		path	
Datasets	π	σ	π	σ
RG65	0.79	0.76	0.77	0.75
MC30	0.78	0.74	0.75	0.72
Verb143	0.75	0.66	0.73	0.63
MTurk771	0.46	0.45	0.43	0.49
TOEFL	0.46	0.39	0.61	0.39
SimLex999	0.33	0.41	0.45	0.43
MEN	0.34	0.35	0.36	0.33
WS353	0.30	0.34	0.36	0.29
Rare-Word	-0.02	0.02	0.02	0.00

TAB. 6.5: Corrélation entre WordNet et les notations

On constate une très bonne corrélation avec les trois premiers gold standards (RG-65, Verb-143, MC30), quelles que soient la distance et la corrélation choisies.

Un deuxième groupe se dégage, avec MTurk et TOEFL, avec une corrélation moyenne, et un classement entre eux qui dépend de la distance (TOEFL meilleur avec *path* + Pearson, MTurk meilleur pour Spearman). Le fait que MTurk soit meilleur pour Spearman s'explique facilement en raison du caractère binaire de la notation dans TOEFL, qui diminue la corrélation sur les rangs des notes.

Si on tient compte de ce facteur, TOEFL reste ici le meilleur des deux pour *path*. Pour bien comprendre cet effet, il faut revenir sur les propriétés de *path* et *wup*. Les synonymes donnent 1 pour les deux distances, donc ce n'est pas de ce côté qu'il faut chercher l'explication, mais sur la corrélation avec les valeurs nulles. Or la propriété de la distance *path* c'est qu'elle ne tient pas compte de la profondeur dans l'arbre, et que, par conséquent, elle va avoir tendance à donner

un score moins élevé que *wup* pour des termes plus proches (par exemple, deux frères seront notés 0,33 avec *path* et 0,87 avec *wup*). En revanche, il y aura une note similaire pour des termes sans lien sémantique.

On peut en déduire que TOEFL introduit majoritairement dans les couples à valeur nulle des termes sémantiquement liés (par une autre relation que la synonymie) et non pas des termes très distants sémantiquement.

Le troisième groupe comprend SimLex, MEN et WordSim, avec des corrélations comprises entre $0,29$ et $0,45$, une faible variation suivant la distance choisie. Pour l'ensemble de ses valeurs, SimLex est le plus corrélé, sauf dans la combinaison *wup* avec Spearman, où c'est MEN qui domine.

RareWord forme un groupe à part avec une claire absence de corrélation. Ce phénomène très intéressant pose le problème de la fiabilité des notations. On peut y voir un effet de la sélection (internautes sélectionnés sur leur propre affirmation qu'ils connaissaient ces mots rares) ou un effet du fait que les relations sémantiques sont plus difficiles à établir pour des mots rares, ce qui tendrait à montrer un net biais lié à la fréquence.

MTurk, avec la même méthode mais un test rigoureux pour sélectionner les internautes, obtient une corrélation moyenne (0,46) ; on note aussi que MTurk utilise des mots fréquents. Pour tirer davantage de conclusions, il faudrait en savoir davantage sur les protocoles de notation. Le simple fait de faire appel à des internautes, en tout cas, n'empêche pas Verb-143 d'obtenir une excellente corrélation.

Les résultats avec les deux distances sur WordNet sont assez proches, et nous avons voulu connaître la corrélation entre elles. Le tableau 6.6 confirme cette impression, au moins pour les gold standards considérés.

	wup vs path	
Datasets	π	σ
TOEFL	0.87	0.94
MC30	0.81	0.98
RG65	0.80	0.92
Rare-Word	0.88	0.83
WS353	0.73	0.90
Verb143	0.84	0.79
MTurk771	0.69	0.89
MEN	0.70	0.79
SimLex999	0.69	0.77

Tab. 6.6: Corrélation entre les deux distances de WordNet

On relève que SimLex et TOEFL sont à l'opposé ici, ce qui tendrait à montrer

que SimLex contient plus de relations sémantiques proches que de synonymes ou de relations sémantiques lointaines, et que pour TOEFL c'est l'inverse ; il y a 25% de synonymes dans TOEFL ; nous avons vu que sur les 75% restant, il y a beaucoup de relations sémantiques proches. Les résultats donnés ici tendraient à montrer que le total synonymes + relations sémantiques lointaines est tout de même supérieur à 50%. En revanche, pour SimLex, une étude approfondie des données serait nécessaire pour s'avancer davantage sur les biais possibles.

Le tableau 6.7 représente le résultat de ces mêmes calculs sur les gold standards adaptés au français. Il faut rappeler que Wolf est un *silver standard* et non un gold standard (voir 4.4.4.1). Ici, 29 paires de WordSim353 et 84 paires dans SimLex999 ont au moins un mot non indexé dans Wolf.

	wup		path		wup vs path	
Datasets	π	σ	π	σ	π	σ
MC30	0.78	0.74	0.75	0.72	0.81	0.98
RG65	0.66	0.67	0.73	0.69	0.76	0.92
WS353	0.31	0.34	0.40	0.31	0.71	0.90
SimLex999	0.21	0.33	0.46	0.35	0.71	0.85

TAB. 6.7: Corrélation entre Wolf et les adaptations au français

Les résultats pour WordSim sont comparables à ceux pour l'anglais, et ceux pour les autres sont plus médiocres ; cependant, le classement reste globalement le même.

Le tableau 6.8 représente les résultats sur les gold standards adaptés à l'arabe. Contrairement au cas du français, le wordnet utilisé par NLTK, Arabic WordNet (AWN), est un gold standard, mais beaucoup moins riche, pour l'instant, que Princeton WordNet. 330 paires de mots de WordSim ne sont pas indexées dans AWN. Aucune des paires de mots de RG-65 (et donc de MC30, qui est un sous-ensemble), n'est indexée dans AWN.

	wup		path		wup vs path	
Datasets	π	σ	π	σ	π	σ
WordSim353	0.19	0.20	0.27	0.34	0.76	0.72

TAB. 6.8: Corrélation entre AWN et les adaptations à l'arabe

On constate ici qu'il y a une nette baisse de qualité, y compris dans les corrélation *wup - path*; ici le petit nombre de paires qui a pu être pris en compte (23) est sans doute la cause de cette dégradation. Un autre facteur a probablement joué, aussi bien pour le français que pour l'arabe, c'est la méthode d'adaptation (ici, section 4.2.2, page 85).

6.3 Paramétrage global

6.3.1 Corpus

Conformément à notre objectif, nous avons sélectionné plusieurs corpus en trois langues différentes : anglais, français et arabe standard moderne. Le tableau 6.9 détaille les caractéristiques des corpus utilisés.

Corpus	**Nb mots**	**Mots uniques**	**Seuil**	**Vocabulaire**	**Support WordNet**
Wikipédia 2017 anglais	2 409 291 852	9 045 033	600	96 967	47 118
Wikipédia 2017 français	804 476 834	4 348 227	300	86 697	18 600
Grand Corpus en arabe	4 165 244 672	10 633 401	600	93 118	4 159
Wikipédia 2017 arabe	117 472 209	2 140 757	300	33 974	3 566
Corpus Standard en arabe	8 327 894	532 695	100	9 843	822

Tab. 6.9: Caractéristiques des corpus

Seuil et vocabulaire sont ceux qui ont été définis au chapitre précédent (page 115), de même pour support (page 115). Les corpus pour l'arabe autres que Wikipedia ont été créés par Lebboss (2016). Ils dérivent d'un très grand corpus de sept milliards et demi de mots (qui a malheureusement été perdu) dont le Grand Corpus représente un peu plus de la moitié et le Corpus Standard (le nom est celui donné par l'auteur) 1,5%.

6.3.2 Corpus réduit

Comme indiqué dans la conclusion du chapitre 1, page 59, nous avons choisi WebSOM comme base de référence, permettant d'évaluer les progrès réalisés depuis. Mais l'établissement de la matrice d'origine, avant d'appliquer le random mapping, déborde largement la mémoire du système sur tous les corpus sauf le Corpus Standard arabe.

Nous aurions pu utiliser, comme Glove, des techniques plus modernes ; nous avons commencé par utiliser un corpus réduit pour effectuer nos tests et la qualité des résultats obtenus nous a indiqué que cela n'en valait pas la peine. Le corpus réduit est constitué à partir des 30.000 premières lignes de Wikipedia (anglais, français, arabe). Le seuil pour déterminer le vocabulaire a été fixé à 100.

Ces corpus réduits nous permettent d'étudier en même temps l'influence de la quantité de données sur les performances des autres méthodes, et nous ont été utiles également pour tester FastText.

Les caractéristiques des corpus réduits sont représentées dans le tableau 6.10.

Corpus	Réduction	Nb mots	Mots uniques	Vocabulaire	Support WordNet
Anglais	3%	81 197 934	773 871	34 297	27 463
Français	6%	52 858 341	712 520	29 349	11 383
Arabe	19%	22 993 330	760 545	23 401	1 463

TAB. 6.10: Caractéristiques des corpus réduits

6.3.3 Support des gold standards

Nous avons vu au chapitre précédent que toutes les méthodes qui utilisent un gold standard ont besoin de connaître son *support* (section 5.3, page 115). Les tableaux suivants indiquent le nombre de supports pour chacun des corpus utilisés, sur une ligne, et le pourcentage que ce nombre constitue par rapport au gold standard complet, sur la ligne suivante. Cette information constitue un indice de la qualité de l'évaluation.

Le tableau 6.11 présente les supports pour l'anglais ; on relèvera surtout le peu de fiabilité de Rare Word (support de 44% sur le corpus complet et de 27,3% sur le corpus réduit) ; la fiabilité la plus faible est ensuite celle du gold standard BM (support de 61,7% sur le corpus réduit). Le tableau 6.12 présente la situation pour le français et le tableau 6.13 celle de l'arabe. Pour ce dernier, on relèvera qu'avec le corpus réduit, la fiabilité de tous les gold standards adaptés descend drastiquement quand le corpus diminue (le support représente à peine plus de 50% du gold standard), effet qu'on n'observe pas dans les autres langues. Cela signifie en tout cas que les paires des gold standards adaptés à l'arabe n'ont pas du tout la même distribution que les paires originales.

Corpus	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL	GA	BM	AP	BLESS	WNA	WNC
Wikip.	MC	65	352	2999	771	907	118	995	296	18830	4073	393	200	2997	1874
%	100	100	100	100	100	44.6	82.5	99.6	92.5	96.3	76.5	97.7	100	99.9	89.2
Réduit	29	58	344	2667	756	555	96	967	228	15281	3283	321	175	2996	671
%	96.6	89.2	97.7	88.9	98.1	27.3	67.1	96.8	71.3	78.1	61.7	79.9	87.5	99.8	31.9

TAB. 6.11: Corpus anglais : support des gold standards

Corpus	MC	RG	WS	SL	WNA	WNC
Wikip.	30	65	346	956	3000	458
%	100	100	98.2	95.5	100	100
Réduit	25	56	313	715	1996	458
%	83.3	86.2	88.9	71.5	66.5	100

TAB. 6.12: Corpus français : support des gold standards

Corp	MC	RG	WS	WNA	WNC
Grand corpus	29	63	318	2998	275
%	96.6	96.9	90.3	99.9	67.6
Wikip.	28	61	300	2998	220
%	93.3	96.8	85.2	99.9	54.1
Wikip réduit	16	36	228	777	58
%	53.3	55.4	64.8	25.9	14.3
Corpus standard	10	23	154	423	29
%	33.3	35.3	43.7	14.1	7.1

TAB. 6.13: Corpus arabe : support des gold standards

Nous avons donné le nombre de classes des gold standards de catégorisation ; nous les redonnons ici, dans le tableau 6.14 avec le nombre des classes de WNC suivant les corpus (rappelons que les classes de WNC sont des synsets de WordNet, de petite taille, et non pas des catégories larges comme les autres).

Le Corpus standard arabe a été écarté des évaluations semi directes présentées (sauf avec wup), en raison du fait que son support (29 mots) et ses classes (26 classes) sont en nombre pratiquement identique, ce qui conduit à des résultats très peu généralisables.

Corp	BM	AP	BLESS	WNC
Wikip. ang	56	21	27	466
Wikip. ang réduit	56	21	27	321
Wikip. français	-	-	-	219
Wikip. français réduit	-	-	-	219
Grand corpus arabe	-			104
Wikip. arabe	-	-	-	99
Wikip. arabe réduit	-	-	-	31
Corpus standard arabe	-	-	-	26

TAB. 6.14: Nombre de classes par gold standard

6.3.4 Lecture des évaluations

Les évaluations pour chaque méthode sont présentées en deux parties : les évaluations directes (gold standards ou sondage) et les évaluation semi-directes par catégorisation.

Dans l'évaluation directe, les tableaux contiennent les résultats par langue. Ils incluent l'évaluation directe par gold standard attributionnels, avec la corrélation de Spearman. Quand Pearson diverge de manière significative de Spearman, les deux corrélations sont données, Spearman à gauche et Pearson à droite.

Pour TOEFL, un deuxième résultat est indiqué entre parenthèses, c'est celui de la méthode par notation (voir chapitre précédent, page 121). Pour le paramètre **Dist**, nous indiquons les valeurs obtenues avec le cosinus par défaut. Les tableaux de WebWOM sont parmi les rares où la distance euclidienne ne donne pas que des résultats inférieurs, la section WebSOM est la seule où les deux seront indiqués.

Ces tableaux incluent également le gold standard relationnel **GA** avec l'exactitude (*accuracy*) (voir la section 5.4.2, chapitre 5 page 122), et les sondages quand il y a lieu, **TSF** ou **TSA**, à droite dans les tableaux, avec sa note (voir la section 5.5, page 123).

L'ensemble des abbréviations des noms de gold standard et de paramètres est donné dans la section 5.9, page 139 et suivantes.

Les évaluations semi-directes sont de trois sortes :

- D'abord l'évaluation par gold standard. Pour ces tableaux on trouve les trois modèles de clustering, chacun avec ses meilleurs résultats pour RI (Rand index), PC (Pureté du clustering) et FM (F-mesure) avec les gold standard utilisés. Les détails sur les modèles et leur paramétrage précis sont donnés dans la section 5.6.1.

- Ensuite l'évaluation par *wup_sim*, qui mesure la cohérence des clusters rapportée à WordNet ; elle s'applique avec chacun des trois modèles de clustering.
- Enfin, l'évaluation topologique, qui mesure la cohérence topologiue des clusters et ne s'applique qu'avec SOM.

L'évaluation topologique étant la seule de son espèce et étant fondée sur un principe profondément différent des autres, nous ne l'avons pas présentée par méthode de représentation et lui avons consacré une section à part. Les évaluations des sections suivantes sont des récapitulations pour dégager une évaluation globale à partir de l'ensemble des résultats présentés dans les sections sur les méthodes.

6.3.5 Choix des meilleurs paramètres

Pour chaque méthode, nous effectuons cinq itérations avec chaque jeu de valeurs pour les paramètres. Nous sélectionnons le meilleur jeu par une procédure de vote, en comptant le nombre de scores améliorés, par catégories (les gold standards sont considérés comme une seule catégorie de même poids que les évaluations par clustering, l'évaluation par wordnet comme une autre). Puis nous testons un autre jeu de valeurs, et comptons combien de scores sont améliorés, jusqu'à déterminer le jeu donnant le meilleur résultat.

Il y a une exception à cette règle concernant la dimension des vecteurs : pour les grands corpus (Wikipedia complet dans les trois langues et le Grand corpus arabe), nous avons dû utiliser, pour les trois modèles concernés, la dimension 200 (essentiellement en raison du temps pris par SkipGram).

En règle générale on constate que l'ensemble des évaluations évolue dans la même direction. Il est difficile ici de donner une procédure générale rigoureuse avec une pondération par type d'évaluation, car la situation dans les différentes langues est très diverse.

6.4 Websom

Nous avons testé les paramètres suivants, pour lesquels nous indiquons en gras les valeurs donnant le meilleur résultat, quelle que soit la langue donnée. Sauf indication explicite du contraire, les expérimentations présentées suivent toutes ce paramétrage.

- **Dimension réduite**[1]**:** $\{50, \mathbf{90}, 120, 130, 150\}$; il est intéressant que la meilleure valeur corresponde à celle choisie par les auteurs de WebSOM, pourtant sur un corpus complètement différent ;
- **Facteur d'influence du vecteur du milieu :** $\eta \in \{0.1\ , \mathbf{0.2}, 0.3, 0.4, 0.5\}$;
- **Fenêtre :** {**3** 5 7}; là aussi il s'agit de la valeur choisie par les auteurs de WebSOM.

Seuls les corpus réduits et le Corpus Standard en arabe sont utilisables avec cette méthode.

6.4.1 Évaluation directe

Cette méthode de vectorisation étant celle qui présente le plus de cas où le paramètre **Dist** donne quelques meilleurs résultats avec la distance euclidienne, nous présentons à cette occasion des tableaux avec les deux valeurs. L'explication de ce phénomène tient sans doute au fait que les vecteurs qu'elle produit (par Random Mapping) sont quasi-normalisés.

Le tableau 6.15 montre donne les résultats pour l'anglais. Dans l'ensemble, le seul test que WebSOM réussit c'est le TOEFL (avec la méthode par notation). On relèvera un net écart entre Spearman et Pearson dans les trois premiers gold standards.

Dist	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL	WNA	GA
cosinus	0.36|0.17	0.27|0.09	0.31|0.18	0.19	0.14	0.05	0.05	0.10	0.41|0.16 (0.64)	0.15	0.02
euclide	0.39|0.26	0.31|0.17	0.32|0.25	0.21	0.15	0.05	0.05	0.09	0.41|0.19 (0.61)	0.15	0.01

TAB. 6.15: WebSOM, éval. directe, anglais

Le tableau 6.16 indique les résultats pour le français. Là aussi ils sont mauvais.

