Construire des représentations denses à partir de thésaurus distributionnels

Olivier Ferret
CEA, LIST, Laboratoire Vision et Ingénierie des Contenus, Gif-sur-Yvette, F-91191 France.
olivier.ferret@cea.fr

RÉSUMÉ

Dans cet article, nous nous intéressons à un nouveau problème, appelé plongement de thésaurus, consistant à transformer un thésaurus distributionnel en une représentation dense de mots. Nous proposons de traiter ce problème par une méthode fondée sur l'association d'un plongement de graphe et de l'injection de relations dans des représentations denses. Nous avons appliqué et évalué cette méthode pour un large ensemble de noms en anglais et montré que les représentations denses produites obtiennent de meilleures performances, selon une évaluation intrinsèque, que les représentations denses construites selon les méthodes de l'état de l'art sur le même corpus. Nous illustrons aussi l'intérêt de la méthode développée pour améliorer les représentations denses existantes à la fois de façon endogène et exogène.

ABSTRACT

Distributional Thesaurus Embedding and its Applications

In this article, we propose considering a new problem, called thesaurus embedding and consisting in turning a distributional thesaurus into word embeddings. We propose more precisely a method for performing such task based on the association of graph embedding and knowledge injection into distributed representations. We have applied and evaluated it at a large scale for English nouns and showed that the resulting embeddings outperform state-of-the-art embeddings built from the same corpus. We also illustrate the application of the developed method for improving already existing word embeddings both in endogenous and exogenous ways.

MOTS-CLÉS : Sémantique distributionnelle, thésaurus, plongements lexicaux.

KEYWORDS: Distributional semantics, thesaurus, word embeddings.

1 Introduction

Dans le cadre du Traitement Automatique des Langues (TAL), beaucoup des travaux menés initialement (Grefenstette, 1994; Lin, 1998; Curran & Moens, 2002) dans le domaine de la sémantique distributionnelle se sont focalisés sur la notion de thésaurus distributionnel. Les travaux plus récents dans ce domaine se sont davantage concentrés sur les notions de similarité et de proximité sémantiques (Budanitsky & Hirst, 2006) ainsi que sur la représentation des données distributionnelles. Cette tendance s'est renforcée encore plus récemment avec le courant des travaux sur les représentations distribuées et les plongements lexicaux (*word embeddings*), construits par des réseaux de neurones (Mikolov *et al.*, 2013) ou par d'autres moyens (Pennington *et al.*, 2014).

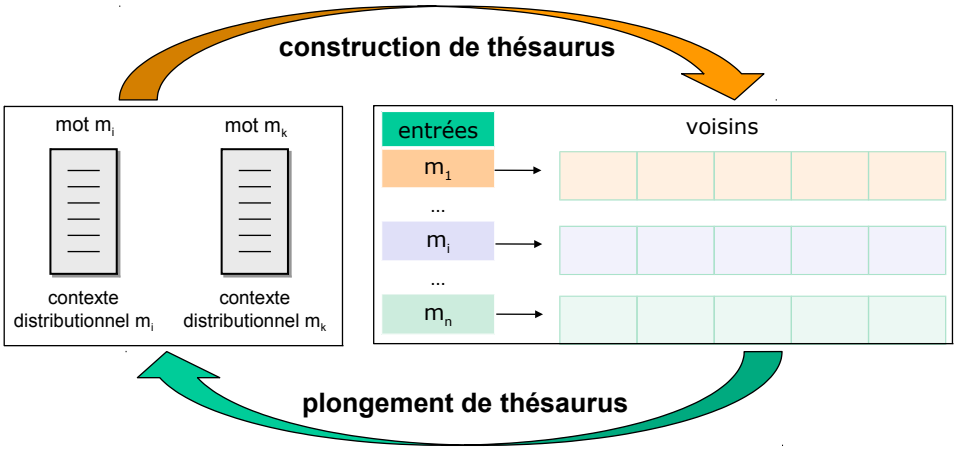

FIGURE 1: Dualité de l'information de similarité sémantique

D'un point de vue plus général, les thésaurus distributionnels et les données distributionnelles, c'est-à-dire les contextes distributionnels des mots, peuvent être considérés comme des représentations duales de la même information de similarité sémantique, ainsi que l'illustre la figure 1. Les données distributionnelles en constituent une forme intensionnelle tandis que les thésaurus distributionnels en sont la forme extensionnelle, obtenue par l'application d'une mesure de similarité aux données distributionnelles. Passer d'une représentation intensionnelle à une représentation extensionnelle correspond à l'opération classique sous-tendant la construction des thésaurus distributionnels. Dans le contexte des plongements lexicaux, Perozzi *et al.* (2014a) généralisent cette opération à la construction de réseaux lexicaux.

Le passage inverse d'une représentation extensionnelle à une représentation intensionnelle est, à notre connaissance, un problème nouveau dans le contexte de l'analyse distributionnelle. L'intérêt de cette transformation est double. En premier lieu, quelle que soit sa forme originelle, une connaissance sémantique peut ainsi être convertie sous la forme la plus adaptée à son contexte d'usage spécifique. Par exemple, la forme du thésaurus est utilisable facilement pour des tâches comme l'expansion de requêtes tandis que celle des plongements lexicaux est plus adaptée à une utilisation comme traits dans des classifieurs statistiques.

En second lieu, chaque forme est également associée à des méthodes spécifiques d'amélioration. Beaucoup de travaux ont ainsi été menés pour améliorer les contextes distributionnels par l'étude de différents paramètres, ce qui a conduit à une amélioration importante des thésaurus distributionnels. À l'inverse, des travaux tels que (Ferret, 2013; Claveau *et al.*, 2014) se sont focalisés sur l'amélioration des thésaurus en tant que tels, en s'appuyant sur leurs caractéristiques propres. Il serait donc intéressant de pouvoir transposer ces améliorations au niveau des contextes distributionnels selon une transformation inverse de la construction des thésaurus, telle qu'illustrée par la figure 1.

Dans cet article, nous proposons d'étudier une telle transformation, appelée plongement de thésaurus, conduisant à transformer un thésaurus distributionnel en plongements lexicaux. Nous montrerons en particulier qu'une telle transformation peut-être réalisée sans perte importante d'information et que les idées sous-jacentes à ce processus peuvent en outre être réutilisées pour l'amélioration endogène de plongements lexicaux déjà existants. Enfin, nous illustrerons l'intérêt de ce processus pour construire plus efficacement des plongements lexicaux incorporant des connaissances externes.