Dist	MC	RG	WS	SL	WNA	TSF
cosinus	0.11	0.27	0.06	0.12	0.11	0.16
euclide	0.16	0.21	0.07	0.14	0.13	-

TAB. 6.16: WebSOM, éval. directe, français

Le tableau 6.17 indique les résultats pour l'arabe. On constate de bons résultats voire très bons résultats, largement au-dessus de ceux des autres langues, dans MC

1. Rappelons que le vecteur final est obtenu par concaténation de 3 vecteurs à dimension réduite.

et RG ; mais ce sont deux tous petits gold standards batis sur les mêmes paires de mots, et le support est de 16 pour le premier et 36 pour le deuxième. Nous avons constaté, dans les tests sur le Corpus Standard (non inclus dans le tableau pour ne pas encombrer les données), que la corrélation avec ces gold standards était encore meilleure, allant jusqu'à 0.92 pour MC avec cosinus (0.9 avec euclide ; un peu discordant avec Pearson à 0.71 dans ce cas) ; le reste étant similaire.

Dist	MC	RG	WS	WNA	TSA
cosinus	0.78	0.66	0.06	0.19	0.26
euclide	0.78	0.62	0.09	0.2	-

Tab. 6.17: WebSOM, éval. directe, arabe

Dans l'ensemble, même ici, cosinus et distance euclidienne donnent des valeurs très proches, ce qui ajoute à la justification de ne publier que les résultats obtenus avec cosinus pour la suite.

6.4.2 Évaluation semi-directe

6.4.2.1 Par gold standards

Le tableau 6.18 indique les résultats obtenus avec les gold standards de catégorisation sur le corpus anglais réduit, avec les évaluations PC, RI et FM. On constate que les trois modèles, quel que soit le gold standard (et donc y compris le nôtre, WNC), donnent un résultat similaire : un bon ou très bon Rand Index, une pureté de clustering médiocre et une F-mesure catastrophique. Une telle discordance entre les trois mesures n'est pas apriori indicateur d'une bonne qualité de clustering. En regardant de plus près, il y a très peu de TP et de FN et les FP sont très inférieurs aux TN, d'où un très bon RI, et une précision très faible (avec un rappel moyen), d'où une F-mesure basse.

	SOM				Kmeans++				EM			
	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.27	0.32	0.36	0.52	0.29	0.33	0.40	0.53	0.36	0.29	0.38	0.45
RI	0.93	0.87	0.84	0.99	0.94	0.88	0.86	0.98	0.93	0.87	0.86	0.96
FM	0.15	0.16	0.18	0.10	0.15	0.17	0.19	0.03	0.06	0.16	0.19	0.07

Tab. 6.18: WebSOM + catégorisation, anglais réduit

Le tableau 6.19 indique les mêmes résultats sur le corpus français réduit, avec l'unique gold standard disponible (celui que nous avons construit à partir de WordNet), WNC. De même pour le tableau 6.20 avec le corpus arabe réduit. Pour le

corpus standard arabe, le plus petit, les résultats sont non pertinents étant donné que le support est très petit (29 mots).

Dans les deux langues nous constatons le même classement, avec une pureté de clustering meilleure mais une F-mesure pratiquement nulle, et la même convergence entre les trois modèles.

Les mêmes constats s'imposent pour ce modèle par conséquent.

WNC	SOM	Kmeans++	EM
PC	0.50	0.53	0.54
RI	0.99	0.98	0.99
FM	0.06	0.04	0.05

TAB. 6.19: WebSOM + catégorisation, français réduit

	SOM	Kmeans++	EM
PC	0.56	0.59	0.56
RI	0.96	0.94	0.84
FM	0.03	0.02	0.01

TAB. 6.20: WebSOM + catégorisation, arabe réduit

6.4.2.2 Par wup

Le tableau 6.21 présente les résultats de wup (qui évalue la cohérence des clusters relativement à WordNet). Les trois modèles présentent une évaluation comparable par rapport aux différents corpus. La cohérence est moyenne pour l'anglais, puis baisse, sans doute en rapport avec la richesse moins grande des WordNet français et arabe.

	SOM	Kmeans++	EM
Anglais réduit	0.51	0.53	0.50
Français réduit	0.46	0.47	0.45
Arabe réduit	0.39	0.38	0.40
Corpus standard arabe	0.35	0.37	0.36

TAB. 6.21: Websom, cohérence WordNet avec wup

6.5 CBOW

6.5.1 Paramètres

Voilà les paramètres testés avec en gras les valeurs donnant les meilleurs résultats. Rappelons qu'avec les grands corpus, nous avons utilisé 200 pour dimension des vecteurs.

- **Fonction d'activation :** {**Negative_sampling** , Hierarchical_softmax}
- **Nombre d'itérations :** $\{5, 10, \mathbf{15}\}$;
- **Dimension vecteur :** $\{50, 100, 200, 250, \mathbf{300}\}$;
- **Fenêtre :** $\{\mathbf{4}, 6, 8, 10, 12\}$

6.5.2 Évaluation directe

Le tableau 6.22 indique les résultats obtenus pour l'anglais. Le point notable ici est la contreperformance de CBOW par rapport à SimLex et à WNA (les gold standards les plus riches en paires de mots), et sa performance très moyenne par rapport à Verb et à RareWord. Pour le reste, les résultats sont bons.

Corp	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL	WNA	GA
Wiki complet	0.77	0.79	0.68	0.74	0.64	0.48	0.38	0.34	0.62 (0.79)	0.04	0.71
Wiki réduit	0.83	0.77	0.71	0.74	0.66	0.49	0.49	0.35	0.66 (0.84)	0.07	0.69

Tab. 6.22: CBOW, éval. directe, anglais

Le tableau 6.23 indique les résultats obtenus pour le français. Ils sont globalement semblables à ceux de l'anglais, en un peu moins bons ; le résultat du sondage est moyen. SimLex francophone et WNA ont les plus mauvais résultats, comme en anglais.

Corp	MC	RG	WS	SL	WNA	TSF
Wiki complet	0.71	0.68	0.53	0.26	0.11	-
Wiki réduit	0.69	0.65	0.52	0.26	0.12	0.62

Tab. 6.23: CBOW, éval. directe, français

Le tableau 6.24 indique les résultats obtenus pour l'arabe. Les deux corrélations divergent notablement (à gauche Spearman). Ces divergences ne concernent que les tous petits gold standards, sur lesquels Pearson est largement meilleur que Spearman. Il n'y avait pas de divergence dans ce sens là pour WebSOM. En dehors de ce fait, WNA a l'un des plus mauvais scores, TSA est faible et la meilleure corrélation concerne MC et RG, comme pour WebSOM.

Corp	MC	RG	WS	WNA	TSA
Grand corpus	0.55 | 0.84	0.59 | 0.85	0.38	0.24	-
Wiki complet	0.56 | 0.90	0.51 | 0.88	0.50	0.20	-
Wiki réduit	0.71 | 0.93	0.52 | 0.86	0.53	0.20	0.69
Corpus standard	0.46 | 0.95	0.52 | 0.92	0.53	0.16	-

TAB. 6.24: CBOW, éval. directe, arabe

6.5.3 Évaluation semi-directe

6.5.3.1 Par gold standards

Le tableau 6.25 indique les résultats obtenus avec les gold standards de catégorisation et le corpus réduit. Les résultats sont ici aussi cohérents entre les différents modèles et les différents gold standards, du point de vue du classement des évaluations : Rand Index > pureté du clustering > F-mesure. WNC et BLESS ont la meilleure valeur pour la pureté, et BLESS la meilleure F-mesure, suivi par WNC.

	SOM				Kmeans++				EM			
Éval.	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.53	0.67	0.79	0.72	0.52	0.60	0.67	0.77	0.52	0.61	0.60	0.73
RI	0.96	0.94	0.95	0.99	0.96	0.92	0.92	0.99	0.97	0.93	0.90	0.98
FM	0.39	0.48	0.61	0.49	0.31	0.35	0.46	0.42	0.39	0.40	0.34	0.42

TAB. 6.25: CBOW + catégorisation, anglais réduit

Le tableau 6.26 indique les catégorisations obtenues avec le corpus complet. Les résultats sont comparables aux précédents, quasi identiques pour RI, avec pour les autres mesures une légère baisse pour SOM et pour WNC, une stabilité ou une légèrement augmentation dans les autres cas. On peut donc estimer que la taille du corpus a peu influencé les résultats. BLESS a cette fois la meilleure pureté, suivi par WNC, et garde la tête pour la F-mesure.

	SOM				Kmeans++				EM			
	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.54	0.58	0.72	0.63	0.53	0.60	0.77	0.64	0.52	0.60	0.62	0.60
RI	0.96	0.90	0.94	0.99	0.96	0.90	0.95	0.99	0.96	0.92	0.91	0.99
FM	0.37	0.33	0.59	0.41	0.34	0.33	0.61	0.31	0.31	0.37	0.45	0.32

TAB. 6.26: CBOW + catégorisation, anglais complet

Les tableaux 6.27 et 6.28 indiquent les résultats sur le français avec WNC. Il y a toujours les mêmes constatations, mais le corpus réduit baisse nettement en

F-mesure et en pureté, contrairement à la situation pour l'anglais. WNC garde à peu près la même valeur en pureté que pour l'anglais.

	SOM	Kmeans++	EM
PC	0.65	0.67	0.69
RI	0.99	0.99	0.99
FM	0.31	0.23	0.25

Tab. 6.27: CBOW + catégorisation, français complet

	SOM	Kmeans++	EM
PC	0.48	0.53	0.54
RI	0.98	0.98	0.99
FM	0.11	0.04	0.05

Tab. 6.28: CBOW + catégorisation, français réduit

Les tableaux 6.29 (pour le grand corpus), 6.30 (pour le corpus wikipedia complet) et 6.31 (pour le corpus wikipedia réduit) indiquent les résultats sur l'arabe avec WNC. Le corpus standard a été écarté pour les raisons indiquées section 6.3.3, page 153. Il y a les mêmes constatations que pour l'anglais. On notera des résultats inférieurs en F-mesure aux autres corpus, peut-être liées à la génération de WNC (qui ne donne pas forcément le même type de résultat d'un WordNet à un autre), voir section 5.8.5, page 135.

	SOM	Kmeans++	EM
PC	0.49	0.55	0.52
RI	0.97	0.97	0.96
FM	0.08	0.07	0.05

Tab. 6.29: CBOW + catégorisation, grand corpus arabe

	SOM	Kmeans++	EM
PC	0.53	0.53	0.54
RI	0.98	0.98	0.99
FM	0.07	0.04	0.05

Tab. 6.30: CBOW + catégorisation, arabe complet

	SOM	Kmeans++	EM
PC	0.55	0.62	0.61
RI	0.96	0.91	0.85
FM	0.06	0.02	0.01

Tab. 6.31: CBOW + catégorisation, arabe réduit

6.5.3.2 Par wup

Le tableau 6.32 présente les résultats de wup. On constate des résultats assez bons pour l'anglais, qui baissent sur le français, puis davantage sur l'arabe, toujours en parallèle avec la richesse des WordNet respectifs. La comparaison des quatre corpus de taille différente en arabe semble montrer que la taille du corpus en revanche n'est pas un facteur pertinent. Les trois modèles donnent toujours des résultats semblables.

	SOM	Kmeans	EM
Anglais complet	0.67	0.68	0.68
Anglais réduit	0.66	0.66	0.67
Français complet	0.55	0.57	0.57
Français réduit	0.56	0.54	0.52
Grand corpus arabe	0.46	0.46	0.45
Arabe complet	0.47	0.45	0.44
Arabe réduit	0.48	0.45	0.43
Corpus standard arabe	0.43	0.44	0.45

TAB. 6.32: CBOW, cohérence WordNet avec wup

6.6 SkipGram

6.6.1 Paramètres

Voici les paramètres testés avec en gras les valeurs retenues.

- **Fonction d'activation :** {**Negative_sampling** , Hierarchical_softmax}
- **Nombre d'itérations :** $\{5, 10, \mathbf{15}\}$;
- **Dimension vecteur :** $\{50, 100, 200, 250, \mathbf{300}\}$;
- **Fenêtre :** $\{\mathbf{4}, 6, 8, 10, 12\}$

6.6.2 Évaluation directe

Le tableau 6.33 indique les résultats obtenus pour l'anglais. Ici aussi on relève une contre-performance avec SimLex et WNA ; globalement les résultats présentent les mêmes caractéristiques que pour CBOW.

Corp	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL	WNA	GA
Wiki complet	0.78	0.75	0.67	0.73	0.64	0.49	0.44	0.31	0.61 (0.82)	0.02	0.69
Wiki réduit	0.80	0.69	0.69	0.73	0.65	0.45	0.55	0.33	0.64 (0.87)	0.05	0.68

TAB. 6.33: SkipGram, éval. directe, anglais

Le tableau 6.34 indique les résultats obtenus pour le français. Nous avons ici aussi le même profil que pour l'anglais, avec la contre-performance sur SimLex.

Le tableau 6.35 indique les résultats obtenus pour l'arabe. On observe les mêmes divergences que pour CBOW avec les petits corpus. WordSim donne des

Corp	MC	RG	WS	SL	WNA
Wiki complet	0.65	0.64	0.53	0.23	0.15
Wiki réduit	0.71	0.66	0.53	0.21	0.08

TAB. 6.34: SkipGram, éval. directe, français

résultats de moyens à mauvais ; paradoxalement, plus le corpus est petit, plus les résultats sont bons.

Corp	MC	RG	WS	WNA	TSA
Grand corpus	0.56 | 0.88	0.57 | 0.82	0.35	0.21	-
Wiki complet	0.37 | 0.92	0.43 | 0.92	0.46	0.15	-
Wiki réduit	0.51 | 0.93	0.49 | 0.89	0.54	0.16	0.70
Corpus standard	0.68 | 0.95	0.56 | 0.92	0.50	0.18	-

TAB. 6.35: SkipGram, éval. directe, arabe

6.6.3 Évaluation semi-directe

6.6.3.1 Par gold standards

Le tableau 6.36 indique les catégorisations obtenues entre les gold standards et le corpus complet et le tableau 6.37 celles du corpus réduit. On retrouve à peu près les mêmes résultats qu'avec CBOW, y compris sur l'absence d'impact de la taille du corpus, et le fait que WNC et BLESS ont la meilleure pureté. Pour la F-mesure c'est BLESS qui est en tête sans conteste.

	SOM				Kmeans++				EM			
	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.53	0.60	0.69	0.67	0.48	0.54	0.66	0.71	0.51	0.51	0.66	0.67
RI	0.96	0.91	0.91	0.99	0.96	0.90	0.92	0.99	0.96	0.91	0.92	0.99
FM	0.40	0.41	0.50	0.46	0.31	0.35	0.44	0.40	0.35	0.31	0.42	0.38

TAB. 6.36: SkipGram + catégorisation, anglais complet

Les tableaux 6.38 et 6.39 indiquent les résultats sur le français avec WNC. La pureté reste similaire à ce qu'elle est pour l'anglais. Pour la F-mesure, les résultats sont contrastés entre les modèles de clustering plus qu'entre le corpus réduit et le corpus plein. C'est SOM qui donne le meilleur résultat.

Les tableaux 6.40 (pour le grand corpus), 6.41 (pour le corpus wikipedia complet) et 6.42 (pour le corpus wikipedia réduit) indiquent les résultats sur l'arabe

	SOM				Kmeans++				EM			
	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.54	0.59	0.77	0.73	0.53	0.55	0.71	0.73	0.55	0.50	0.70	0.71
RI	0.96	0.92	0.95	0.99	0.96	0.92	0.96	0.99	0.96	0.90	0.93	0.97
FM	0.44	0.36	0.62	0.48	0.31	0.31	0.40	0.24	0.40	0.28	0.54	0.30

TAB. 6.37: SkipGram + catégorisation, anglais réduit

	SOM	Kmeans++	EM
PC	0.65	0.67	0.66
RI	0.99	0.99	0.98
FM	0.32	0.13	0.09

TAB. 6.38: SkipGram + catégorisation, français complet

	SOM	Kmeans++	EM
PC	0.69	0.73	0.74
RI	0.99	0.99	0.99
FM	0.31	0.18	0.19

TAB. 6.39: SkipGram + catégorisation, français réduit

avec WNC. La faible F-mesure se confirme, en revanche la pureté reste à peu près semblable à celle pour les autres langues.

	SOM	Kmeans++	EM
PC	0.52	0.56	0.54
RI	0.98	0.99	0.99
FM	0.09	0.08	0.10

TAB. 6.40: SkipGram + catégorisation, grand corpus arabe

	SOM	Kmeans++	EM
PC	0.56	0.60	0.61
RI	0.99	0.99	0.98
FM	0.09	0.07	0.06

TAB. 6.41: SkipGram + catégorisation, arabe complet

6.6.3.2 Par wup

Le tableau 6.43 présente les résultats de wup. On relève toujours l'absence de corrélation avec la taille du corpus, et une corrélation avec la richesse de WordNet.

	SOM	Kmeans++	EM
PC	0.58	0.62	0.63
RI	0.98	0.95	0.87
FM	0.09	0.04	0.02

Tab. 6.42: SkipGram + catégorisation, arabe réduit

	SOM	Kmeans	EM
anglais complet	0.69	0.70	0.68
anglais réduit	0.69	0.67	0.68
français complet	0.53	0.55	0.54
français réduit	0.50	0.50	0.52
grand corpus arabe	0.40	0.41	0.38
arabe complet	0.42	0.41	0.42
arabe réduit	0.40	0.42	0.41
standard arabe	0.37	0.39	0.39

Tab. 6.43: SkipGram, cohérence WordNet avec wup

6.7 Glove

6.7.1 Paramètres

Voici les paramètres testés avec en gras les valeurs retenues.

- **Nombre d'itérations :** $\{5, 10, \mathbf{15}\}$;
- **Dimension vecteur :** $\{50, 100, 200, 250, \mathbf{300}\}$;
- **Fenêtre :** $\{4, 6, 8, \mathbf{10}, 12\}$

6.7.2 Évaluation directe

Le tableau 6.44 indique les résultats obtenus pour l'anglais. Ici aussi on relève la même contre-performance avec SimLex et WNA ; la situation est très comparable à celle de CBOW et SkipGram.

Le tableau 6.45 indique les résultats obtenus pour le français, qui eux aussi sont comparables à ceux de CBOW et SkipGram.

Le tableau 6.46 indique les résultats obtenus pour l'arabe. Ils sont également similaires à ceux de CBOW et SkipGram, avec la divergence Spearman / Pearson pour les deux premiers gold standards. Le résultat au sondage est bon.

Corp	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL	WNA	GA
Wiki complet	0.72	0.75	0.67	0.73	0.66	0.46	0.51	0.34	0.63 (0.82)	0.04	0.65
Wiki réduit	0.68	0.68	0.65	0.70	0.64	0.46	0.42	0.33	0.64 (0.85)	0.06	0.64

TAB. 6.44: GloVe, éval. directe, anglais

Corp	MC	RG	WS	SL	WNA	TSF
Wiki complet	0.69	0.70	0.54	0.23	0.11	-
Wiki réduit	0.66	0.61	0.53	0.19	0.10	0.53

TAB. 6.45: GloVe, éval. directe, français

Corp	MC	RG	WS	WNA	TSA
Grand corpus	0.67 | 0.94	0.59 | 0.89	0.35	0.24	-
Wiki complet	0.66 | 0.92	0.61 | 0.92	0.23	0.1	-
Wiki réduit	0.71 | 0.95	0.55 | 0.90	0.53	0.12	0.66
Corpus standard	0.77 | 0.96	0.66 | 0.94	0.43	0.1	-

TAB. 6.46: GloVe, éval. directe, arabe

6.7.3 Évaluation semi-directe

6.7.3.1 Par gold standards

Le tableau 6.47 indique les catégorisations obtenues entre les gold standards et le corpus complet et le tableau 6.48 ceux du corpus réduit. On retrouve les constatations déjà faites sur la pureté avec BLESS et WNC en tête, et sur la F-mesure avec BLESS en tête. Ils confirment également que SOM obtient de meilleures performances.

	SOM				Kmeans++				EM			
	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.52	0.62	0.82	0.66	0.51	0.55	0.80	0.67	0.54	0.52	0.74	0.67
RI	0.96	0.93	0.95	0.99	0.96	0.92	0.95	0.99	0.97	0.91	0.94	0.99
FM	0.36	0.39	0.65	0.43	0.29	0.31	0.61	0.25	0.53	0.33	0.57	0.36

Tab. 6.47: GloVe + catégorisation, anglais complet

	SOM				Kmeans++				EM			
	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC	BM	AP	BLESS	WNC
PC	0.43	0.53	0.64	0.66	0.43	0.49	0.60	0.68	0.44	0.50	0.54	0.65
RI	0.95	0.87	0.90	0.99	0.95	0.91	0.90	0.97	0.95	0.91	0.90	0.93
FM	0.29	0.23	0.45	0.31	0.27	0.24	0.36	0.07	0.22	0.25	0.33	0.09

Tab. 6.48: GloVe + catégorisation, anglais réduit

Les tableaux 6.49 et 6.50 indiquent les résultats sur le français avec WNC. Il y a toujours les mêmes constatations, mais on note qu'en plus le corpus réduit baisse nettement en F-mesure et en pureté, contrairement à la situation pour l'anglais. SOM donne les meilleures valeurs en F-mesure et en pureté (pour la pureté seule ce n'est pas toujours le cas, mais même quand il n'est pas le premier, il n'est pas loin derrière).

	SOM	Kmeans++	EM
PC	0.64	0.67	0.66
RI	0.99	0.99	0.98
FM	0.25	0.15	0.13

Tab. 6.49: GloVe + catégorisation, français complet

	SOM	Kmeans++	EM
PC	0.65	0.68	0.68
RI	0.99	0.99	0.99
FM	0.20	0.17	0.14

Tab. 6.50: GloVe + catégorisation, français réduit

Les tableaux 6.51 (pour le grand corpus), 6.52 (pour le corpus wikipedia complet) et 6.53 (pour le corpus wikipedia réduit) indiquent les résultats sur l'arabe

avec WNC. On observe toujours une F-mesure très basse, pratiquement nulle, et une pureté moyenne.

	SOM	Kmeans++	EM
PC	0.51	0.53	0.55
RI	0.98	0.98	0.99
FM	0.09	0.07	0.10

TAB. 6.51: GloVe + catégorisation, grand corpus arabe

	SOM	Kmeans++	EM
PC	0.55	0.57	0.57
RI	0.98	0.98	0.95
FM	0.09	0.06	0.04

TAB. 6.52: GloVe + catégorisation, arabe complet

WNC	SOM	Kmeans++	EM
PC	0.58	0.62	0.62
RI	0.98	0.91	0.86
FM	0.08	0.03	0.02

TAB. 6.53: GloVe + catégorisation, arabe réduit

6.7.3.2 Par wup

Le tableau 6.54 présente les résultats de wup. Les mêmes constats que pour les autres modèles peuvent être faits ici.

	SOM	Kmeans	EM
anglais complet	0.65	0.63	0.62
anglais réduit	0.62	0.61	0.64
français complet	0.54	0.53	0.53
français réduit	0.50	0.50	0.52
grand corpus arabe	0.43	0.45	0.40
arabe complet	0.44	0.44	0.41
arabe réduit	0.46	0.43	0.39
Corpus standard arabe	0.40	0.41	0.42

TAB. 6.54: GloVe, cohérence WordNet avec wup

6.8 FastText

6.8.1 Paramètres

Le principal intérêt de FastText, qui est une extension de SkipGram, est de donner une représentation vectorielle à des mots qui ne sont pas inclus dans le vocabulaire. Pour pouvoir tester cela, il nous fallait donc un gold standard dont le support n'était pas inclu dans le corpus. Ceci écarte tous les grands corpus, puisque plus de 99% du support de tous les gold standards y sont inclus. Par conséquent nous n'avons testé FastText que sur les corpus réduits. Ceci écarte également l'évaluation semi-directe par wup, puisqu'elle ne présente pas de mots qui soient absents du corpus d'entraînement.

Un autre facteur est intervenu, c'est la taille gigantesque prise par le modèle sur un corpus, et qui rendait difficile son application sur de grands corpus (en effet, tous les vecteurs des ngrammes comprenant entre trois et six caractères sont générés).

Les paramètres choisis sont ceux indiqués pour SkipGram.

6.8.2 Évaluation directe

Le tableau 6.55 indique les résultats obtenus pour l'anglais. Ils sont plutôt moins bons que les précédents sauf WebSOM, et la contre-performance atteint aussi Verb.

Corp	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL	WNA	GA
Wiki réduit	0.66	0.56	0.60	0.73	0.63	0.42	0.32	0.22	0.51 (0.74)	0.09	0.57

Tab. 6.55: FastText, éval. directe, anglais

Le tableau 6.56 indique les résultats obtenus pour le français.

Corp	MC	RG	WS	SL	WNA
Wiki réduit	0.67	0.58	0.45	0.23	0.07

Tab. 6.56: FastText, éval. directe, français

Le tableau 6.57 indique les résultats obtenus pour l'arabe. On constate à nouveau la discordance entre Spearman et Pearson pour les petits gold standards.

Les résultats sont assez cohérents dans toutes les langues pour indiquer que FastText obtient une corrélation moyenne avec la plupart des gold standards. Sa capacité à déduire des mots non appris ne lui permet pas de dépasser les autres méthodes que nous avons vues, ni même d'atteindre leur qualité (à part WebSOM),

Corp	MC	RG	WS	WNA
Wiki réduit	0.24 | 0.86	0.32 | 0.87	0.47	0.15
Corpus standard	0.19 | 0.78	0.28 | 0.75	0.48	0.18

TAB. 6.57: FastText, éval. directe, arabe

en contradiction avec les données fournies par les auteurs (voir section 2.3.4, page 57).

6.9 GraPaVec

6.9.1 Paramètres

Pour GraPaVec nous n'avons eu qu'un seul jeu de vecteurs, extraits du Corpus Standard, avec les paramètres :

- **Seuil des marques :** 3500;
- **JokerLength :** 4;
- **Seuil des patterns :** 300.

6.9.2 Évaluation directe

Le tableau 6.58 indique les résultats obtenus pour l'arabe. On y constate encore la discordance Spearman | Pearson que nous avons déjà trouvée avec la plupart des autres méthodes (sauf WebSOM). Il y a très peu de corrélation avec WordSim arabisé, toutefois plus qu'avec WebSOM.

Corp	MC	RG	WS	WNA
Corpus standard	0.43 | 0.84	0.63 | 0.80	0.13	0.15

Tab. 6.58: GraPaVec, éval. directe, arabe

6.9.3 Évaluation semi-directe

6.9.3.1 Par gold standards

Étant donné que nous n'avons GraPaVec que pour l'arabe, corpus standard, il n'y a que WNC qui puisse être utilisé comme gold standard. Par ailleurs, nous n'utilisons ici que SOM, dont on a vu qu'il était l'un des meilleurs modèles pour le clustering, parce que les vecteurs de GraPaVec sont très creux et qu'il aurait fallu les densifier au préalable pour une utilisation optimale de Kmeans. Voir le tableau 6.59 pour les résultats, dont on constate qu'ils sont en dessous des autres, même de WebSOM (0,56 de F-mesure).

PC	RI	FM
0.46	0.70	0.01

Tab. 6.59: GraPavec, catégorisation, corpus standard arabe

6.9.3.2 Par wup

Le tableau 6.60 présente les résultats de wup. La seule chose qu'on peut dire sur si peu de données c'est que les résultats sont cohérents avec ceux qu'on a déjà vu et semblables pour les trois modèles ; ils sont supérieurs à ceux de WebSOM.

	SOM	Kmeans++	EM
Corpus standard arabe	0.39	0.38	0.38

TAB. 6.60: GraPaVec, cohérence WordNet avec wup

6.10 Évaluation par catégorisation

6.10.1 Paramétrage du clustering

Les détails sur les paramètres et leurs valeurs par défaut ont été indiqués section 5.6.1, page 126.

Paramétrage de SOM. Nous avons pris les valeurs par défaut de l'algorithme telle qu'elles sont données par l'auteur, après en avoir testé d'autres. Par exemple, nous avons choisi la distance euclidienne après avoir constaté que le cosinus nous donnait de moins bons résultats (contrairement à la mesure de proximité des vecteurs de mots pour l'évaluation directe).

Paramétrage de Kmeans++. Le paramètre *random_state* est fixé à 5 (valeurs testées de 1 à 5). Les autres paramètres sont pris avec les valeurs par défaut.

Paramétrage de EM. Les valeurs prises sont les valeurs par défaut.

Pour tous les modèles, nous avons testé avec et sans normalisation des vecteurs d'entrée. Le meilleur résultat est obtenu sans normalisation.

6.10.2 Nombre de clusters

Pour obtenir des résultats comparables, comme les gold standards ont tous un nombre de classes différents et qu'aussi bien la pureté que la F-mesure sont sensibles au nombre de classes, il faut entraîner les algorithmes avec le nombre de clusters correspondant au nombre de classes.

Pour un gold standard donné, tous les algorithmes auront donc un nombre de clusters fixé en fonction du nombre de classes. Comme SOM a besoin de faire une carte en deux dimensions, on sélectionne le nombre le plus proche (au dessus) qui se décompose en deux facteurs. Si plusieurs décompositions sont possibles, on choisit celle qui se rapproche le plus d'un carré (pour la facilité de la visualisation). Puis on donne le même nombre de clusters aux autres algorithmes.

Ce nombre varie entre 21 et 466.

6.10.3 Observations

Concernant le pouvoir discriminant de nos trois mesures, Rand Index est peu probant car donnant toujours d'excellents résultats à toutes les méthodes avec les gold standards. Dans nos tests, les résultats varient entre 0.8 et 0.99, sauf une fois, pour Grapavec (0.70), mais sur une technique de représentation que nous n'avons guère pu tester et WNC.

Pour les autres mesures, on constate que la pureté est toujours supérieure à la F-mesure. Dans les corpus arabes, la F-mesure est en général très basse, mais nous n'avons qu'un seul gold standard, qui reste très expérimental.

Concernant les modèles, nous pouvons observer, sur les données qui ont été présentées (comme sur la plupart des autres tests effectués), que SOM est le modèle qui donne en général les meilleurs résultats en F-mesure et pureté. Kmeans++ vient en deuxième, et EM est systématiquement troisième. Entre SOM et Kmeans++, le choix peut être dicté par la simplicité de paramétrage et la rapidité (Kmeans++) ou par la qualité des résultats (SOM). EM est à la fois lent et peu performant.

Concernant les gold standards, il ressort très nettement de nos observations que BLESS et, dans une moindre mesure, WNC pour l'anglais, sont les meilleurs gold standards, qui maximisent la pureté et la F-mesure. Du point de vue de la taille, WNC est en anglais le deuxième gold standard, après BM.

WNC pour le français semble se comporter de manière similaire, même si c'est avec des valeurs inférieures.

Son support a une bonne représentativité pour le plus grand corpus en anglais et pour les deux corpus en français ; cette représentativité baisse pour l'anglais avec le corpus réduit à près de 32%, mais ça ne semble pas réellement affecter les résultats.

Pour l'arabe cette représentativité décroît régulièrement avec la taille ; le support du plus grand corpus n'a que 67,6% de représentativité. Et du point de vue de la performance, la F-mesure très basse aurait tendance à nous faire disqualifier WNC pour l'arabe.

Sans doute doit-on voir dans ces éléments en partie un effet de la moindre

couverture de Wolf par rapport à PWN et surtout de AWN par rapport à Wolf, qui serait partiellement compensée par la taille du corpus. Mais on relèvera le contraste de ce point de vue avec WNA, voir plus loin section 6.11.3.

Pour résumer l'ensemble des observations : on peut éliminer le Rand index et EM de nos résultats. Pour l'arabe, le gold standard dont nous disposons n'est pas probant. Pour l'anglais, le plus fiable est BLESS. Avec ces simplifications, nous pouvons présenter nos résultats dans deux tableaux très résumés (pour le reste, voir les tableaux dans chaque méthode), l'un pour WNC sur le français et l'anglais, l'autre pour BLESS sur l'anglais.

Dans le tableau 6.61, qui sous chaque méthode de représentation, il y a deux colonnes pour les évaluations par SOM et par Kmeans. Pour chaque corpus, on indique la pureté de clustering et la F-mesure. Et dans les cases, on indique le numéro d'ordre de la méthode pour la mesure donné par ce modèle. Par exemple, une case à l'intersection de CBOW et anglais réduit d'une part, de SOM et de PC d'autre part, contient le chiffre 2; cela signifie que CBOW, pour le corpus anglais réduit, a été classé deuxième par SOM pour la pureté de clustering.

		CBOW		SkipGram		Glove		WebSOM	
		SOM	Kmeans	SOM	Kmeans	SOM	Kmeans	SOM	Kmeans
Anglais complet	PC	3	2	1	1	2	3		
	FM	3	2	1	1	2	3		
Anglais réduit	PC	2	1	1	2	3	3	4	4
	FM	1	1	2	2	3	3	4	4
Français complet	PC	1	1	1	1	3	1		
	FM	2	1	1	3	3	2		
Français réduit	PC	2	2	1	1	2	3	4	4
	FM	2	1	1	2	3	3	4	4

TAB. 6.61: Classement des méthodes par WNC

Si on comptabilise le nombre de fois où une méthode est classée première, c'est SkipGram qui vient nettement en tête, suivi par CBOW et par Glove, WebSOM étant toujours dernier.

Le tableau 6.62 donne les résultats pour BLESS. Ici, avec la même méthode de calcul, c'est GloVe qui arrive en tête, suivi par CBOW et par SkipGram, WebSOM restant dernier.

		CBOW		SkipGram		Glove		WebSOM	
		SOM	Kmeans	SOM	Kmeans	SOM	Kmeans	SOM	Kmeans
Anglais complet	PC	2	2	3	3	1	1		
	FM	2	1	3	3	1	1		
Anglais réduit	PC	1	2	2	1	3	3	4	4
	FM	2	1	1	2	3	3	4	4

Tab. 6.62: Classement des méthodes par BLESS

6.10.4 Évaluation par wup

Concernant wup, nous avons pu constater que les trois modèles de clustering ont des résultats approchants. La taille du corpus ne semble pas pertinente, seule la richesse de WordNet semble jouer.

Le tableau 6.63 montre sur le corpus anglais réduit ce que ces résultats nous indiquent quant à la qualité des représentations vectorielles. En gras les meilleurs résultats et en italique les plus mauvais. Ils indiquent WebSOM comme étant la plus mauvaise représentation, SkipGram comme étant la meilleure, suivie de près par CBOW puis GloVe. Les résultats sont identiques sur le corpus complet.

	SOM	Kmeans	EM
CBOW	0.66	0.66	0.67
SkipGram	**0.69**	**0.67**	**0.68**
GloVe	0.62	0.61	0.64
WebSOM	*0.51*	*0.53*	*0.50*

Tab. 6.63: Wup sur l'anglais

En revanche, pour le français et l'arabe, la situation est légèrement différente, car CBOW y est systématiquement en tête. Le tableau 6.64 nous permet de comparer aussi GraPaVec, mais les résultats sur le français sont semblables. Comme dans les autres, WebSOM est bon dernier. GloVe est second, suivi pratiquement à égalité par GraPaVec et SkipGram.