2 Plongement d'un thésaurus distributionnel

Un thésaurus distributionnel est généralement vu comme un ensemble d'entrées avec, pour chaque entrée, une liste de voisins sémantiques ordonnés selon l'ordre décroissant de leur similarité sémantique avec cette entrée. Les voisins d'une entrée étant aussi des entrées du thésaurus, celui-ci peut donc être représenté comme un graphe dont les sommets sont les mots du thésaurus, qu'ils soient entrées ou voisins, et les arêtes sont les relations de voisinage sémantique entre ces mots, pondérées en fonction de leur niveau de similarité sémantique. Ce graphe est non dirigé lorsque la mesure de similarité sémantique est symétrique, ce qui est le cas que nous considérons ici. Si le mot m_2 est un voisin de l'entrée m_1 et que m_3 est un voisin de m_2, le graphe comportera ainsi une arête entre m_1 et m_2 et une arête entre m_2 et m_3, liant ainsi indirectement m_1 et m_3. Si m_3 est un voisin de m_1, une arête directe existera aussi entre m_1 et m_3. La notion explicite d'ordre parmi les voisins d'une entrée du thésaurus n'est en revanche pas représentée de façon explicite. Une telle représentation a par exemple été adoptée par Claveau *et al.* (2014) pour améliorer un thésaurus en réordonnançant les voisins de ses entrées.

Une des spécificités des thésaurus distributionnels dans ce contexte est que même si le poids entre deux mots est corrélé à leur véritable similarité sémantique, des travaux tels que (Ferret, 2010; Claveau *et al.*, 2014) ont montré que la pertinence sémantique des voisins d'un mot décroît très rapidement à mesure que le rang de ces voisins augmente. Pour tenir compte de cette particularité, notre stratégie de plongement des thésaurus distributionnels s'articule autour de deux étapes : une première étape construit un plongement en s'appuyant sur la structure de graphe sous-jacente à ces thésaurus tandis qu'une seconde étape adapte ce premier plongement afin de tenir compte de leurs spécificités en termes de similarité sémantique. Nous détaillons ces deux étapes dans les deux sections suivantes.

2.1 Plongement de graphe

Le problème du plongement de graphe dans une perspective de réduction de dimensions [1] consiste à associer à chaque sommet d'un graphe une représentation vectorielle telle que les proximités entre les vecteurs ainsi produits soient représentatives des proximités dans le graphe des sommets auxquels ils sont associés. Dans notre cas, les sommets étant des mots, les vecteurs construits sont des plongements lexicaux. Ce problème n'est pas nouveau et a déjà fait l'objet de nombreux travaux (Yan *et al.*, 2007), allant des méthodes spectrales (Belkin & Niyogi, 2001) aux méthodes plus récentes de nature neuronale (Perozzi *et al.*, 2014b; Cao *et al.*, 2016; Grover & Leskovec, 2016).

Les graphes pouvant être représentés par leur matrice d'adjacence, ce problème est aussi fortement associé à celui de la factorisation de matrices. La stratégie de base en la matière est de réaliser une décomposition de la matrice en éléments propres, à l'instar de l'Analyse Sémantique Latente (LSA) (Landauer & Dumais, 1997). Néanmoins, une telle décomposition est coûteuse d'un point de vue calculatoire et pour des matrices de taille importante, comme dans le contexte du filtrage collaboratif (Koren, 2008), des techniques de factorisation de matrice moins contraintes ont été développées.

Pour notre première étape de transformation d'un thésaurus distributionnel en plongements lexicaux, nous avons donc testé les trois méthodes suivantes :

1. Il est à noter que la notion de plongement de graphe renvoie aussi à d'autres problématiques en théorie des graphes qui sont en dehors de notre champ d'étude.

— l'algorithme LINE (Tang *et al.*, 2015), une méthode récente pour le plongement de graphes pondérés ;
— la décomposition en valeurs singulières (SVD) de la matrice d'adjacence du thésaurus ;
— l'application de la factorisation de matrice (FM) proposée par Hu *et al.* (2008) à la matrice d'adjacence du thésaurus.

LINE définit un modèle probabiliste sur l'espace $V \times V$, avec V, l'ensemble des sommets du graphe considéré, afin de rendre compte de la probabilité jointe de deux sommets. Ce modèle est fondé sur la représentation de chaque sommet sous la forme d'un vecteur de faible dimension, vecteur qui est le résultat du plongement. Ce vecteur est obtenu par la minimisation d'une fonction objectif prenant la forme de la divergence de Kullback-Leibler entre le modèle probabiliste et la distribution empirique observée dans le graphe considéré. Cette minimisation est réalisée par une descente de gradient stochastique (SGD). Tang *et al.* (2015) proposent plus précisément deux modèles : l'un s'appuie sur les relations directes entre les sommets tandis que le second définit la proximité de deux sommets en fonction du nombre de voisins qu'ils partagent. Nous avons adopté le second modèle, qui donne globalement de meilleurs résultats pour plusieurs évaluations.

Dans notre deuxième option, la SVD factorise la matrice d'adjacence T du thésaurus initial en un produit $U \cdot \Sigma \cdot V^\intercal$ où U et V sont orthonormales et Σ est la matrice diagonale des valeurs propres. Nous avons classiquement adopté la version tronquée de la SVD en ne retenant que les d premiers éléments de Σ. Le produit résultant a donc pour résultat une version approximée T_d de la matrice d'adjacence originelle telle que $T_d = U_d \cdot \Sigma_d \cdot V_d^\intercal$. Au final, les pratiques habituelles, en particulier issues de la LSA, conduisent une représentation des mots donnée par le produit $U_d \cdot \Sigma_d$. Néanmoins, Caron (2001) suggère que $U_d \cdot \Sigma_d^P$ avec $P < 1$ est une meilleure option. Levy *et al.* (2015) ont étudié ce point dans le contexte des matrices de cooccurrences lexicales et trouvé que $P = 0$ ou $P = 0,5$ sont clairement de meilleures options que $P = 1$, avec un léger avantage à $P = 0$. De façon analogue, nous avons trouvé que l'option $P = 0$ est la plus intéressante dans notre contexte.

Notre dernier choix est fondé sur une forme moins contrainte de factorisation de matrice dans laquelle T est décomposée en deux matrices telles que $U \cdot V = \hat{T} \approx T$, avec $T \in \mathbb{R}^{m \cdot n}$, $U \in \mathbb{R}^{m \cdot d}$, $V \in \mathbb{R}^{d \cdot n}$ et $d \ll m, n$. U et V sont obtenues en minimisant la fonction objectif suivante :

$$\sum_{i,j} (t_{ij} - u_i^\intercal v_j)^2 + \lambda(\|u_i\|^2 + \|v_j\|^2) \tag{1}$$

où le premier terme minimise l'erreur de reconstruction de T par le produit $U \cdot V$ tandis que le second terme de régularisation, contrôlé par le paramètre λ, permet d'éviter une forme de sur-adaptation de cette reconstruction aux données particulières considérées. Nous utilisons U comme plongement du thésaurus initial. L'approche que nous avons adoptée pour obtenir U est plus précisément celle de Hu *et al.* (2008), une variante dans laquelle les termes t_{ij} sont transformés en scores de confiance et la minimisation de l'expression 1 est obtenue selon la méthode *Alternating Least Squares*. Un des intérêts de cette approche de factorisation de matrice est sa capacité à gérer les valeurs non définies. Dans le contexte des systèmes de recommandation, cette capacité correspond à une forme de retour de pertinence implicite. Dans notre contexte, elle offre la possibilité de compenser la relative faiblesse de la densité de notre graphe d'entrée, qui n'inclut pas les voisins sémantiques les plus éloignés d'une entrée.

2.2 Amélioration endogène des plongements de thésaurus

Comme nous l'avions indiqué en préambule, la représentation d'un thésaurus distributionnel par une structure de graphe telle que nous l'avons envisagée à la section précédente ne tient pas compte d'un fait important : la valeur de similarité sémantique entre une entrée de ce thésaurus et ses voisins n'est pas linéairement corrélée avec leur rang du point de vue de la pertinence sémantique effective de ces voisins. Plus précisément, cette pertinence sémantique décroît très rapidement à mesure que le rang des voisins augmente, ce qui donne une importance prédominante aux premiers voisins.

Pour prendre en compte cette observation, nous avons adopté une stratégie endogène consistant à utiliser les premiers voisins de chaque entrée du thésaurus initial comme contraintes pour adapter le plongement de ce thésaurus construit grâce aux méthodes de plongement de graphe que nous avons présentées à la section précédente. De façon générale, ce type d'adaptation consiste à modifier les valeurs des vecteurs correspondant aux plongements lexicaux en fonction d'un objectif spécifique touchant aux relations de proximité entre ces vecteurs. Cette problématique a déjà fait l'objet de travaux dans le contexte de l'injection de connaissances externes, comme des relations sémantiques, dans des plongements lexicaux construits par des méthodes neuronales. Dans ce cadre, l'objectif est de modifier les vecteurs considérés de telle manière que des vecteurs représentant des mots liés par une relation explicite de proximité sémantique soient rapprochés. À l'inverse, des vecteurs peuvent être éloignés s'ils sont associés à des mots en relation d'antagonisme. Dans notre cas, l'objectif est d'utiliser les relations de voisinage entre les entrées d'un thésaurus et leurs tout premiers voisins pour renforcer leur influence dans les plongements produits, en adéquation avec leur plus grande qualité en termes de similarité sémantique.

Les méthodes pour réaliser une telle adaptation des plongements lexicaux peuvent schématiquement être divisées en deux catégories : celles opérant lors de la construction même des plongements, généralement par la modification de la fonction objectif régissant cette construction (Yih *et al.*, 2012; Zhang *et al.*, 2014), et celles appliquées après la production des plongements (Yu & Dredze, 2014; Xu *et al.*, 2014). Nous nous sommes plus particulièrement concentrés sur l'utilisation ou l'adaptation de méthodes de la seconde catégorie mais nous avons adapté aussi une méthode de la première catégorie à notre stratégie endogène.

La première méthode que nous avons considérée est la méthode dite de *retrofit* de Faruqui *et al.* (2015), qui adapte un ensemble de vecteurs lexicaux denses q_i en minimisant la fonction objectif suivante par le biais d'un algorithme de propagation d'étiquettes (Bengio *et al.*, 2006) :

$$\sum_{i=1}^{n} \left[\|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \|q_i - q_j\|^2 \right] \quad (2)$$

où $\hat{q}_i$ sont les vecteurs q_i à la suite de leur adaptation. Le premier terme garantit une forme de stabilité en assurant que les vecteurs adaptés ne s'écartent pas trop des vecteurs initiaux tandis que le second terme met en œuvre l'adaptation proprement dite en rapprochant les vecteurs q_i et q_j des mots i et j impliqués dans une relation faisant partie des connaissances externes E. Dans notre cas, ces connaissances correspondent aux relations entre chaque entrée du thésaurus initial et ses premiers voisins.

La deuxième méthode que nous avons utilisée, *counterfit* (Mrkšić *et al.*, 2016), se différencie de *retrofit* essentiellement par l'ajout à la fonction objectif d'un terme répulsif permettant d'écarter les vecteurs des mots faisant partie de relations de dissimilarité, elles aussi données *a priori*. Néanmoins, les thésaurus distributionnels ne contiennent pas de relations de ce type puisqu'ils sont fondés au

contraire sur une notion de similarité sémantique [2]. Nous avons donc éliminé ce facteur et obtenu la fonction objectif suivante, minimisée par SGD :

$$\sum_{i=1}^{N} \sum_{j \in N(i)} \tau(dist(\hat{q}_i, \hat{q}_j) - dist(q_i, q_j)) + \sum_{(i,j) \in E} \tau(dist(\hat{q}_i, \hat{q}_j)) \quad (3)$$

avec $dist(x, y) = 1 - \cos(x, y)$ et $\tau(x) = \max(0, x)$, la fonction de perte hinge. Comme dans l'équation 2, le premier terme tend à préserver les vecteurs initiaux. Dans ce cas, cette préservation ne se focalise pas sur les vecteurs mêmes mais sur les distances entre un vecteur et ses plus proches voisins $N(i)$. Le second terme est très similaire au second terme de l'équation 2, avec l'utilisation de la distance dérivée de la mesure Cosinus au lieu de la distance euclidienne [3].