6.10.5 Évaluation topologique

En gardant le même paramétrage pour SOM, nous avons déterminé les dimensions de la carte topologique à partir d'une estimation du nombre de mots dans chaque synset. Comme la moyenne de ce nombre se situe autour de 5 (si on ne tient pas compte des singletons), nous avons donc divisé le nombre de mots du support de WordNet (voir tableaux 6.9 et 6.10, pages 151 et 152) par 5 pour calculer la dimension de la carte.

	SOM	Kmeans	EM
CBOW	**0.43**	**0.44**	**0.45**
SkipGram	0.37	0.39	0.39
GloVe	0.40	0.41	0.42
WebSOM	*0.35*	*0.37*	*0.36*
GraPaVec	0.39	0.38	0.38

TAB. 6.64: Wup sur l'arabe - corpus standard

Nous explorons ici les propriétés de cette méthode d'évaluation. Prévenons tout de suite qu'il s'agit plus d'une exploration liminaire que d'une véritable évaluation, et que ce que nous indiquons ici constitue, faute de temps, plus une piste de réflexion qu'une réflexion aboutie.

En effet, si l'idée de départ de comparer deux topologies nous semble toujours valide, nos expérimentations nous ont fait prendre conscience de plusieurs obstacles à franchir, essentiellement deux.

Tout d'abord, une distance sur une arborescence présente des différences importantes avec une distance sur une carte à deux dimensions par exemple. On verra, dans la section suivante, que c'est probablement la nature particulière de ce type de distance qui a fait échouer notre projet de gold standard attributionnel.

Le deuxième obstacle tient à la plus ou moins grande pauvreté du WordNet considéré, qui rend délicate les comparaisons interlangues.

Rappelons que le paramètre θ ($0 < \theta < 1$) est la distance maximale prise en compte dans WordNet, mesurée avec *wup_sim*, et que le paramètre δ ($\delta \geq 1$) est la distance maximale prise en compte dans la topologie des clusters (la carte SOM), mesurée avec *ds* (notre distance, présentée page 139).

Nous testons tout d'abord les propriétés de θ et de δ. Les valeurs de δ pour 4 et au delà nous donnent une F-mesure de 0; nous avons surtout exploré les valeurs 1 et 2.

Nous utilisons la F-mesure et Rand Index (qui n'a pas du tout le même type de résultats qu'avec la catégorisation à l'aide de gold standards). Nous avons choisi le corpus anglais complet comme source pour nos premières expériences parce qu'il est celui qui a le support maximal, et donc les valeurs les plus fiables pour *wup_sim*.

6.10.5.1 Anglais, $\delta = 1$

Les figures 6.1 et 6.2 montrent respectivement l'évolution de la F-mesure et de Rand Index en fonction de θ pour le corpus anglais complet avec les trois méthodes

CBOW, SkipGram et Glove, avec $\delta = 1$.

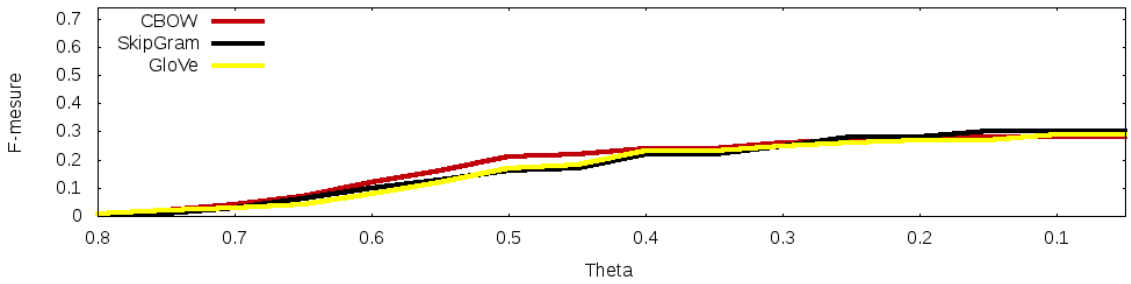

FIG. 6.1: Évolution de la F-mesure selon θ, anglais, $\delta = 1$

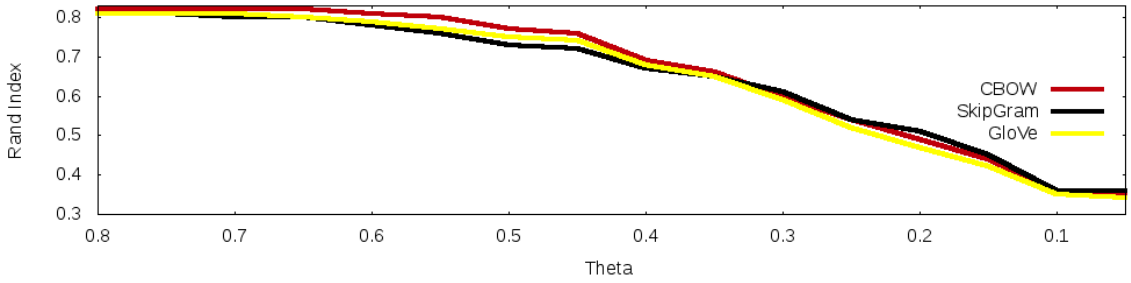

FIG. 6.2: Évolution du Rand Index selon θ, anglais, $\delta = 1$

On constate que les deux mesures évoluent régulièrement en fonction de θ. La figure 6.3 montre l'évolution croisée de ces mesures, toujours avec la même valeur de δ, et avec CBOW.

Les figures 6.4 et 6.5 montrent la même évolution pour SkipGram et pour GloVe.

On observe partout un point d'inflexion pour RI, avant son plateau au minimum de sa valeur ; pour F-mesure, il y a aussi un plateau, moins net, et qui commence plus tôt, autour de 0,25.

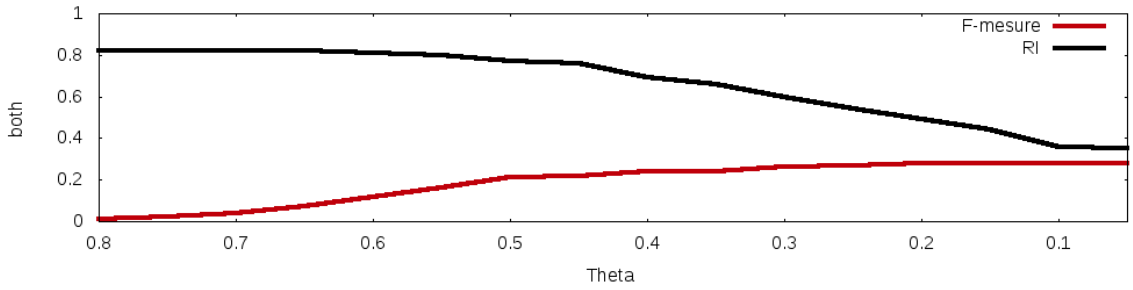

FIG. 6.3: FM et RI selon θ, CBOW, anglais, $\delta = 1$

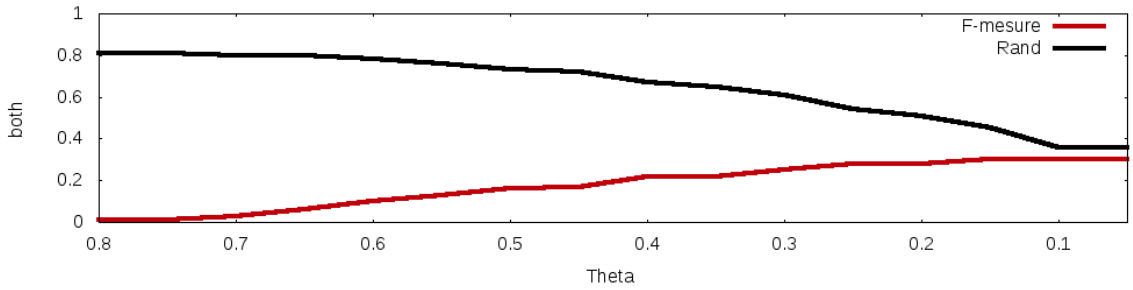

FIG. 6.4: FM et RI selon θ, SkipGram, anglais, $\delta = 1$

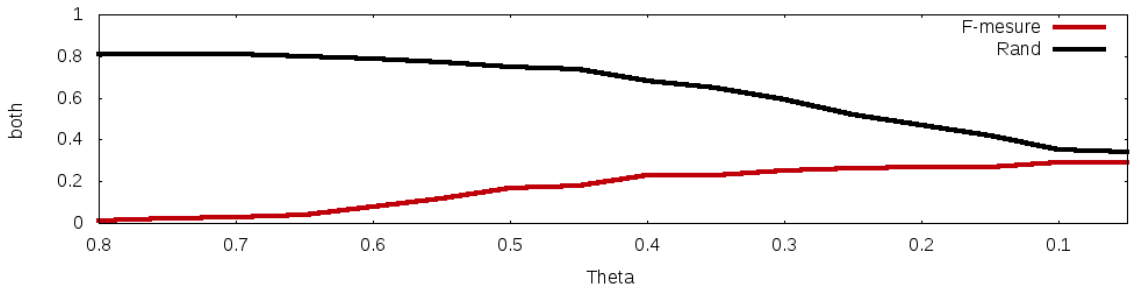

FIG. 6.5: FM et RI selon θ, GloVe, anglais, $\delta = 1$

6.10.5.2 Anglais, $\delta = 2$

Les figures 6.6 et 6.7 montrent la situation avec $\delta = 2$, qui entraîne une amélioration de la F-mesure, avec toujours un plateau autour de 0,1. Sur ce plateau, la F-mesure est pratiquement le double de ce qu'elle était avec $\delta = 1$

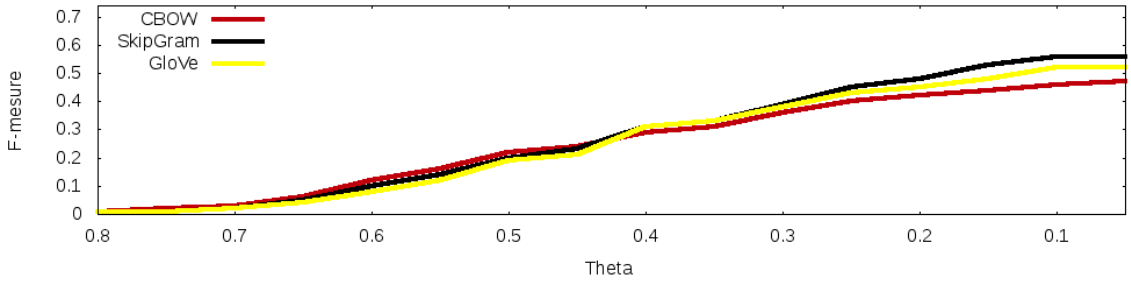

FIG. 6.6: Évolution de la F-mesure selon θ, anglais, $\delta = 2$

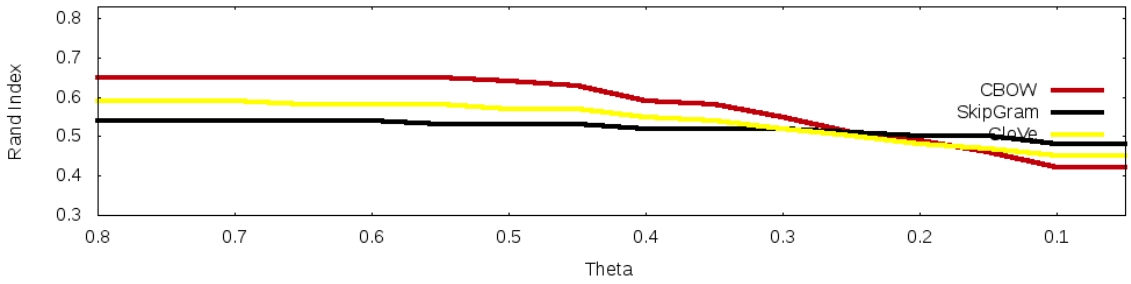

FIG. 6.7: Évolution du Rand Index selon θ, anglais, $\delta = 2$

Mais le phénomène intéressant ici c'est qu'avec $\delta = 2$, les courbes s'écartent et les méthodes ne sont plus aussi proches. Pour la F-mesure, SkipGram passe en tête dès $\theta = 0,3$, avec GloVe en second et CBOW en dernier. Pour Rand, CBOW est largement en-tête pour la plupart des valeurs, Glove est second, mais à l'approche du point d'inflexion, la situation s'inverse, et c'est SkipGram qui passe en tête. Donc si on prend comme référence le point d'inflexion, SkipGram est bon premier, Glove second pour RI et troisième pour FM, et l'inverse avec CBOW.

Nous nous contenterons d'un exemple avec CBOW pour illustrer l'évolution croisée quand $\delta = 2$, avec la figure 6.8. On remarque qu'ici les courbes se croisent en 0,15, qui est aussi le point d'inversion de l'ordre des méthodes (et donc celui où elles donnent toutes le même résultat.

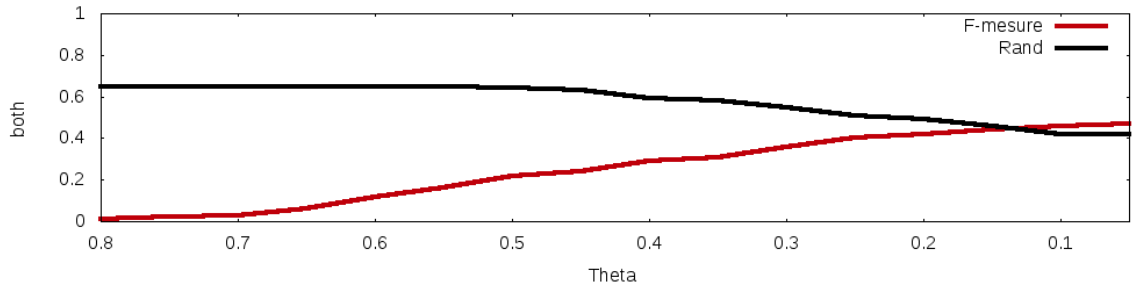

FIG. 6.8: FM et RI selon θ, CBOW, anglais, $\delta = 1$

6.10.5.3 Français, $\delta = 1$

Nous allons maintenant étudier la situation pour le français, corpus complet. Les figures 6.9 et 6.10 montrent respectivement l'évolution de la F-mesure et de Rand Index en fonction de θ pour le corpus français complet avec $\delta = 1$.

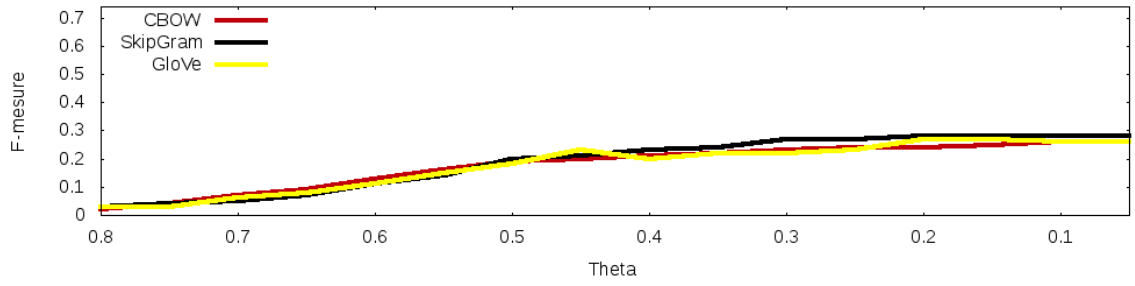

FIG. 6.9: Évolution de la F-mesure selon θ, français, $\delta = 1$

Les courbes ont les mêmes formes et les mêmes propriétés que pour l'anglais. On le montre seulement pour CBOW, avec la figure 6.11.

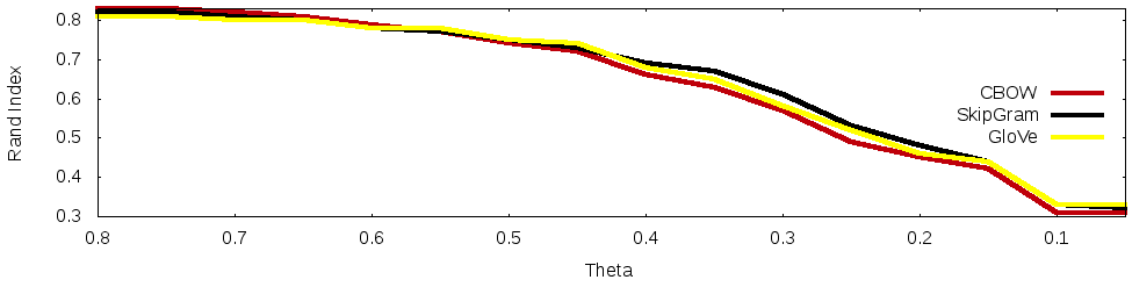

FIG. 6.10: Évolution du Rand Index selon θ, français, $\delta = 1$

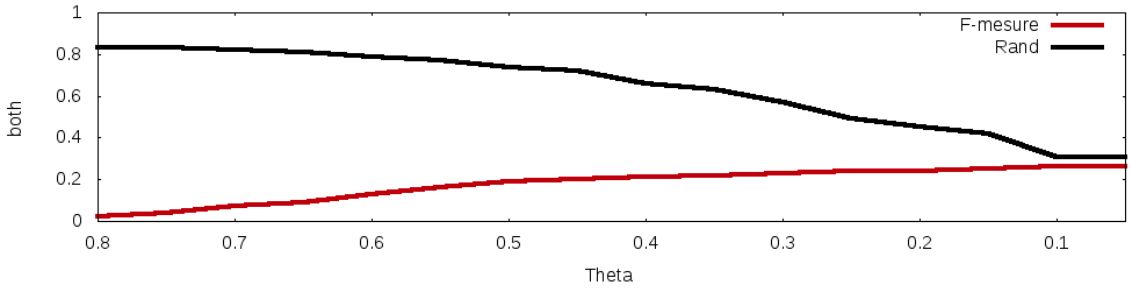

FIG. 6.11: FM et RI selon θ, CBOW, français, $\delta = 1$

6.10.5.4 Français, $\delta = 2$

Les figures 6.12 et 6.13 montrent la situation avec $\delta = 2$. La situation est la même qu'en anglais, avec l'écartement des courbes, l'ordre des méthodes est le même, SkipGram en tête, et les valeurs de F-mesure doublent par rapport à $\delta = 1$.