La dernière méthode que nous avons testée pour l'amélioration des plongements de thésaurus, que nous appellerons *rankfit* dans ce qui suit, est une transposition de la méthode proposée par Liu *et al.* (2015). L'objectif de cette dernière est d'intégrer au sein de plongements lexicaux des contraintes d'ordre issues de relations externes prenant la forme : similarité$(w_i, w_j) >$ similarité(w_i, w_k), abrégée dans ce qui suit par $s_{ij} > s_{ik}$. Cette approche est particulièrement intéressante dans notre contexte puisque les voisins d'une entrée d'un thésaurus distributionnel sont ordonnés et peuvent donc être vus comme un ensemble de contraintes de ce type. Plus précisément, i correspond dans notre cas à une entrée de thésaurus tandis que j et k sont deux de ses voisins tels que *rang(j)* $>$ *rang(k)* [4]. Néanmoins, Liu *et al.* (2015) ont intégré ces contraintes en modifiant la fonction objectif du modèle Skip-Gram de Mikolov *et al.* (2013), ce qui n'est pas adapté à notre contexte. Nous avons donc transposé les principes de Liu *et al.* (2015) pour une modification des plongements après leur construction, détachée du modèle Skip-Gram. L'idée générale est d'adapter les vecteurs afin de minimiser $s_{ij} - s_{ik} \quad \forall (i, j, k) \in E$. L'objectif à minimiser s'exprime plus spécifiquement par :

$$\sum_{(i,j,k) \in E} \tau(s_{ik} - s_{ij}) \quad (4)$$

où $\tau(s_{ik} - s_{ij}) = \max(0, s_{ik} - s_{ij})$ correspond là aussi à la fonction de perte hinge et la similarité entre les mots i et j, s_{ij} est donnée par l'application de la mesure Cosinus à leurs vecteurs associés. La minimisation de cet objectif est réalisée par SGD, comme dans le cas de *counterfit*.

counterfit et *rankfit* étant proches en termes d'implémentation, nous avons également défini une méthode hybride, *counter-rankfit*, qui associe les contraintes de proximité des mots et de leur ordonnancement relatif. Cette association est réalisée par la conjugaison des fonctions objectif de *counterfit* et *rankfit* au travers de l'addition du second terme de l'équation 3, son terme d'adaptation, à l'équation 4. Dans cette configuration, la conservation du premier terme de la fonction de *counterfit*, centré sur la préservation des plongements initiaux, n'a pas été jugée utile à l'issue d'expérimentations préliminaires.

2. Nous avons testé l'utilisation des relations entre les entrées d'un thésaurus et leurs voisins distants comme relations de dissimilarité mais les résultats obtenus n'ont pas été probants.

3. La mesure Cosinus étant utilisée pour l'évaluation des similarités sémantiques entre mots par l'entremise de leurs vecteurs, cette distance devrait en principe être plus pertinente dans ce contexte que la distance euclidienne.

4. C'est-à-dire que k est un voisin plus proche de l'entrée i du thésaurus que son voisin j.

3 Expérimentations

3.1 Cadre expérimental

Pour évaluer l'approche proposée, nous avons construit un thésaurus distributionnel à partir d'un corpus de référence. Nous avons en l'occurrence choisi le corpus AQUAINT-2, déjà utilisé dans d'autres travaux (Ferret, 2010; Claveau *et al.*, 2014). Ce corpus de taille moyenne – environ 380 millions de mots – est constitué d'articles journalistiques en anglais auxquels nous avons appliqué une lemmatisation et une suppression des mots grammaticaux. Selon (Bullinaria & Levy, 2012), la lemmatisation permet d'obtenir de meilleurs résultats en termes de similarité sémantique. Elle permet aussi d'atteindre un niveau de résultats donné avec une quantité moindre de données.

La construction de notre thésaurus distributionnel de référence, Thés$_{cnt}$, a été réalisée selon une approche classique de type « comptage » en adoptant les paramètres faisant consensus dans plusieurs études systématiques récentes (Baroni *et al.*, 2014; Kiela & Clark, 2014; Levy *et al.*, 2015) :

— contextes distributionnels : cooccurrents restreints aux noms, verbes et adjectifs de fréquence supérieure à 10 et collectés dans une fenêtre de 3 mots (+/-1 mot autour du mot cible) ;
— cooccurrents directionnels, suite aux conclusions de Bullinaria & Levy (2012) ;
— fonction de pondération des cooccurrents dans les contextes = information mutuelle ramenée aux valeurs positives (PPMI : *Positive Pointwise Mutual Information*) en utilisant le facteur de lissage (*context distribution smoothing*) proposée par Levy *et al.* (2015), égal à 0,75 ;
— mesure de similarité entre contextes, pour évaluer la similarité sémantique de deux mots = mesure Cosinus ;
— filtrage des contextes : suppression des cooccurrents n'ayant qu'une seule occurrence.

La construction du thésaurus à partir de ces données distributionnelles a été réalisée classiquement (Lin, 1998; Curran & Moens, 2002) en extrayant les voisins sémantiques les plus proches pour chacune de ses entrées. Plus précisément, la mesure de similarité retenue a été appliquée entre les contextes distributionnels de chaque entrée et de ses voisins possibles, qui sont dans les deux cas les noms de fréquence supérieure à 10 dans le corpus de référence. Ces voisins sont finalement ordonnés selon l'ordre décroissant des valeurs de cette mesure.

L'évaluation d'objets distributionnels tels que des thésaurus ou des plongements lexicaux est actuellement un sujet de recherche actif dans la mesure où les tests de similarité sémantique utilisant des données telles que WordSim-353 (Gabrilovich & Markovitch, 2007) ou plus récemment SimLex-999 (Hill *et al.*, 2015) ne sont pas sans montrer des insuffisances hypothéquant leur fiabilité, à commencer par leur couverture limitée (Batchkarov *et al.*, 2016). De ce fait, à l'instar de Ferret (2010), nous avons adopté le type d'évaluation intrinsèque pour la similarité sémantique précédemment réalisé par Curran & Moens (2002) en l'implémentant à large échelle avec deux ressources de référence, complémentaires en termes de types de relations : d'une part, les synonymes de WordNet 3.0 (Miller, 1990) [W], caractérisant une similarité fondée sur des relations paradigmatiques et d'autre part, le thésaurus Moby (Ward, 1996) [M], rassemblant un ensemble plus large de types de relations, plus représentatives de la notion de proximité sémantique. Nous avons aussi considéré la fusion de ces deux ressources [W+M]. Pour nous concentrer sur l'évaluation des voisins sémantiques extraits, nous avons filtré ces ressources pour en supprimer les entrées et leurs mots associés ne faisant pas partie du vocabulaire du corpus AQUAINT-2. Le nombre d'entrées évaluées ainsi que le nombre moyen par entrée de mots associés dans ces ressources sont donnés par les $3^{ème}$ et $4^{ème}$ colonnes du tableau 1.