L'exemple de CBOW illustre le même croisement que pour l'anglais, à peu près au même point : voir la figure 6.14.

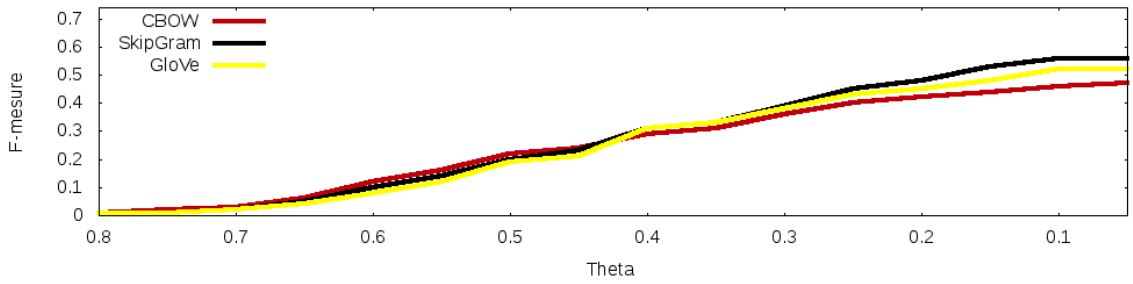

FIG. 6.12: Évolution de la F-mesure selon θ, français, $\delta = 2$

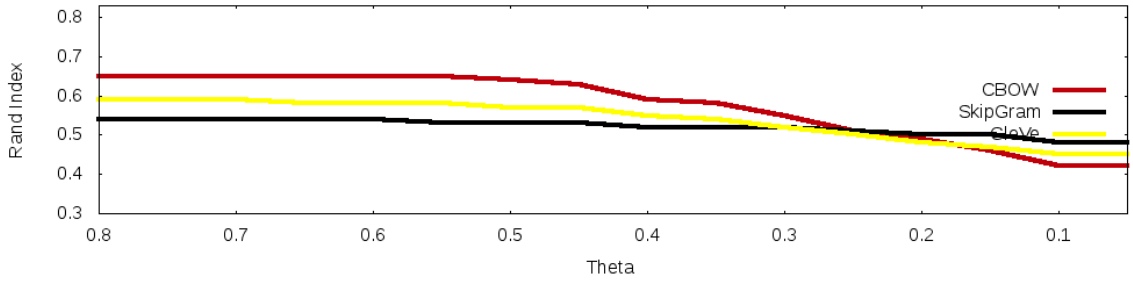

FIG. 6.13: Évolution du Rand Index selon θ, français, $\delta = 2$

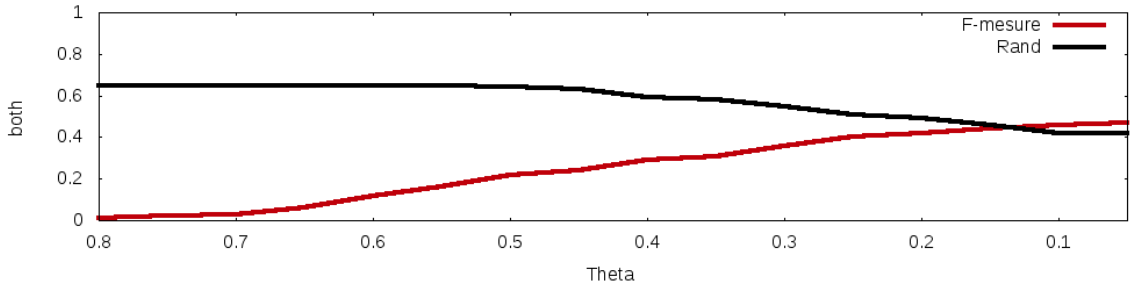

FIG. 6.14: FM et RI selon θ, CBOW, français, $\delta = 1$

6.10.5.5 Arabe, $\delta = 1$

Nous allons maintenant étudier la situation pour l'arabe, corpus complet. Les figures 6.15 et 6.16 montrent respectivement l'évolution de la F-mesure et de Rand Index en fonction de θ pour le corpus arabe complet avec $\delta = 1$.

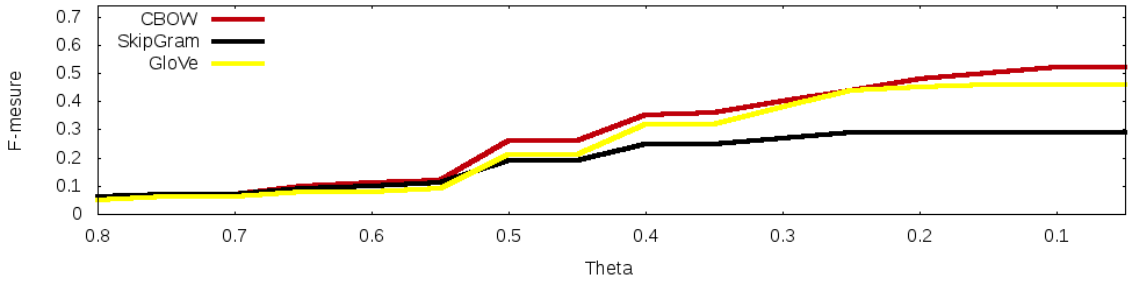

FIG. 6.15: Évolution de la F-mesure selon θ, arabe, $\delta = 1$

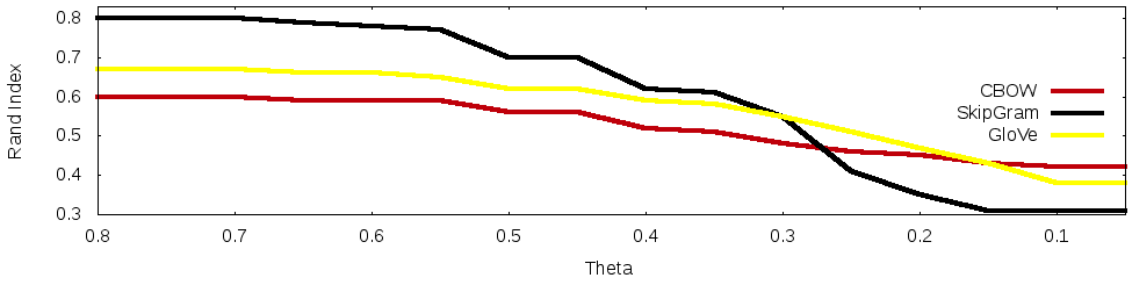

FIG. 6.16: Évolution du Rand Index selon θ, arabe, $\delta = 1$

Les courbes ont globalement les mêmes caractéristiques qu'avec les autres langues, mais avec une différence notable, la séparation des méthodes dès $\delta = 1$, avec l'ordre CBOW > GloVe > SkipGram. On montre le croisement des courbes seulement pour CBOW, avec la figure 6.17.

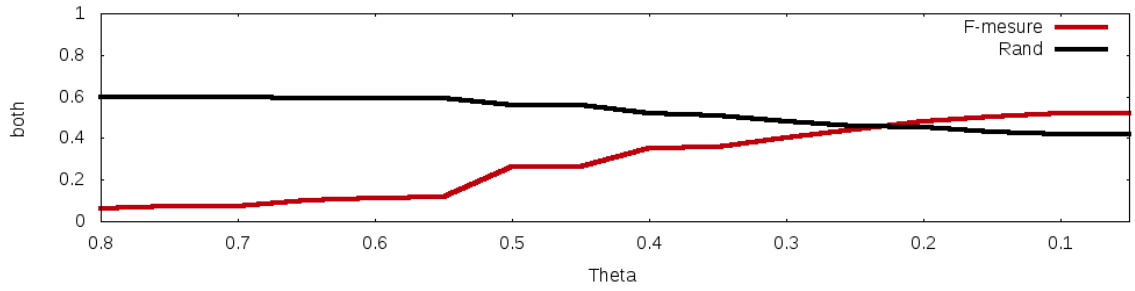

FIG. 6.17: FM et RI selon θ, CBOW, arabe, $\delta = 1$

6.10.5.6 Arabe, $\delta = 2$

Les figures 6.18 et 6.19 montrent la situation avec $\delta = 2$. L'ordre des méthodes est le même, CBOW > GloVe > SkipGram en tête, et les valeurs de F-mesure sont en augmentation par rapport à $\delta = 1$.

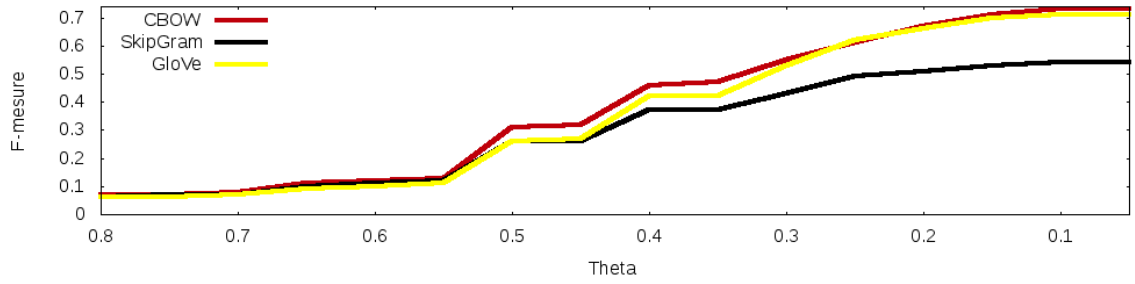

FIG. 6.18: Évolution de la F-mesure selon θ, arabe, $\delta = 2$

L'exemple de CBOW illustre le même croisement que pour l'anglais, à peu près au même point : voir la figure 6.20.

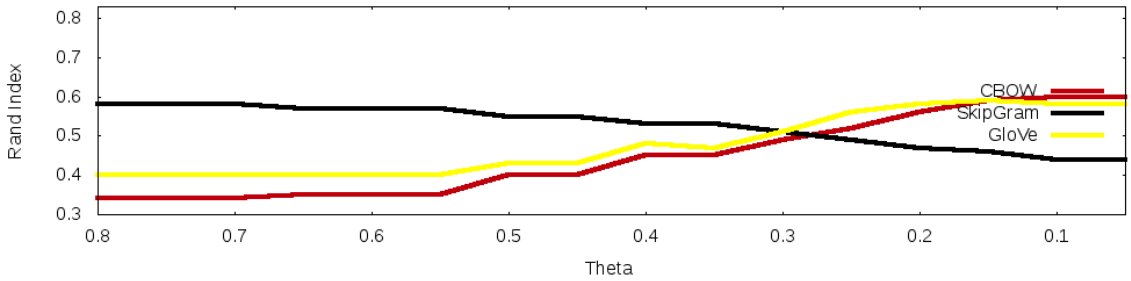

FIG. 6.19: Évolution du Rand Index selon θ, arabe, $\delta = 2$

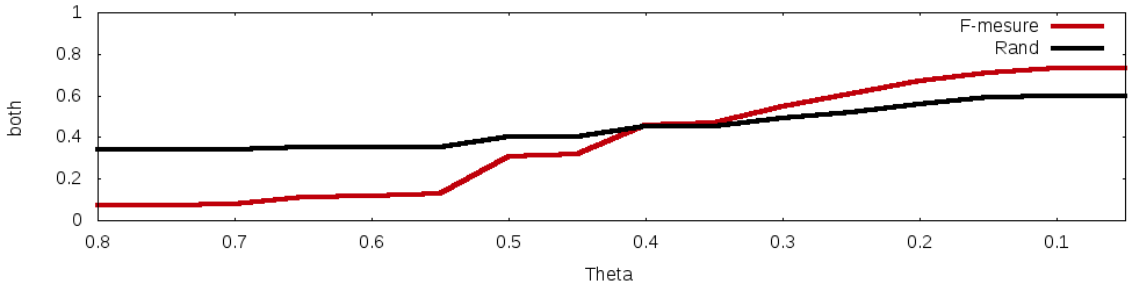

FIG. 6.20: FM et RI selon θ, CBOW, arabe, $\delta = 2$

6.10.5.7 Discussion

Pour résumer notre propos, sur les quelques expérimentations que nous avons fait, nous sommes simplement arrivés à déterminer les meilleurs paramètres (parmi ceux testés) pour θ (0,1), et pour δ (2). On trouve deux ordres, avec toujours GloVe en second, et soit CBOW (pour l'arabe et dès $\delta = 1$), soit SkipGram (pour les autres) en tête. Ici beaucoup d'explorations restent à faire, pour déterminer si par exemple le cas particulier de l'arabe est lié à la pauvreté de AWN, et si l'accord entre les autres langues est vraiment probant. Par ailleurs, il faudrait aussi établir, à l'aide de comparaisons fines, si cette mesure constitue bien une évaluation possible, quel type de relations elle évalue.

6.11 Évaluation directe

Nous passons en revue les caractéristiques propres et le pouvoir discriminant de chaque gold standard, puis faisons une étude plus approfondie de celui que nous

avons introduits. Nous examinons ensuite les résultats d'ensemble de nos sondages, pour finir avec l'évaluation par *wup*, qui forme une variante d'évaluation directe.

6.11.1 Gold standards usuels

Les gold standards usuels, définis sur l'anglais, discriminent tous clairement WebSOM. Quasiment tous, sauf Verb-143, favorisent CBOW, même si c'est avec une mauvaise corrélation (SimLex, avec 0,35) ; pour TOEFL, si on prend la méthode par notation, et pour Verb-143, c'est SkipGram qui donne les meilleurs résultats. Le tableau 6.65 montre l'exemple du corpus anglais réduit, qui est celui qui a le plus de méthodes représentées ; mais les résultats avec le corpus complet confirment ceux-là. En italique les résultats les plus faibles, en gras les plus élevés.

	MC	RG	WS	MEN	MTurk	RW	Verb	SL	TOEFL
WebSOM	*0.36*	*0.27*	*0.31*	*0.19*	*0.14*	*0.05*	*0.05*	*0.10*	*0.41 (0.64)*
CBOW	**0.83**	**0.77**	**0.71**	**0.74**	**0.66**	**0.49**	0.49	**0.35**	**0.66** (0.84)
SG	0.80	0.69	0.69	0.73	0.65	0.45	**0.55**	0.33	0.64 (**0.87**)
GloVe	0.68	0.68	0.65	0.70	0.64	0.46	0.42	0.33	0.64 (0.85)
Fasttext	0.66	0.56	0.60	0.73	0.63	0.42	0.32	0.22	0.51 (0.74)

TAB. 6.65: Gold standards usuels, corpus anglais réduit

FastText est le deuxième plus mauvais (avec une note médiane) après WebSOM, pour tous les gold standards, sauf MEN.

Par conséquent l'ensemble de ces données permet de classer CBOW comme meilleure méthode, SkipGram comme deuxième méthode, suivi de très près par Glove. Ensuite vient FastText, et WebSOM comme bon dernier. L'unique gold standard relationnel, Google Analogy, présente exactement le même classement, ce qui vient confirmer sa validité.

Mais d'autres éléments sont intéressants dans ce tableau, par rapport aux gold standards eux-mêmes : MEN favorise FastText, et, surtout, SimLex présente une corrélation médiocre pour les trois meilleures méthodes. On peut ici soit remettre en question SimLex en tant que gold standard, ou se demander si l'ensemble des méthodes présentées est encore loin d'atteindre la qualité de représentation exigée par ce gold standard. Le fait qu'il y ait corrélation dans les résultats avec les autres gold standards favorise ce dernier point de vue.

6.11.2 Gold standards adaptés

La situation concernant les gold standards adaptés à d'autres langues est très différente. Voyons par exemple le tableau 6.66 qui présente les résultats sur le corpus réduit en français. Cette fois, si WebSOM a toujours les plus mauvais résultats, c'est plutôt SkipGram qui domine, CBOW en deuxième ; la troisième place est disputée entre les deux autres.

	MC	RG	WS	SL
WebSOM	*0.11*	*0.27*	*0.06*	*0.12*
CBOW	0.69	0.65	0.52	**0.26**
SG	**0.71**	**0.66**	**0.53**	0.21
GloVe	0.66	0.61	**0.53**	0.19
Fasttext	0.67	0.58	0.45	0.23

Tab. 6.66: Gold standards adaptés, corpus français réduit

La faiblesse des corrélations avec SimLex adapté est accentuée. D'une manière générale, les corrélations sont moins bonnes, ce qui peut être dû à l'adaptation elle-même.

Le tableau 6.67 montre les résultats pour l'arabe sur le corpus standard, celui sur lequel le plus de méthodes ont été appliquées. On y constate une situation beaucoup plus hétéroclite. WebSOM n'est plus la plus mauvaise méthode, elle est la meilleure pour Spearman avec MC-30 et RG-65. Et comme nous l'avons déjà vu, la divergence des corrélations Pearson | Spearman est un autre indice. Bien qu'elle ne soit pas vérifiée pour toutes les méthodes, elle l'est pour toutes les méthodes prédictives. Nous pouvons en conclure soit que les gold standards ne sont pas adaptés pour cette évaluation, ou que les méthodes prédictives ne sont pas efficaces pour l'arabe.

La divergence des corrélations est observée aussi avec WebSOM pour les gold standards anglais MC30 et RG65, ainsi qu'avec WordSim. Mais dans ce cas, et comme il s'agit du seul cas où on observe cette divergence en anglais, et de la plus mauvaise méthode de vectorisation pour l'ensemble des évaluations fiables, on peut estimer qu'elle est plus due à la vectorisation qu'aux gold standards. De plus, cette divergence va dans le sens inverse, donc ne peut pas être attribuée à la même cause.

Dans tous les cas de figure, ces évaluations sur l'arabe ne sont pas probantes.

	MC	RG	WS
WebSOM	**0.92**	0.71	0.08
CBOW	0.46 | 0.95	0.52 | 0.92	**0.53**
SG	0.68 | 0.95	0.56 | 0.92	0.50
GloVe	**0.77 | 0.96**	**0.66 | 0.94**	0.43
Fasttext	*0.19 | 0.78*	*0.28 | 0.75*	0.48
GraPaVec	0.43 | 0.84	0.63 | 0.80	0.13

TAB. 6.67: Gold standards adaptés, corpus standard arabe

6.11.3 WNA

WNA est le plus grand gold standard attributionnel. Son support représente pratiquement 100% (de WNA) quand le corpus est grand. Quand le corpus est petit, le support reste le même en anglais, baisse en français (66,5%) et plus en arabe (29,5%, 14,1% pour le plus petit corpus). La couverture de PWN est largement supérieure à celles des deux autres ; celle de Wolf est supérieure à celle de AWN, et il semble bien que ce soit ce facteur qui explique la dégradation de la représentativité du support. Malgré tout, même dans les cas les moins représentatifs, WNA reste la plus grande ressource disponible.

Le tableau 6.68 récapitule les résultats de WNA, avec les résultats par langue, corpus et méthode. On constate que les corrélations vont de 0.01 à 0.31, ce qui rend ce gold standard inutilisable pour la comparaison. Les scores varient selon les langues, tout en restant très faibles : le plus bas est l'anglais (de 0.02 à 0.15, la plupart des scores étant en dessous de 0.10), juste derrière le français (de 0.07 à 0.15, la plupart des scores étant au dessus de 0.10), le plus haut est l'arabe (de 0.10 à 0.24), comme si ces scores étaient inversement proportionnels à la couverture du WordNet. En revanche il ne semble ressortir aucun lien avec la méthode choisie, et il n'y a pas de corrélation avec les autres gold standards, même pour l'anglais.

Langue	**Corp**	WebSOM	CBOW	SkipGram	GloVe	FastText	GraPaVec
Anglais	Wiki complet	-	0.04	0.02	0.04	-	-
	Wiki réduit	0.15	0.07	0.05	0.06	0.09	-
Français	Wiki complet	-	0.11	0.15	0.11	-	-
	Wiki réduit	0.11	0.12	0.08	0.10	0.07	-
Arabe	Grand corpus	-	0.24	0.21	0.24	-	-
	Wiki complet	-	0.20	0.15	0.10	-	-
	Wiki réduit	0.19	0.20	0.16	0.12	0.15	-
	Corpus standard	0.19	0.16	0.18	0.10	0.18	0.15

TAB. 6.68: WNA

Après avoir fait ce constat, nous avons cherché à savoir pourquoi, malgré une taille conséquente et une représentativité tout à fait correcte, au moins dans les grands corpus, les résultats étaient aussi médiocres. Nous avons étudié le corpus des paires aléatoires, et constaté que d'une part elles sont effectivement pour la plupart des associations improbables, comme «recto - pastel», mais d'autre part que beaucoup de paires de ce type se retrouvaient avec une note entre 0.20 et 0.35, ce qui est bien supérieur à ce qu'un humain aurait donné comme note (probablement 0). Le lien avec la taille du WordNet trouve sans doute son explication par là : étant donné la façon dont WordNet est constitué (par synset), un petit WordNet a beaucoup moins de chance de générer des paires complètement aléatoires qu'un grand WordNet.

Cette note relativement élevée est liée au fait que dans la hiérarchie de WordNet tous les termes sont descendants d'un même noeud et donc tous apparentés. Nous suggérons que la métrique de *wup_sim* devrait être transformée pour obtenir une distribution plus gaussienne, en donnant une note plus forte aux vrais voisins et une note plus faible aux mots très éloignés. On pourrait également envisager de ne retenir que des paires dont les mots ont une fréquence plus grande dans le corpus.

En tout état de cause, nous pouvons pour l'instant retenir que cette méthode d'évaluation est un échec complet.

6.11.4 Évaluation par sondage

Nous étions limité, concernant les participants au sondage, par le nombre de mots à comparer de manière raisonnable. Au départ, lors du sondage sur le français, nous avions estimé que plus de trois mots dans les réponses serait difficile pour les participants. Pour l'arabe, nous avons testé avec quatre mots et avons constaté que cela ne posait pas de problème. Le résultat du sondage est présenté dans le tableau 6.69.

	français	arabe
WebSOM	*0.18*	*0.26*
CBOW	**0.62**	0.69
SkipGram	-	**0.70**
GloVe	0.53	0.66

Tab. 6.69: Résultat des sondages

Nous avons constaté un grand accord inter-évaluateurs. Malgré leur caractère incomplet, ces résultats confirment la mauvaise évaluation de WebSOM et la meilleur évaluation de CBOW / SkipGram (très proches l'un de l'autre), ajoutant un indice de plus à notre évaluation sur les langues autres que l'anglais.

6.12 Évaluation interne par substitution

Pour pouvoir faire la comparaison entre le maximum de méthodes de représentation, nous n'avons appliqué cette évaluation que sur les corpus réduits et sur Corpus standard arabe. Nous écartons FastText des méthodes en raison de son fonctionnement sur les mots absents du corpus (la représentation des mots dupliqués et la représentation des mots d'origine est quasiment la même en termes de ngrammes). Nous n'avons pas eu accès à la méthode Grapavec (en cours de réfection) et donc n'avons pas pu l'utiliser.

Nous avons fixé le nombre de mots à dupliquer à 1000. Nous avons fixé pour chaque corpus un seuil minimum pour déterminer le vocabulaire, et deux seuils, un minimum et un maximum, pour les mots à dupliquer, conformément à l'équation 5.22. Voici la liste des valeurs :

1. Corpus réduit anglais : $\#(m) \geq 100$: $|V| = 34\,297$; $250 \leq \#(m) \leq 122\,675$.
2. Corpus réduit français : $\#(m) \geq 100$: $|V| = 29\,349$; $250 \leq \#(m) \leq 72\,191$.
3. Corpus réduit arabe : $\#(m) \geq 100$: $|V| = 23\,401$; $250 \leq \#(m) \leq 27\,504$.
4. Corpus standard arabe : $\#(m) \geq 100$: $|V| = 9\,843$; $250 \leq \#(m) \leq 9\,972$.

Le tableau 6.70 montre les résultats de la méthode par substitution. La meilleure représentation selon cette méthode est CBOW, suivi de près par Skip-Gram et Glove ; la plus mauvaise représentation est WebSOM. Les résultats sont donc parfaitement en accord avec l'évaluation directe pour l'anglais. La seule remarque qu'on peut faire est que l'écart entre WebSOM et CBOW est plus réduit ici qu'avec les gold standards.

	anglais réduit	français réduit	arabe réduit	corpus standard arabe
WebSOM	0.77	*0.88*	*0.53*	*0.60*
CBOW	**0.97**	**0.96**	**0.90**	**0.93**
SG	0.95	0.94	0.89	0.90
GloVe	0.96	0.94	0.88	0.87

TAB. 6.70: Résultats de la méthode par substitution

6.13 Conclusion

Plusieurs éléments importants se dégagent de cette étude, et il y en a sans doute beaucoup d'autres que nous n'avons pas eu le temps de dégager.

Par exemple, concernant le choix entre la distance euclidienne et le cosinus pour comparer les vecteurs de mots, qui est une question récurrente, nos milliers de tests ont montré que le cosinus est systématiquement meilleur pour comparer avec les gold standards attributionnels, sauf quand les vecteurs sont normalisés ou quasi-normalisés (par la méthode du *random mapping*). Mais même dans ce cas, les résultats sont très proches et ne justifient pas de conserver les deux mesures.

Les meilleurs paramètres pour les différents modèles de clustering ont été discutés dans la partie qui leur est consacrée. Le seul constat commun à toutes les méthodes concerne la normalisation des vecteurs d'entrée : les résultats sont toujours meilleurs sans normalisation pour nos données.

Pour ceux des méthodes de représentation vectorielle, qui ont été donnés pour chacun, le seul constat commun à toutes les méthodes c'est que la dimension des vecteurs la plus grande parmi les tests effectués est en général la meilleure, sauf pour WebSOM, où c'est la dimension médiane de nos tests qui s'est avérée la meilleure. En revanche, la meilleure taille de la fenêtre est variable suivant le modèle.

Les autres éléments ayant déjà été discutés dans leur partie, il est temps maintenant de passer à la conclusion générale.

Conclusion

Dans notre conclusion, nous reprenons rapidement les faits les plus marquants concernant les évaluations existantes, et une appréciation particulière pour nos propres évaluations.

Tout d'abord, reprenons deux enseignements généraux : il vaut mieux utiliser le cosinus que la distance euclidienne pour le calcul des ressemblances entre vecteurs de mots, et ne pas normaliser ces vecteurs pour l'évaluation semi-directe (ce n'est pas vrai pour d'autres types de données, voir par exemple la base IRIS).

Ensuite, sur l'anglais, tous corpus confondus, nous avons observé une convergence globale entre les évaluations directes avec les gold standards traditionnellement utilisés, l'évaluation directe par sondage, l'évaluation interne par substitution et l'évaluation semi-directe par wup.

Toutes ces évaluations indiquent que WebSOM est la plus mauvaise représentation, et que la meilleure représentation est l'une de celles de Word2Vec. Entre CBOW et SkipGram, la plupart des évaluations donnent CBOW en tête, et une minorité donne SkipGram en tête. Dans tous les cas, la deuxième meilleure représentation est aussi dans Word2Vec. La troisième représentation, en général pas loin derrière, est GloVe. Les écarts qui séparent le peloton de tête (CBOW - Skipgram - Glove) ne sont pas en général très importants, quelle que soit l'évaluation. L'évaluation semi-directe par BLESS sur l'anglais a donné GloVe en tête, CBOW en second et SkipGram en troisième, et c'est la seule dans nos tests à donner ainsi la première position à Glove. FastText est quatrième, WebSOM est bon dernier. Sur GraPaVec, nos quelques expérimentations (sur le corpus standard arabe) le mettent en troisième pour l'évaluation semi-directe, et en dessous de WebSOM pour l'évaluation directe.

FastText et GraPaVec sont plus difficilement comparables, l'un en raison de sa visée particulière à représenter des mots absents du corpus d'apprentissage, excluant son évaluation comparée en l'absence de tels mots dans le gold standard (et de sa gourmandise en mémoire), l'autre en raison de données très partielles (par conséquent les résultats sont à prendre avec des pincettes).

Parmi nos méthodes, l'évaluation directe par sondage et l'évaluation interne

par substitution confirment le classement dominant. L'évaluation semi-directe par wup classe SkipGram en tête, en convergence avec une minorité des évaluations traditionnelles, et en convergence avec une partie des évaluations semi-directes par WNC (dans l'ordre : SkipGram, CBOW, GloVe). Au contraire de BLESS, nous avons pu valider ces résultats sur deux langues ; par contre, pour l'arabe, nous n'avons pas pu les valider, sans doute en raison de la pauvreté de AWN.

L'évaluation par WNA s'est avérée désastreuse et uniquement inversement corrélée avec la richesse du WordNet considéré (plus le WordNet est riche et moins le résultat de l'évaluation est bon), avec aucun autre facteur.

En retour, à l'exception de WNA, ces résultats valident nos méthodes, ce qui nous permet d'ajouter des évaluations pour les langues autres que l'anglais, où nous sommes très peu pourvus, ce qui était un des objectifs majeurs de cette thèse.

Sur les autres langues, sans équivoque, toutes ces évaluations (y compris celle par wup) sauf une (WNC, citée ci-dessus, inverse les deux premiers) donnent CBOW en tête, SkipGram en second, Glove en troisième, FastText en quatrième et WebSOM bon dernier.

Ceci nous permet déjà de tirer quelques conclusions sans appel sur certains gold standards utilisés dans la littérature, qui ne se conforment pas du tout à ce résultat et sont donc invalidés.

Pour l'arabe, seul WordSim adapté donne des résultats compatibles. Cela est encore confirmé par la divergence entre les corrélations de Spearman et Pearson sur les autres gold standards. Pour le français, SimLex adapté est le plus proche des résultats obtenus ; mais les autres restent compatibles dans la mesure où le trio de tête reste le même et où les corrélations entre les membres du trio ne sont pas très différentes ; ils ne peuvent donc pas être totalement invalidés. Mais leur fiabilité est moins grande que celle de SimLex adapté.

L'évaluation topologique reste à explorer et à justifier sur des résultats plus fiables que ceux que nous avons obtenus.

Bien que notre gold standard WNA se soit avéré un échec, nous en avons analysé les causes, et nous ne désespérons pas d'y remédier.

D'une manière générale, la divergence d'une évaluation avec les autres peut avoir deux significations : l'évaluation n'est pas sensible aux mêmes effets que les autres, ou elle est mal conçue. S'agissant de WNA, tout converge pour indiquer que ce à quoi elle est sensible n'a rien à voir avec l'objectif d'évaluer les représentations vectorielles.

En bref, sur les six évaluations que nous avons proposées, une a complètement échoué (WNA), deux autres remplissent parfaitement leur rôle et permettent une évaluation en l'absence de gold standard (substitution et wup) ; une quatrième, le sondage a confirmé les autres résultats, ce qui semble valider le protocole adopté.

La cinquième, WNC, s'il a bien fonctionné sur le français et l'anglais, n'a pas donné de résultats convaincants sur l'arabe.

La sixième, l'évaluation topologique, reste à l'état de piste. Nous pensons qu'il est intéressant de la poursuivre, car il s'agit de l'unique évaluation externe d'une topologie de clusters qui ait jamais été proposée, en tout cas concernant la catégorisation de mots. En effet, pour ces topologies, il n'existe que des évaluations internes.

Les propriétés de WordNet lui ouvrent de nombreuses potentialités comme gold standard. Avec les 200 000 lemmes indexés dans PWN, on peut calculer à peu près 20 milliards de similitudes. En comparaison, même la plus grande ressource, MEN, avec ses 3000 paires, fait pâle figure. Par conséquent, en tirer des gold standards reste à notre avis une tâche à continuer, ainsi que l'enrichir. Wolf pourrait par exemple être enrichi par Glawi, tiré de Wiktionnaire, pour tendre vers la richesse de PWN.

Selon Sagot (2017), «À ce jour, le WOLF n'a pas encore fait l'objet d'une évaluation orientée-tâche». Par conséquent, secondairement, ce travail introduit une des premières évaluations orientées tâche de Wolf. On a pu voir ici que, malgré son caractère de *silver standard*, il a permis une évaluation des représentations vectorielles pour le français, et par conséquent, au contraire de AWN dans certaines de ces évaluations, il est validé pour notre tâche.

Nous espérons que ce travail a tenu quelques unes de ses promesses. Il nous a en tout cas ouvert des perspectives de recherche.

Mes publications

[5] Nourredine Aliane, Jean-Jacques Mariage et Gilles Bernard. «Étude comparative entre word2vec, GloVe et WebSOM en utilisant WordNet». In : *Atelier "Fouille de textes à EGC'2018*. Paris, jan. 2018.

[6] Nourredine Aliane, Jean-Jacques Mariage et Gilles Bernard. «L'évaluation des représentations vectorielles de mots en utilisant WordNet». In : *Actes de la 25e conférence sur le Traitement Automatique des Langues Naturelles (TALAN2018)*. Rennes, France, Mai 2018.

[23] Gilles Bernard, Nourredine Aliane et Otman Manad. «An Experimentation Line for Underlying Graphemic Properties - Acquiring Knowledge from Text Data with Self Organizing Maps». In : *ICINCO 2015 - Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, Volume 1, Colmar, Alsace, France, 21-23 July, 2015.* 2015, p. 659–666. URL : https://doi.org/10.5220/0005577706590666.

[83] Georges Lebboss, Gilles Bernard, Nourredine Aliane et Mohammad Hajjar. «Towards the Enrichment of Arabic WordNet with Big Corpora». In : *Proceedings of the 9th International Joint Conference on Computational Intelligence, IJCCI 2017, Funchal, Madeira, Portugal, November 1-3, 2017.* 2017, p. 101–109. DOI : 10.5220/0006505701010109. URL : https://doi.org/10.5220/0006505701010109.

[98] Otman Manad, Nourredine Aliane et Gilles Bernard. «Un protocole d'expérimentation sur les propriétés graphémiques avec l'algorithme SOM». In : *16ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2016, 18-22 Janvier 2016, Reims, France*. 2016, p. 105–110.

Bibliographie

[1] Lahsen Abouenour, Karim Bouzoubaa et Paolo Rosso. «On the Evaluation and Improvement of Arabic WordNet Coverage and Usability». In : *Lang. Resour. Eval.* 47.3 (sept. 2013), p. 891–917. issn : 1574-020X. doi : 10.1007/s10579-013-9237-0. url : http://dx.doi.org/10.1007/s10579-013-9237-0.

[2] Dimitris Achlioptas. «Database-friendly Random Projections : Johnson-Lindenstrauss with Binary Coins». In : *J. Comput. Syst. Sci.* 66.4 (juin 2003), p. 671–687. issn : 0022-0000. doi : 10.1016/S0022-0000(03)00025-4. url : http://dx.doi.org/10.1016/S0022-0000(03)00025-4.

[3] Judit Ács et Andras Kornai. «Evaluating embeddings on dictionary-based similarity». In : *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany : Association for Computational Linguistics, 2016, p. 78–82. doi : 10.18653/v1/W16-2514. url : http://aclweb.org/anthology/W16-2514.