Référence	Méthode	#mots éval.	#syn./ mot	R@100	R-préc.	MAP	P@1	P@5	P@10
WordNet	Thés$_{cnt}$	10 544	2,9	29,0	11,3	13,1	15,7	6,6	4,3
	GloVe			18,4	5,7	6,7	8,4	3,8	2,5
	SGNS			22,4	8,7	10,3	12,3	5,2	3,3
Moby	Thés$_{cnt}$	9 269	49,9	11,5	8,8	4,5	32,6	21,6	16,7
	GloVe			7,8	5,5	2,5	21,4	14,4	11,1
	SGNS			6,0	4,9	2,3	20,6	12,7	9,5
W+M	Thés$_{cnt}$	12 326	38,6	11,9	10,7	8,1	30,9	18,6	14,1
	GloVe			7,9	5,6	3,7	18,7	11,9	9,0
	SGNS			6,5	6,7	5,3	20,7	11,5	8,3

TABLE 1: Évaluation du thésaurus initial et de deux modèles de référence

Du point de vue méthodologique, le type d'évaluation que nous avons réalisé reprend le paradigme de la recherche d'information en assimilant chaque entrée du thésaurus évalué à une requête et ses voisins sémantiques aux documents renvoyés. Nous avons donc adopté les mêmes mesures : la R-précision (R-préc.), précision au rang R, R correspondant au nombre de mots associés de référence ; la MAP (Mean Average Precision), moyenne des précisions pour chacun des rangs auxquels un mot associé de référence a été trouvé ; la précision aux rangs 1, 5 et 10. Nous donnons aussi le rappel global pour les 100 premiers voisins (R@100).

Le tableau 1 donne les résultats de l'évaluation, selon ces mesures, de notre thésaurus initial Thés$_{cnt}$ ainsi que l'évaluation dans le même cadre de deux modèles de référence pour la construction de plongements lexicaux à partir de textes : le modèle GloVe de Pennington *et al.* (2014) et le modèle Skip-Gram avec échantillonnage négatif (SGNS) de Mikolov *et al.* (2013) [5]. Ces deux modèles ont été construits comme Thés$_{cnt}$ à partir d'une version lemmatisée du corpus AQUAINT-2 mais en conservant tous les mots. Le tableau 2 montre que la variante SGNS$_{mp}$ de SGNS construite dans les mêmes conditions que Thés$_{cnt}$, avec les seuls mots pleins, obtient de moins bons résultats que SGNS, d'où le choix de conserver tous les mots, même si cette option n'est pas toujours la meilleure dans d'autres contextes (Tang *et al.*, 2016). Pour chacun des deux modèles de référence ont été retenus les valeurs de paramètres reconnues les meilleures selon les travaux antérieurs et testées sur ce corpus. Pour GloVe : vecteurs de 400 dimensions, taille de fenêtre = 5, addition des vecteurs des mots et des contextes, 100 itérations ; pour SGNS : vecteurs de 400 dimensions, taille de fenêtre = 5 [6], 10 exemples négatifs et valeur par défaut pour le sous-échantillonnage des mots les plus fréquents.

Cette évaluation met en avant deux grandes tendances. Tout d'abord, Thés$_{cnt}$ dépasse nettement GloVe et SGNS pour toutes nos références [7]. Cette supériorité d'une approche de type « comptage » par rapport à deux approches de type « prédiction » peut être vue pour une part comme contradictoire avec les conclusions de Levy *et al.* (2015). Notre analyse est que l'utilisation des cooccurrents directionnels, un paramètre rarement testé, explique en grande partie cette supériorité. Les performances de A2ST

5. Suivant les conclusions de Levy *et al.* (2015), SGNS a été préféré au modèle CBOW (Continuous Bag-Of-Word).

6. Pour la variante SGNS$_{mp}$, les résultats du tableau 2 sont obtenus avec une fenêtre de taille égale à 3 afin de s'adapter à la réduction du nombre de mots considérés dans les textes. Les performances de cette valeur de paramètre sont seulement très légèrement supérieures à celles d'une fenêtre de 5 mots.

7. La significativité statistique des différences a été évaluée grâce à un test de Wilcoxon pour échantillons appariés avec un seuil de significativité de 0,05.

Référence	Méthode	R@100	R-préc.	MAP	P@1	P@5	P@10
WordNet	A2ST	-4,4	-3,1	-3,3	-4,0	-1,5	-0,9
	SGNS$_{mp}$	-3,4	-1,2	-1,5	-1,8	-0,7	-0,5
Moby	A2ST	-2,0	-2,1	-1,3	-8,5	-5,2	-3,7
	SGNS$_{mp}$	-1,2	-0,9	-0,4	-3,3	-2,3	-1,8
W+M	A2ST	-2,1	-3,0	-2,5	-8,4	-4,5	-3,3
	SGNS$_{mp}$	-1,3	-1,2	-0,9	-3,4	-2,0	-1,5

TABLE 2: Évaluation de A2ST par rapport à Thés$_{cnt}$ et de SGNS$_{mp}$ par rapport à SGNS

(Ferret, 2010), un thésaurus strictement équivalent à Thés$_{cnt}$ dans sa construction en dehors de la directionnalité des cooccurrents, en apporte une illustration assez directe au travers du tableau 2. La seconde conclusion de cette évaluation est que SGNS tend à dépasser GloVe pour la référence W+M mais pas dans tous les cas. SGNS est largement supérieur à GloVe avec WordNet comme référence mais la tendance est dans une moindre mesure inversée avec Moby comme référence, suggérant ainsi que GloVe représente mieux la proximité sémantique et SGNS, la similarité sémantique. Même pour la référence W+M, le rappel global de GloVe est meilleur que celui de SGNS. GloVe est donc meilleur pour le rappel tandis que SGNS est supérieur pour classer les « bons » voisins sémantiques parmi les premiers voisins. Dans ce qui suit, nous ne donnerons que les résultats obtenus avec SGNS.

3.2 Évaluation du plongement de graphe

En adoptant le cadre présenté ci-dessus, nous avons évalué l'application des trois méthodes de plongement de graphe présentées à la section 2.1 pour le plongement de notre thésaurus initial Thés$_{cnt}$. Pour toutes ces méthodes, les principaux paramètres sont le nombre de voisins pris en compte ainsi que le nombre de dimensions des vecteurs. Dans tous les cas, la valeur du premier paramètre a été fixée à 5,000, LINE n'étant pas très influencée par ce paramètre, et la valeur du second à 600 [8]. Pour LINE, 10 milliards d'échantillons des valeurs de similarité ont été utilisés et pour la factorisation de matrice (FM), nous avions $\lambda = 0,075$.

Méthode	R-préc.	MAP	P@1	P@5	P@10
Thés$_{cnt}$	10,7	8,1	30,9	18,6	14,1
SGNS	6,7	5,3	20,7	11,5	8,3
SVD	**8,2**	**5,8**	**23,4**	**14,4**	**11,4**
LINE	7,4	5,3	20,8	12,8	10,0
FM	4,9	3,0	14,2	8,9	7,0

TABLE 3: Évaluation des méthodes de plongement de graphe (référence : W+M)

Le tableau 3 donne les résultats de l'évaluation des trois méthodes avec la référence W+M et montre que SVD apparaît clairement comme la meilleure méthode de plongement, même si LINE est une alternative compétitive. En outre, les résultats de SVD et LINE dépassent tous deux ceux de SGNS,

8. Ces paramètres ont été optimisés sur le thésaurus A2ST du tableau 2 issu de (Ferret, 2010).

plus sensiblement pour SVD, ce qui est un premier constat intéressant : cette première étape du processus de plongement des thésaurus produit des représentations déjà plus performantes que celles produites par une méthode reconnue de l'état de l'art. Néanmoins, le tableau 3 montre aussi que le niveau du thésaurus initial Thés$_{cnt}$ n'est pas encore atteint, justifiant donc la seconde étape de notre processus. Enfin, il faut noter que la factorisation de matrice est assez nettement une mauvaise option, au moins sous la forme testée.

3.3 Évaluation globale du plongement de thésaurus

Toujours dans le même cadre d'évaluation, le tableau 4 donne le résultat de l'évaluation de la totalité du processus de plongement de thésaurus pour les différentes méthodes de la seconde phase. Dans tous les cas, nous avons retenu pour la première phase le plongement du thésaurus initial Thés$_{cnt}$ produit par la méthode SVD, l'option très clairement la meilleure selon le tableau 3. Pour les méthodes *retrofit* et *counterfit*, seules les relations entre chaque entrée du thésaurus et ses deux premiers voisins ont été retenues. Pour *rankfit*, le voisinage a été étendu aux 50 premiers voisins. Pour les processus d'optimisation, nous avons adopté les valeurs par défaut : 10 itérations pour *retrofit* et 20 itérations pour *counterfit* ; de même 20 itérations pour *rankfit* et *counter-rankfit*. Pour toutes les optimisations par SGD, le taux d'apprentissage était égal à 0,01.

Méthode	**R-préc.**	**MAP**	**P@1**	**P@5**	**P@10**
Thés$_{cnt}$	10,7	8,1	30,9	18,6	14,1
SVD	8,2	5,8	23,4	14,4	11,4
retrofit	**10,0**	**7,9**	28,9	18,1	13,5
counterfit	9,5	7,6	26,8	**18,5**	**13,6**
rankfit	8,7	6,3	25,1	15,1	11,5
counter-rankfit	9,7	7,3	**29,4**	17,3	12,7

TABLE 4: Évaluation du processus global de plongement de thésaurus (référence : W+M)

Le tableau 4 montre en premier lieu que toutes les méthodes testées améliorent de façon significative le premier plongement. Il montre aussi que les différentes méthodes sont proches en termes de résultats. *retrofit*, qui se détache le plus, n'est ainsi nettement supérieure à *counterfit* que pour la R-précision et la précision au rang 1. *rankfit* est en revanche la moins bonne méthode sans ambiguïté et son association avec *counterfit* n'est intéressante que pour la précision au rang 1. Il faut plus globalement noter que l'association de SVD et de *retrofit* obtient des résultats très proches de ceux du thésaurus initial Thés$_{cnt}$, ce qui permet d'affirmer, en relation avec notre objectif premier, qu'il est possible de construire un plongement de thésaurus en conservant l'information sur la similarité sémantique des mots qu'il contient.

4 Amélioration endogène de plongements lexicaux

Dans le processus de construction du plongement d'un thésaurus, nous avons vu que les relations supposées les plus fiables d'un thésaurus distributionnel peuvent être utilisées pour améliorer un plongement lexical construit à partir de ce thésaurus. Cette adaptation étant réalisée a posteriori,

elle peut être en fait appliquée à toute forme de plongement lexical produit à partir du corpus dont est issu le thésaurus considéré. Comme dans le cas du plongement d'un thésaurus, il s'agit d'une forme d'amorçage dans lequel les connaissances extraites d'un corpus – en l'occurrence des relations de similarité sémantique – sont utilisées pour améliorer les représentations lexicales élaborées à partir de ce même corpus, en l'occurrence le modèle SGNS du tableau 1. À l'instar de la plupart des plongements lexicaux de ce type, SGNS se fonde sur des relations de cooccurrence lexicale de premier ordre. Adapter un plongement de type SGNS avec des relations issues d'un thésaurus distributionnel construit à partir du même corpus que ce plongement peut donc être vu comme une façon d'y incorporer des relations de cooccurrence de second ordre.

Méthode	**R-préc.**	**MAP**	**P@1**	**P@5**	**P@10**
SGNS	6,7	5,3	20,7	11,5	8,3
SVD + retrofit	10,0	7,9	28,9	18,1	13,5
SGNS + counter-rankfit	7,4	5,9	26,0	14,0	9,8
SGNS + retrofit	7,3	5,9	24,4	14,3	10,3

TABLE 5: Évaluation de l'amélioration endogène d'un plongement SGNS (référence : W+M)

Pour cette expérimentation, nous avons appliqué à la fois les méthodes *retrofit* et *counter-rankfit* avec les mêmes paramètres qu'à la section 3.3. Les résultats du tableau 5 valident clairement le bénéfice de cette approche : *retrofit* et *counter-rankfit* améliorent toutes deux le modèle SGNS. Comme précédemment, les deux méthodes ont des résultats très proches, avec une supériorité de *counter-rankfit* pour la précision au rang 1 comme seul élément notable. Il est également à noter que les modèles SGNS améliorés sont encore assez loin des meilleurs résultats de notre méthode de plongement de thésaurus (*SVD + retrofit*).

5 Amélioration exogène de plongements lexicaux

Le processus de plongement de thésaurus que nous avons présenté peut aussi contribuer de façon globale à la production de plongements lexicaux incorporant de façon plus effective des connaissances sémantiques externes que les méthodes *retrofit* et *counterfit* dans leur application directe. La méthode pour ce faire comporte deux phases : dans un premier temps, les connaissances externes sont intégrées à un thésaurus distributionnel construit à partir du corpus considéré ; ensuite, ce thésaurus se voit appliquer le processus de plongement décrit à la section 2.