[4] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca et Aitor Soroa. «A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches». In : *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. NAACL '09. Boulder, Colorado : Association for Computational Linguistics, 2009, p. 19–27. isbn : 978-1-932432-41-1. url : http://dl.acm.org/citation.cfm?id=1620754.1620758.

[5] Nourredine Aliane, Jean-Jacques Mariage et Gilles Bernard. «Étude comparative entre word2vec, GloVe et WebSOM en utilisant WordNet». In : *Atelier "Fouille de textes à EGC'2018*. Paris, jan. 2018.

[6] Nourredine Aliane, Jean-Jacques Mariage et Gilles Bernard. «L'évaluation des représentations vectorielles de mots en utilisant WordNet». In : *Actes de la 25e conférence sur le Traitement Automatique des Langues Naturelles (TALAN2018)*. Rennes, France, Mai 2018.

[7] Abdulrahman Almuhareb et Massimo Poesio. «Concept learning and categorization from the web». In : *Cognitive Science - COGSCI* (jan. 2005).

[8] Sanjeev Arora, Li Yuanzhi, Liang Yingyu, Ma Tengyu et Risteski Andrej. «A Latent Variable Model Approach to PMI-based Word Embeddings». In : *Transactions of the Association for Computational Linguistics* 4 (2016), p. 385–399. URL : http://aclweb.org/anthology/Q16-1028.

[9] David Arthur et Sergei Vassilvitskii. «K-means++ : the advantages of careful seeding». In : *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. 2007.

[10] Umut Asan et Secil Ercan. «Computational Intelligence Systems in Industrial Engineering : with Recent Theory and Applications». In : sous la dir. de C. Kahraman. Atlantis Press, jan. 2012. Chap. 14 An Introduction to Self-Organizing Maps, p. 299–319.

[11] Oded Avraham et Yoav Goldberg. «Improving Reliability of Word Similarity Evaluation by Redesigning Annotation Task and Performance Measure». In : *CoRR* abs/1611.03641 (2016). arXiv : 1611.03641. URL : http://arxiv.org/abs/1611.03641.

[12] Simon Baker et Anna Reichart Roiand Korhonen. «An Unsupervised Model for Instance Level Subcategorization Acquisition». In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar : Association for Computational Linguistics, 2014, p. 278–289. DOI : 10.3115/v1/D14-1034. URL : http://aclweb.org/anthology/D14-1034.

[13] Marco Baroni, Georgiana Bernard et Germán Kruszewski. «Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors». In : *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference* 1 (2014), p. 238–247.

[14] Marco Baroni et Alessandro Lenci. «How We BLESSed Distributional Semantic Evaluation». In : *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. GEMS '11. Edinburgh, Scotland : Association for Computational Linguistics, 2011, p. 1–10. ISBN : 978-1-937284-16-9. URL : http://dl.acm.org/citation.cfm?id=2140490.2140491.

[15] Marco Baroni, Brian Murphy, Eduard Barbu et Massimo Poesio. «Strudel : A Corpus-Based Semantic Model Based on Properties and Types». In : *Cognitive Science* 34.2 (2010), p. 222–254. DOI : 10.1111/j.1551-

6709.2009.01068.x. URL : https://doi.org/10.1111/j.1551-6709.2009.01068.x.

[16] Siamak Barzegar, Brian Davis, Manel Zarrouk, Siegfried Handschuh et André Freitas. «SemR-11: A Multi-Lingual Gold-Standard for Semantic Similarity and Relatedness for Eleven Languages». Anglais. In : *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan : European Language Resources Association (ELRA), mai 2018. ISBN : 979-10-95546-00-9.

[17] Yoshua Bengio, Réjean Ducharme, Pascal Vincent et Christian Jauvin. «A Neural Probabilistic Language Model». In : *Journal of Machine Learning Research* 3 (2003), p. 1137–1155.

[18] Younès Bennani. *Apprentissage connexionniste*. Paris : Hermes science publications, Lavoisier, 2006.

[19] Gilles Bernard. «Experiments on Distributional Categorization of Lexical Items with Self Organizing Maps». In : *International Workshop on Self Organizing Maps WSOM'97*. (Helsinki University of Technology, Espoo, Finland, 1997). 1997, p. 304–309.

[20] Gilles Bernard. «L'épithète et l'ordre des mots dans le syntagme substantif». In : *22ième Congrès de Linguistique et Philologie Romane*. (Bruxelles, Belgique). Sous la dir. d'A. Englebert, M. Pierrard, L. Rosier et D. Van Raemdonck. Max Niemeyer, 2000, p. 47–54.

[21] Gilles Bernard. «Opérationalisme et mécanisation : la modélisation de l'interprétation». In : *Parcours interprétatifs et parcours énonciatifs : Théories et applications*. (Tromsø University, Norway). Sous la dir. d'Aboubakar Ouattara. Ophrys, Paris, 2003, p. 100–110.

[22] Gilles Bernard. «Un système minimaliste et dynamique de détection de catégories distributionnelles». In : *5ème colloque Hypertextes et Hypermédias : Réalisations, outils, méthodes*. (Paris, France). Sous la dir. de Jean-Pierre Balpe, Alain Lelu, Stéphane Natkin et Imad Saleh. Hermès, 1999, p. 229–241.

[23] Gilles Bernard, Nourredine Aliane et Otman Manad. «An Experimentation Line for Underlying Graphemic Properties - Acquiring Knowledge from Text Data with Self Organizing Maps». In : *ICINCO 2015 - Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, Volume 1, Colmar, Alsace, France, 21-23 July, 2015.* 2015, p. 659–666. URL : https://doi.org/10.5220/0005577706590666.

[24] Gilles Bernard et Georges Lebboss. «Methods for word encoding : a survey». In : *Proceedings of 2017 International Conference on Engineering & Technology*. (Akdeniz University, Antalya, Turkey). IEEE, 2017, p. 7–12.

[25] Gilles Bernard et Jean-Jacques Mariage. «Post-Processing of Grammatical Patterns produced by Self Organizing Maps». In : *8th Conference on Pattern Recognition and Information Processing PRIP'05*. (Minsk, Belarus, 2005). 2005, p. 412–415.

[26] Yves Bestgen. «Improving Text Segmentation Using Latent Semantic Analysis : A Reanalysis of Choi, Wiemer-Hastings, and Moore». In : *Computational Linguistics* 32.3 (2006), p. 455.

[27] Ella Bingham et Heikki Mannila. «Random Projection in Dimensionality Reduction : Applications to Image and Text Data». In : *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '01. San Francisco, California : ACM, 2001, p. 245–250. isbn : 1-58113-391-X. doi : 10.1145/502512.502546. url : http://doi.acm.org/10.1145/502512.502546.

[28] William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease et Christiane Fellbaum. «Introducing the Arabic WordNet Project». In : *In Proceedings of the third International WordNet Conference*. Sous la dir. de Fellbaum Sojka Choi et Vossen. Citeseer. 2006, p. 295–300.

[29] D. M. Blei, A. Y. Ng et M. I. Jordan. «Latent Dirichlet allocation». In : *Journal of Machine Learning Research* 3 (2003), p. 993–1022.

[30] Claudine Bodson. «Termes et relations sémantiques en corpus spécialisés : rapport entre patrons de relations sémantiques (PRS) et types sémantiques (TS)». Thèse de doct. Université de Montréal, 2005.

[31] Piotr Bojanowsk, Edouard Grave, Armand Joulin et Mikolov Tomas. «Enriching Word Vectors with Subword Information». In : *CoRR* (2016). arXiv : 1607.04606. url : http://arxiv.org/abs/1607.04606.

[32] Elia Bruni, Nam-Khanh Tran et Marco Baroni. «Multimodal Distributional Semantics». In : *JAIR* (2014).

[33] Guénaël Cabanes et Younès Bennani. «Apprendre les contraintes topologiques dans les cartes auto-organisatrices». In : *Extraction et gestion des connaissances (EGC'2011), Actes, 25 au 29 janvier 2011, Brest, France*. 2011, p. 137–148. url : http://editions-rnti.fr/?inprocid=1000939.

[34] François-Régis Chaumartin. «WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture». In : *Colloque BD lexicales*. Montréal, Canada, avr. 2007. URL : https://hal.archives-ouvertes.fr/hal-00611240.

[35] Kenneth Ward Church et Patrick Hanks. «Word Association Norms, Mutual Information, and Lexicography». In : *27th Annual Meeting of the Association for Computational Linguistics*. 1989.

[36] Vincent Claveau et Ewa Kijak. «Thésaurus distributionnels pour la recherche d'information et vice-versa». In : *Conférence en Recherche d'Information et Applications*. Actes de la conférence CORIA 2015. Paris, France, mar. 2015. URL : https://hal.archives-ouvertes.fr/hal-01226532.

[37] Ronan Collobert et Jason Weston. «A Unified Architecture for Natural Language Processing : Deep Neural Networks with Multitask Learning». In : *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland : ACM, 2008, p. 160–167. ISBN : 978-1-60558-205-4. DOI : 10.1145/1390156.1390177. URL : http://doi.acm.org/10.1145/1390156.1390177.

[38] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu et Pavel P. Kuksa. «Natural Language Processing (almost) from Scratch». In : *CoRR* abs/1103.0398 (2011). arXiv : 1103.0398. URL : http://arxiv.org/abs/1103.0398.

[39] Antoine Cornuéjols et Laurent Miclet. *Apprentissage artificiel : Concepts et algorithmes*. Eyrolles, juin 2010, p. 804. URL : https://hal.inria.fr/inria-00538947.

[40] David Crystal. *Dictionary of linguistics and phonetics*. T. 30. John Wiley & Sons, 2011.

[41] Fred Daum et Jim Huang. «Curse of dimensionality and particle filters». In : *Proceedings of the IEEE* 4 (2003).

[42] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas et Richard A. Harshman. «Indexing by Latent Semantic Analysis». In : *Journal of the American Society of Information Science* 41.6 (1990), p. 391–407.

[43] A.P. Dempster, N.M. Laird et D.B. Rubin. «Maximum likelihood from incomplete data via the EM algorithm». In : *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 39.1 (1977), p. 1–38.

[44] Paramveer S. Dhillon, Jordan Rodu, Dean P. Foster et Lyle H. Ungar. «Two Step CCA : A new spectral method for estimating vector models of words». In : *Proceedings of the 29th International Conference on Machine learning*. ICML'12. Edinburgh, U.K., 2012.

[45] E. Diday. «Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques». fre. In : *Revue de Statistique Appliquée* 19.2 (1971), p. 19–33. url : http://eudml.org/doc/105908.

[46] M. Dittenbach, Dieter Merkl et A. Rauber. «The Growing Hierarchical Self-Organizing Map». In : *Proceedings of the International Joint Conference on Neural Networks 2000 (IJCNN'2000)*. 2000.

[47] Sabri Elkateb, William Black, Horacio Rodríguez, Musa Alkhalifa, Piek Vossen, Adam Pease et Christiane Fellbaum. «Building a WordNet for Arabic». In : *In proceedings of the Fifth International Conference on language Ressources and Evaluation*. Citeseer. Genoa,Italy, 2006.

[48] William F. Battig et William E. Montague. «Category norms of verbal items in 56 categories A replication and extension of the Connecticut category norms». In : *Journal of Experimental Psychology* 80 (juin 1969). doi : 10.1037/h0027577.

[49] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi et Chris Dyer. «Problems With Evaluation of Word Embeddings Using Word Similarity Tasks». In : *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany : Association for Computational Linguistics, 2016, p. 30–35. doi : 10.18653/v1/W16-2506. url : http://aclweb.org/anthology/W16-2506.

[50] Christiane Fellbaum. «La représentation des verbes dans le réseau sémantique "WordNet"». In : *Langages* 33.136 (1999), p. 27–40.

[51] Christiane Fellbaum. *WORDNET an electronic lexical database*. The MIT Press Cambridge, Massachusetts, London, England, 1998.

[52] D. A. Ferrucci. «Introduction to "This is Watson"». In : *IBM J. Res. Dev.* 56.3 (mai 2012), p. 235–249. issn : 0018-8646. doi : 10.1147/JRD.2012.2184356. url : http://dx.doi.org/10.1147/JRD.2012.2184356.

[53] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman et Eytan Ruppin. «Placing Search in Context : The Concept Revisited». In : *Proceedings of the 10th International Conference on World Wide Web*. WWW '01. New York, NY, USA : ACM, 2001, p. 406–414.

[54] Arthur Flexer. «On the use of self organizing maps for clustering and visualization». In : *Intelligent Data Analysis* 5 (2001), p. 373–384.

[55] E. Forgy. «Cluster Analysis of Multivariate Data : Efficiency versus Interpretability of Classification». In : *Biometrics* 21.3 (1965), p. 768–769.

[56] Hershey Friedman et Taiwo Amoo. «Multiple Biases in Rating Scale Construction». In : *Journal of International Marketing and Marketing Research (European Marketing Association)* 24 (oct. 1999).

[57] Jerome H. Friedman. «On Bias, Variance, 0/1&Mdash ;Loss, and the Curse-of-Dimensionality». In : *Data Min. Knowl. Discov.* 1.1 (jan. 1997), p. 55–77. issn : 1384-5810. doi : 10.1023/A:1009778005914. url : https://doi.org/10.1023/A:1009778005914.

[58] Gene H. Golub et Charles F. Van Loan. *Matrix Computations*. Third. The Johns Hopkins University Press, 1996.

[59] Michael Gutmann et Aapo Hyvärinen. «Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics.» In : *Journal of Machine Learning Research* 13 (2012), p. 307–361. url : http://dblp.uni-trier.de/db/journals/jmlr/jmlr13.html#GutmannH12.

[60] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich et Yehuda Koren. «Large-scale learning of word relatedness with constraints.» In : *KDD*. Sous la dir. de Qiang Yang, Deepak Agarwal et Jian Pei. ACM, 2012, p. 1406–1414. isbn : 978-1-4503-1462-6.

[61] Zellig S Harris. «Distributional structure.» In : *Word* (1954). url : http://scholar.google.de/scholar.bib?q=info:NLqRm8tVoHMJ:scholar.google.com/%20&output=citation&hl=de&as_sdt=0,5&ct=citation&cd=0.

[62] Samer Hassan et Rada Mihalcea. «Cross-lingual Semantic Relatedness Using Encyclopedic Knowledge». In : *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 3 - Volume 3*. EMNLP '09. Singapore : Association for Computational Linguistics, 2009, p. 1192–1201. isbn : 978-1-932432-63-3. url : http://dl.acm.org/citation.cfm?id=1699648.1699665.

[63] Robert Hecht-Nielsen. «Counterpropagation Networks». In : *Applied Optics* 26.23 (1987), p. 4979–4984.

[64] Felix Hill, Roi Reichart et Anna Korhonen. «Simlex-999: Evaluating Semantic Models with Genuine Similarity Estimation». In : *Comput. Linguist.* (2015), p. 665–695.

[65] Thomas Hofmann. «Probabilistic latent semantic analysis». In : *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1999, p. 289–296.

[66] Thomas Hofmann. «Unsupervised learning by probabilistic latent semantic analysis». In : *Machine learning* 42.1 (2001), p. 177–196.

[67] H. Hotelling. «Canonical correlation analysis (cca)». In : *Journal of Educational Psychology* (1935).

[68] Guillaume Jacquet, Fabienne Venant et Bernard Victorri. «Polysémie lexicale». In : *Sémantique et traitement automatique du langage naturel*. Hermès. 2005, p. 99–132.

[69] Stanislaw Jastrzebski, Damian Lesniak et Wojciech Marian Czarnecki. «How to evaluate word embeddings ? On importance of data efficiency and simple supervised tasks». In : *CoRR* abs/1702.02170 (2017). arXiv : 1702.02170. url : http://arxiv.org/abs/1702.02170.

[70] William Johnson et Joram Lindenstrauss. «Extensions of Lipschitz mappings into a Hilbert space». In : *Conference in modern analysis and probability (New Haven, Conn., 1982)*. T. 26. Contemporary Mathematics. American Mathematical Society, 1984, p. 189–206.

[71] Armand Joulin, Edouard Grave, Piotr Bojanowsk, Matthijs Douze, Hervé Jégou et Tomas Mikolov. «FastText.zip : Compressing text classification models». In : *CoRR* abs/1612.03651 (2016). arXiv : 1612.03651. url : http://arxiv.org/abs/1612.03651.

[72] Samuel Kaski. «Dimensionality reduction by random mapping : Fast similarity computation for clustering.» In : *International Joint Conference on Neural Networks*. IEEE. 1998, p. 413–418.

[73] Dimitar Kazakov et Simon Dobnik. «Inductive learning of lexical semantics with typed unification grammars». In : (mar. 2019).

[74] Kimmo Kiviluoto. «Topology preservation in self-organizing maps». In : t. 1. Juil. 1996, 294–299 vol.1. isbn : 0-7803-3210-5. doi : 10.1109/ICNN.1996.548907.

[75] T. Kohonen, S. Kaski, K. Lagus et J. Salojarvi. «WEBSOM - Self-organizing maps of document collection». In : *Helsinki University of Technologie, Finland* (1998).

[76] T. Kohonen, M. R. Schroeder et T. S. Huang, éds. *Self-Organizing Maps*. 3rd. Berlin, Heidelberg : Springer-Verlag, 2001. isbn : 3540679219.

[77] Teuvo Kohonen. «Self-organized formation of topologically correct feature maps». In : *Biological cybernetics* 43.1 (1982), p. 59–69.

[78] Teuvo KOHONEN. «Things you haven't heard about the Self-Organizing Map». In : *1993 IEEE International Conference on Neural Networks*. IEEE. 1993, p. 1147–1156.