5.1 Injecter des connaissances externes dans un thésaurus distributionnel

En préambule, il faut préciser que les connaissances externes considérées prennent la forme de relations de similarité sémantique. Leur intégration dans un thésaurus distributionnel est réalisée au niveau de chaque entrée sous la forme d'un réordonnancement de ses voisins suivant deux grandes stratégies. La première considère que l'entrée ne se réduit pas à un seul mot mais regroupe l'ensemble des mots impliqués dans une des relations sémantiques incluant cette entrée. La similarité entre une entrée et chacun de ses voisins potentiels se définit alors comme la similarité entre ce voisin V et l'ensemble $\{E_i\}$ regroupant l'entrée et ses mots liés. Cette définition s'appuie sur la similarité entre

Méthode	R-préc.		MAP		P@1		P@5		P@10	
SGNS	6,7		5,3		20,7		11,5		8,3	
SGNS + retrofit(W)	31,7		29,9		84,0		41,7		26,4	
Thés$_{cnt}$	10,7		8,1		30,9		18,6		14,1	
SVD(Thés$_{cnt}$)	8,2		5,8		23,4		14,4		11,4	
SVD(Thés$_{cnt}$) + retrofit(W)	35,1	(3,4)	32,5	(2,6)	84,8	(0,8)	44,2	(2,5)	29,3	(2,9)
Thés$_{cnt}$ + W	35,7		33,5		88,6		49,1		31,8	
SVD(Thés$_{cnt}$ + W)	17,2		16,0		47,6		31,9		22,7	
SVD(Thés$_{cnt}$ + W) + retrofit(W)	35,4	(3,7)	33,3	(3,4)	86,9	(2,9)	47,1	(5,4)	30,6	(4,2)

TABLE 6: Amélioration exogène de plongements lexicaux (référence : W+M)

mots exploitée lors de la construction du thésaurus. Un tel cas de figure se traite classiquement en calculant la similarité de tous les couples (V, E_i) et en leur appliquant une fonction d'agrégation. Nous avons considéré ici les fonctions *min*, *max* et *moyenne*. Le réordonnancement des voisins est obtenu en recalculant leur similarité avec l'entrée suivant ces modalités. La seconde stratégie consiste simplement à faire remonter les mots liés à l'entrée suivant les relations de similarité aux premiers rangs de ses voisins en leur attribuant une valeur de similarité maximale, sans toucher à l'ordonnancement des autres voisins. Nous avons constaté que les écarts de résultat entre ces deux grandes stratégies, de même qu'entre les différentes variantes de première, ne sont pas très importantes, avec néanmoins un petit avantage à la seconde que nous avons donc retenue pour nos expérimentations.

La ligne *Thés$_{cnt}$ + W* du tableau 6 donne l'évaluation de cette injection dans le thésaurus Thés$_{cnt}$ pour 30 134 relations de synonymie issues de WordNet 3.0. Il est à noter que les résultats obtenus sont très élevés, en particulier pour la précision au rang 1. Ce constat était partiellement attendu dans la mesure où la référence W+M comprend les synonymes de WordNet injectés dans le thésaurus et que la méthode d'intégration favorise leur apparition parmi les rangs les plus élevés. Néanmoins, la plupart des mots liés constituant la référence W+M ne sont pas des synonymes de WordNet et des mesures comme la précision au rang 5 ou même la R-précision ou la MAP ont des valeurs élevées alors même que le nombre moyen de synonymes par entrée se limite à 2,9.

5.2 D'un thésaurus enrichi en connaissances à des plongements lexicaux

Le processus que nous avons décrit à la section précédente produit ce que l'on peut appeler un thésaurus distributionnel enrichi en connaissances. Néanmoins, il n'est pas différent dans sa forme d'un thésaurus distributionnel classique et à ce titre, il peut donc se voir appliquer la procédure de plongement que nous avons présentée à la section 2. La seule différence avec cette procédure concerne sa seconde étape : au lieu d'une amélioration endogène des plongements produits par SVD, cette amélioration est exogène, en utilisant les connaissances ayant permis de produire le thésaurus enrichi.

Les résultats de l'évaluation globale de la nouvelle méthode que nous proposons pour construire des plongements lexicaux intégrant des connaissances externes se retrouvent dans le tableau 6. Plus précisément, trois méthodes sont comparées : une méthode de l'état de l'art, *SGNS + retrofit(W)*, consistant à appliquer le *retrofit* à des plongements produits par SGNS avec des synonymes de WordNet. Le *retrofit* a été choisi à la fois du fait de son efficacité et de sa rapidité. La deuxième

Référence	Méthode	R@100	R-préc.	MAP	P@1	P@5	P@10
WordNet	SGNS + retrofit(W)	88,6	83,7	86,3	94,7	39,1	21,6
	SVD(Thés$_{cnt}$ + W) + retrofit(W)	100	96,4	98,2	99,7	47,1	26,7
Moby	SGNS + retrofit(W)	11,8	10,8	7,4	60,7	35,8	24,4
	SVD(Thés$_{cnt}$ + W) + retrofit(W)	14,6	12,4	8,6	60,3	38,5	27,4

TABLE 7: Amélioration exogène de plongements lexicaux (référence : W et M)

méthode, *SVD(Thés$_{cnt}$) + retrofit(W)*, applique le *retrofit* aux plongements construits par SVD à partir du thésaurus Thés$_{cnt}$. La dernière méthode, *SVD(Thés$_{cnt}$ + W) + retrofit(W)*, correspond au processus complet que nous avons présenté dans lequel les connaissances externes sont d'abord injectées dans le thésaurus Thés$_{cnt}$ avant que celui-ci ne subisse un plongement. Comme à la section 5.1, les connaissances externes (W) étaient constituées de synonymes de WordNet 3.0.

Le premier constat évident à tirer de cette évaluation est que toutes les méthodes intégrant des connaissances externes aux plongements lexicaux conduisent à une augmentation très significative des résultats par rapport à leur point de départ. Cependant, cette augmentation générale ne nivelle pas les différences initiales. Par exemple, de même que *SVD(Thés)* obtient sur le même corpus des résultats supérieurs à SGNS, *SVD(Thés$_{cnt}$) + retrofit(W)* obtient des résultats supérieurs à *SGNS + retrofit(W)* (les différences sont données entre parenthèses). De façon plus notable, le tableau 6 montre que l'intégration des connaissances externes au thésaurus avant de construire son plongement est une stratégie très efficace, ainsi que l'illustrent les différences avec *SGNS + retrofit(W)* entre parenthèses. De plus, comme à la section 2, le relativement faible écart entre *Thés$_{cnt}$ + W* et *SVD(Thés$_{cnt}$ + W) + retrofit(W)* confirme que la méthode que nous avons proposée pour le plongement des thésaurus distributionnels présente de bonnes capacités de préservation de l'information de similarité sémantique présente dans ces thésaurus.