[79] Bart KOSKO. «Bidirectional Associative Memories». In : *IEEE Trans. Syst. Man Cybern.* 18.1 (jan. 1988), p. 49–60. ISSN : 0018-9472. DOI : 10.1109/21.87054. URL : http://dx.doi.org/10.1109/21.87054.

[80] Thomas K LANDAUER et Susan T. DUTNAIS. «A solution to Plato's problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge». In : *PSYCHOLOGICAL REVIEW* 104.2 (1997), p. 211–240.

[81] Claudia LEACOCK, George A. MILLER et Martin CHODOROW. «Using Corpus Statistics and WordNet Relations for Sense Identification». In : *Comput. Linguist.* 24.1 (mar. 1998), p. 147–165. ISSN : 0891-2017. URL : http://dl.acm.org/citation.cfm?id=972719.972726.

[82] Georges LEBBOSS. «Contribution à l'analyse sémantique des textes arabes». Thèse de doct. Université de Paris 8, 2016.

[83] Georges LEBBOSS, Gilles BERNARD, Nourredine ALIANE et Mohammad HAJJAR. «Towards the Enrichment of Arabic WordNet with Big Corpora». In : *Proceedings of the 9th International Joint Conference on Computational Intelligence, IJCCI 2017, Funchal, Madeira, Portugal, November 1-3, 2017.* 2017, p. 101–109. DOI : 10.5220/0006505701010109. URL : https://doi.org/10.5220/0006505701010109.

[84] Rémi LEBRET et Ronan COLLOBERT. «Word Embeddings through Hellinger PCA.» In : *EACL*. Sous la dir. de Gosse BOUMA et Yannick PARMENTIER. The Association for Computer Linguistics, 2014, p. 482–490. ISBN : 978-1-937284-78-7. URL : http://www.aclweb.org/anthology/E14-1051.

[85] Daniel D. LEE et H. Sebastian SEUNG. «Algorithms for Non-negative Matrix Factorization». In : *Advances in Neural Information Processing Systems 13*. Sous la dir. de T. K. LEEN, T. G. DIETTERICH et V. TRESP. MIT Press, 2001, p. 556–562. URL : http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf.

[86] Omer LEVY et Yoav GOLDBERG. «Dependency-Based Word Embeddings.» In : *ACL (2)*. 2014, p. 302–308.

[87] Omer LEVY, Yoav GOLDBERG et Ido DAGAN. «Improving Distributional Similarity with Lessons Learned from Word Embeddings». In : *TACL* 3 (2015), p. 211–225.

[88] Jiwei Li et Dan Jurafsky. «Do Multi-Sense Embeddings Improve Natural Language Understanding ?» In : *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal : Association for Computational Linguistics, 2015, p. 1722–1732. doi : 10.18653/v1/D15-1200. url : http://aclweb.org/anthology/D15-1200.

[89] Ping Li, Trevor J. Hastie et Kenneth W. Church. «Very sparse random projections». In : *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. Exported from https ://app.dimensions.ai on 2018/12/23. 2006, p. 287–296. doi : 10.1145/1150402.1150436. url : https://app.dimensions.ai/details/publication/pub.1028084929%20and%20http://www-stat.stanford.edu/~hastie/Papers/Ping/KDD06_rp.pdf.

[90] Georges Liidi. «Métaphore et travail lexical'». In : *DOCUMENT RESUME* . (1991), p. 17.

[91] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao et Junjie Wu. «Understanding of Internal Clustering Validation Measures». In : *Proceedings of the 2010 IEEE International Conference on Data Mining*. ICDM '10. Washington, DC, USA : IEEE Computer Society, 2010, p. 911–916. isbn : 978-0-7695-4256-0. doi : 10.1109/ICDM.2010.35. url : http://dx.doi.org/10.1109/ICDM.2010.35.

[92] S. Lloyd. «Least Squares Quantization in PCM». In : *IEEE Transactions on Information Theory*. 28.2 (sept. 1982), p. 129–137. issn : 0018-9448. doi : 10.1109/TIT.1982.1056489. url : http://dx.doi.org/10.1109/TIT.1982.1056489.

[93] Edward Loper et Steven Bird. «NLTK : The Natural Language Toolkit». In : *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia : Association for Computational Linguistics*. 2002.

[94] Ferrone Lorenzo et Zanzotto Fabio Massimo. «Symbolic, Distributed and Distributional Representations for Natural Language Processing in the Era of Deep Learning : a Survey». In : *CoRR* abs/1702.00764 (2017). arXiv : 1702.00764. url : http://arxiv.org/abs/1702.00764.

[95] Kevin Lund, Curt Burgess et Ruth Ann Atchley. «Semantic and associative priming in high-dimensional semantic space». In : *Proceedings of the 17th annual conference of the Cognitive Science Society*. T. 17. 1995, p. 660–665.

[96] Thang Luong, Richard Socher et Christopher Manning. «Better Word Representations with Recursive Neural Networks for Morphology». In : *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria : Association for Computational Linguistics, 2013, p. 104–113. URL : http://aclweb.org/anthology/W13-3512.

[97] Otman Manad. «Nettoyage de corpus web pour le traitement automatique des langues». Thèse de doct. Université de Paris 8, 2018.

[98] Otman Manad, Nourredine Aliane et Gilles Bernard. «Un protocole d'expérimentation sur les propriétés graphémiques avec l'algorithme SOM». In : *16ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2016, 18-22 Janvier 2016, Reims, France*. 2016, p. 105–110.

[99] Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze. *Introduction to Information Retrieval*. New York, NY, USA : Cambridge University Press, 2008. ISBN : 0521865719, 9780521865715.

[100] Jean-Jacques Mariage. «De l'auto-organisation vers l'auto-observation». Thèse de doct. Université de Paris 8, 2001.

[101] Thomas M. Martinetz et Klaus J. Schulten. «A "Neural Gas" Network Learns Topologies». In : *Proceedings of the International Conference on Artificial Neural Networks 1991* (Espoo, Finland). Sous la dir. de Teuvo Kohonen, Kai Mäkisara, Olli Simula et Jari Kangas. Amsterdam ; New York : North-Holland, 1991, p. 397–402.

[102] Allan L McCutcheon. *Latent class analysis*. Quantitative Applications in the Social Sciences 64. The University Paper, London : Sage, 1987.

[103] Danielle McNamara. «Computational methods to extract meaning from text and advance theories of human cognition». In : *Topics in Cognitive Science* 3.1 (jan. 2011), p. 3–17. ISSN : 1756-8757.

[104] J. McQueen. «Some methods for classification and analysis of multivariate observations». In : *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, p. 281–297.

[105] Josué Melka et Gilles Bernard. «Jmp8 at SemEval-2017 Task 2: A simple and general distributional approach to estimate word similarity». In : *Proceedings of the 11th International Workshop on Semantic Evaluation*. 2017.

[106] Josué Melka et Jean-Jacques Mariage. «Efficient Implementation of Self-Organizing Map for Sparse Input Data». In : *Proceedings of the 9th International Joint Conference on Computational Intelligence, IJCCI 2017, Funchal, Madeira, Portugal, November 1-3, 2017*. 2017, p. 54–63. DOI : 10.5220/0006499500540063. URL : https://doi.org/10.5220/0006499500540063.

[107] Risto Miikkulainen. «Script Recognition With Hierarchical Feature Maps». In : *Connection Science* 2 (1990), p. 83–101. URL : http://nn.cs.utexas.edu/?miikkulainen:connsci90.

[108] Tomas Mikolov, Kai Chen, Greg Corrado et Jeffrey Dean. «Efficient Estimation of Word Representations in Vector Space». In : *Proceedings of the International Conference on Learning Representation, Workshop Track*. (Arizona). 2013, p. 1301.

[109] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Dean et Jeffrey. «Distributed representations of words and phrases and their compositionality». In : *Advances in Neural Information Processing Systems*. 2013, p. 3111–3119.

[110] Tomas Mikolov, Wen Tau Yih et Geoffrey Zweig. «Linguistic regularities in continuous space word representations». In : *HLT-NAACL*. 2013, p. 746–751.

[111] George A. Miller. «WordNet : A Lexical Database for English». In : *Commun. ACM* 38.11 (nov. 1995), p. 39–41.

[112] George A Miller et Walter G Charles. «Contextual correlates of semantic similarity». In : *Language Cognitive Processes* 6.1 (1991), p. 1–28. URL : http://eric.ed.gov/ERICWebPortal/recordDetail?accno=EJ431389.

[113] Jacques Moeschler. *Sémantique lexicale*. Rapp. tech. Université de Genève, 2013. URL : https://sites.google.com/site/moeschlerjacques/home.

[114] F. Morin et Y. Bengio. «Hierarchical probabilistic neural network language model». In : *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. Citeseer. 2005, p. 246–252.

[115] Filip Mulier et Vladimir Cherkassky. «Self-organization as an iterative kernel smoothing process». In : *Neural computation* 7.6 (1995), p. 1165–1177.

[116] Roberto Navigli et Simone Paolo Ponzetto. «BabelNet : The automatic construction, evaluation and application of a wide-coverage multilingual semantic network». In : *Artificial Intelligence* 193 (2012), p. 217–250. ISSN : 0004-3702. DOI : https://doi.org/10.1016/j.artint.2012.07.001. URL : http://www.sciencedirect.com/science/article/pii/S0004370212000793.

[117] Christopher J. O'Malley, Gary A. Montague, Elaine B. Martin, John M. Liddell, Bo Kara et Nigel J. Titchener-Hooker. «Utilisation of key descriptors from protein sequence data to aid bioprocess route selection». In : *Food and Bioproducts Processing* 90.4 (2012), p. 755–761. issn : 0960-3085. doi : https://doi.org/10.1016/j.fbp.2012.01.005. url : http://www.sciencedirect.com/science/article/pii/S0960308512000065.

[118] K. Pearson. «On Lines and Planes of Closest Fit to Systems of Points in Space». In : *Philosophical Magazine* 2 (6 1901), p. 559–572.

[119] Ted Pedersen, Siddharth Patwardhan et Jason Michelizzi. «WordNet ::Similarity : Measuring the Relatedness of Concepts». In : *Demonstration Papers at HLT-NAACL 2004*. HLT-NAACL–Demonstrations '04. Stroudsburg, PA, USA : Association for Computational Linguistics, 2004, p. 38–41.

[120] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot et E. Duchesnay. «Scikit-learn : Machine Learning in Python». In : *Journal of Machine Learning Research* 12 (2011), p. 2825–2830.

[121] Jeffrey Pennington, Richard Socher et Christopher D. Manning. «GloVe : Global Vectors for Word Representation». In : *Empirical Methods in Natural Language Processing (EMNLP)*. T. 14. 2014, p. 1532–1543.

[122] Alain Polguère. *Lexicologie et sémantique lexicale : notions fondamentales*. Pum, 2003.

[123] R. Rada, H. Mili, E. Bicknell et M. Blettner. «Development and application of a metric on semantic nets». In : *IEEE Transactions on Systems, Man and Cybernetics*. 1989, p. 17–30.

[124] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich et Shaul Markovitch. «A Word at a Time : Computing Word Relatedness Using Temporal Semantic Analysis». In : *Proceedings of the 20th International Conference on World Wide Web*. WWW '11. Hyderabad, India : ACM, 2011, p. 337–346. isbn : 978-1-4503-0632-4. doi : 10.1145/1963405.1963455. url : http://doi.acm.org/10.1145/1963405.1963455.

[125] William M. Rand. «Objective Criteria for the Evaluation of Clustering Methods». In : *Journal of the American Statistical Association* 66.336 (1971), p. 846–850. doi : 10.1080/01621459.1971.10482356. eprint : https://www.tandfonline.com/doi/pdf/10.1080/01621459.1971.10482356.

URL : https://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356.

[126] Lebret Rémi et Lebret Ronan. «Word Emdeddings through Hellinger PCA». In : *CoRR* abs/1312.5542 (2013). arXiv : 1312.5542. URL : http://arxiv.org/abs/1312.5542.

[127] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979. ISBN : 0-408-70929-4.

[128] Horacio Rodríguez, David Farwell, Javi Farreres, Manuel Bertran, Musa Alkhalifa et Antonia Martí. «Arabic WordNet Semi-automatic Extensions using Bayesian Inference». In : *LREC*. Marrakech (Morocco), 2008.

[129] Xin Rong. «Word2Vec Parameter Learning Explained». In : *CoRR* (2014). URL : http://arxiv.org/abs/1411.2738.

[130] Herbert Rubenstein et John B. Goodenough. «Contextual Correlates of Synonymy». In : *Commun. ACM* 8.10 (oct. 1965), p. 627–633. ISSN : 0001-0782. DOI : 10.1145/365628.365657. URL : http://doi.acm.org/10.1145/365628.365657.

[131] Benoît Sagot. «Représentation de l'information sémantique lexicale : le modèle wordnet et son application au français». In : *Revue Française de Linguistique Appliquée* XXII (2017). URL : https://hal.inria.fr/hal-01583995.

[132] Benoît Sagot et Darja Fišer. «Building a free French wordnet from multilingual resources». In : *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. Sous la dir. d'European Language Resources Association (ELRA). Marrakech, Morocco, 2008.

[133] Benoît Sagot et Darja Fišer. «Classification-Based Extension of Wordnets from Heterogeneous Resources». In : *Human Language Technology Challenges for Computer Science and Linguistics - 5th Language and Technology Conference, LTC 2011, Poznań, Poland, November 25-27, 2011, Revised Selected Papers*. 2011, p. 396–407. DOI : 10.1007/978-3-319-08958-4_32. URL : https://doi.org/10.1007/978-3-319-08958-4%5C_32.

[134] Benoît Sagot et Darja Fišer. «Cleaning noisy wordnets». In : *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*. 2012, p. 3468–3472. URL : http://www.lrec-conf.org/proceedings/lrec2012/summaries/1127.html.

[135] Franck Sajous et Nabil Hathout. «GLAWI, a free XML-encoded Machine-Readable Dictionary built from the French Wiktionary». English. In : *Proceedings of the of the eLex 2015 conference*. Herstmonceux, England, août 2015, p. 405–426.

[136] Gerard Salton, Anita Wong et Chung-Shu Yang. «A vector space model for automatic indexing». In : *Communications of the ACM* 18.11 (1975), p. 613–620.

[137] Boris Schling. *The Boost C++ Libraries*. XML Press, 2011. isbn : 0982219199, 9780982219195.

[138] Tobias Schnabel, Igor Labutov, David M. Mimno et Thorsten Joachims. «Evaluation methods for unsupervised word embeddings.» In : *EMNLP*. Sous la dir. de Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin et Yuval Marton. The Association for Computational Linguistics, 2015, p. 298–307. url : http://dblp.uni-trier.de/db/conf/emnlp/emnlp2015.html#SchnabelLMJ15.

[139] Cyrus Shaoul et Chris Westbury. «Word frequency effects in high-dimensional co-occurrence models : A new approach». In : *Behavior research methods* 38 (juin 2006), p. 190–5. doi : 10.3758/BF03192768.

[140] Thabet Slimani. «Description and Evaluation of Semantic Similarity Measures Approaches». In : *CoRR* abs/1310.8059 (2013). arXiv : 1310.8059. url : http://arxiv.org/abs/1310.8059.

[141] C. Spearman. «The Proof and Measurement of Association Between Two Things». In : *American Journal of Psychology* 15 (1904), p. 88–103.

[142] Suraj Subramanian et Deepali Vora. «Unsupervised Text Classification and Search using Word Embeddings on a Self-Organizing Map». In : *International Journal of Computer Applications* (2016).

[143] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample et Chris Dyer. «Evaluation of Word Vector Representations by Subspace Alignment.» In : *EMNLP*. Sous la dir. de Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin et Yuval Marton. The Association for Computational Linguistics, 2015, p. 2049–2054. isbn : 978-1-941643-32-7. url : http://dblp.uni-trier.de/db/conf/emnlp/emnlp2015.html#TsvetkovFLLD15.

[144] Joseph Turian, Lev Ratinov et Yoshua Bengio. «Word Representations : A Simple and General Method for Semi-supervised Learning». In : *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden : Association for Computational

Linguistics, 2010, p. 384–394. URL : http://dl.acm.org/citation.cfm?id=1858681.1858721.

[145] Peter D. TURNEY. «Similarity of Semantic Relations». In : *Comput. Linguist.* 32.3 (sept. 2006), p. 379–416.

[146] Peter D. TURNEY et Patrick PANTEL. «From Frequency to Meaning : Vector Space Models of Semantics». In : *CoRR* abs/1003.1141 (2010). arXiv : 1003.1141. URL : http://arxiv.org/abs/1003.1141.

[147] A. ULTSCH et H. P. SIEMON. «Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis». In : *Proceedings of International Neural Networks Conference (INNC)*. Paris : Kluwer Academic Press, 1990, p. 305–308. URL : http://www.uni-marburg.de/fb12/datenbionik/pdf/pubs/1990/UltschSiemon90.

[148] Jim VAN OVERSCHELDE, Katherine RAWSON et John DUNLOSKY. «Category norms : An updated and expanded version of the Battig and Montague (1969) norms». In : *Journal of Memory and Language - J MEM LANG* 50 (avr. 2004), p. 289–335. DOI : 10.1016/j.jml.2003.10.003.

[149] Piek VOSSEN, éd. *EuroWordNet : A Multilingual Database with Lexical Semantic Networks*. Norwell, MA, USA : Kluwer Academic Publishers, 1998. ISBN : 0-7923-5295-5.

[150] Zhibiao WU et Martha PALMER. «Verbs Semantics and Lexical Selection». In : *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*. ACL '94. Las Cruces, New Mexico : Association for Computational Linguistics, 1994, p. 133–138.

[151] Youssef ZAKI. «Contribution à l'analyse syntaxique de phrase en arabe standard moderne». Thèse de doct. Université de Paris 8, 2019.

Table des figures

Liste des tableaux

Table des matières