Pour finir, le tableau 7 donne une vue plus précise des résultats principaux du tableau 6 en les décomposant selon nos deux références, WordNet et Moby. Comme évoqué ci-dessus, les très hauts résultats obtenus pour WordNet se comprennent aisément par le fait que les connaissances externes considérées sont issues de WordNet. Néanmoins, le tableau 7 montre que les méthodes évaluées pour l'injection de connaissances dans les plongements lexicaux sont très efficaces dans leur capacité à faire mémoriser ces connaissances par les plongements. Comme en attestent la précision au rang 1, la R-précision et la MAP, cette mémorisation est de fait presque parfaite. Plus globalement, un tel niveau de résultat suggère que les plongements ainsi obtenus sont des candidats intéressants pour l'extension des connaissances injectées. Cette capacité de généralisation est d'ailleurs confirmée dans une certaine mesure par les résultats obtenus pour Moby. En effet, les précisions aux rangs 1 et 5 ont dans ce cas un niveau élevé alors qu'une large part des relations de Moby ne sont pas présentes dans WordNet. Cette capacité semble supérieure là aussi pour la méthode que nous proposons par rapport aux méthodes de l'état de l'art. Néanmoins, dans les deux cas, elle est encore restreinte à un sous-ensemble des relations de Moby comme l'illustrent la R-précision et la MAP.

6 Conclusions et perspectives

Dans cet article, nous avons présenté une méthode pour construire des plongements lexicaux à partir de thésaurus distributionnels en préservant l'essentiel de l'information de similarité sémantique

contenue par ces derniers. Une évaluation intrinsèque à grande échelle a permis de montrer que les plongements obtenus se comparent favorablement à des modèles de plongement de l'état de l'art. Nous avons également montré que cette méthode permet à la fois d'améliorer des plongements lexicaux existants de façon endogène et d'obtenir une amélioration exogène de ces mêmes plongements plus performante en termes de résultats.

Une première extension de ce travail a pour objectif d'intégrer davantage les deux étapes du processus de plongement des thésaurus distributionnels en exploitant plus efficacement les caractéristiques intrinsèques de ces derniers, en particulier au niveau de l'ordonnancement des voisins de leurs entrées. Une deuxième extension vise l'amélioration du processus d'injection de connaissances externes dans les thésaurus en testant de nouvelles méthodes d'agrégation des relations associées à une entrée. Enfin, il nous semble intéressant d'étudier plus avant les capacités des plongements issus de ces processus d'injection de connaissances pour étendre des ressources telles que WordNet.

Références

BARONI M., DINU G. & KRUSZEWSKI G. (2014). 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)*, p. 238–247, Baltimore, Maryland.

BATCHKAROV M., KOBER T., REFFIN J., WEEDS J. & WEIR D. (2016). A critique of word similarity as a method for evaluating distributional semantic models. In *1st Workshop on Evaluating Vector-Space Representations for NLP*, p. 7–12, Berlin, Germany.

BELKIN M. & NIYOGI P. (2001). Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems 14*, p. 585–591.

BENGIO Y., DELALLEAU O. & ROUX N. L. (2006). Label Propagation And Quadratic Criterion. In *Semi-Supervised Learning*, p. 193–216. MIT Press.

BUDANITSKY A. & HIRST G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, **32**(1), 13–47.

BULLINARIA J. A. & LEVY J. P. (2012). Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods*, **44**(3), 890–907.

CAO S., LU W. & XU Q. (2016). Deep Neural Networks for Learning Graph Representations. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, p. 1145–1152: AAAI Press.

CARON J. (2001). Computational Information Retrieval. chapter Experiments with LSA Scoring: Optimal Rank and Basis, p. 157–169. Society for Industrial and Applied Mathematics.

CLAVEAU V., KIJAK E. & FERRET O. (2014). Improving distributional thesauri by exploring the graph of neighbors. In *COLING 2014*, p. 709–720, Dublin, Ireland.

CURRAN J. R. & MOENS M. (2002). Improvements in automatic thesaurus extraction. In *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, p. 59–66, Philadelphia, USA.

FARUQUI M., DODGE J., JAUHAR S. K., DYER C., HOVY E. & SMITH N. A. (2015). Retrofitting Word Vectors to Semantic Lexicons. In *NAACL HLT 2015*, p. 1606–1615, Denver, Colorado.

FERRET O. (2010). Similarité sémantique et extraction de synonymes à partir de corpus. In *17ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2010)*, Montréal, Canada.

FERRET O. (2013). Sélection non supervisée de relations sémantiques pour améliorer un thésaurus distributionnel. In *20ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2013)*, p. 48–61, Les Sables d'Olonne, France.

GABRILOVICH E. & MARKOVITCH S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, p. 6–12.

GREFENSTETTE G. (1994). *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers.

GROVER A. & LESKOVEC J. (2016). Node2Vec: Scalable Feature Learning for Networks. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016)*, p. 855–864: ACM.

HILL F., REICHART R. & KORHONEN A. (2015). SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, **41**(4), 665–695.

HU Y., KOREN Y. & VOLINSKY C. (2008). Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining (ICDM'08)*, p. 263–272.

KIELA D. & CLARK S. (2014). A Systematic Study of Semantic Vector Space Model Parameters. In *2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, p. 21–30.

KOREN Y. (2008). Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, p. 426–434.

LANDAUER T. K. & DUMAIS S. T. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, **104**(2), 211–240.

LEVY O., GOLDBERG Y. & DAGAN I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics (TALC)*, **3**, 211–225.

LIN D. (1998). Automatic retrieval and clustering of similar words. In *17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (ACL-COLING'98)*, p. 768–774, Montréal, Canada.

LIU Q., JIANG H., WEI S., LING Z.-H. & HU Y. (2015). Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints. In *ACL-IJCNLP 2015*, p. 1501–1511, Beijing, China.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

MILLER G. A. (1990). WordNet: An On-Line Lexical Database. *International Journal of Lexicography*, **3**(4).

MRKŠIĆ N., Ó SÉAGHDHA D., THOMSON B., GAŠIĆ M., ROJAS-BARAHONA L. M., SU P.-H., VANDYKE D., WEN T.-H. & YOUNG S. (2016). Counter-fitting Word Vectors to Linguistic Constraints. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, p. 142–148, San Diego, California.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). GloVe: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, p. 1532–1543, Doha, Qatar.

PEROZZI B., AL-RFOU R., KULKARNI V. & SKIENA S. (2014a). *Inducing Language Networks from Continuous Space Word Representations*, In *Complex Networks V: 5th Workshop on Complex Networks (CompleNet 2014)*, p. 261–273. Springer International Publishing: Bologna, Italy.

PEROZZI B., AL-RFOU R. & SKIENA S. (2014b). DeepWalk: Online Learning of Social Representations. In *KDD 2014*, p. 701–710.

TANG G., RAO G., YU D. & XUN E. (2016). *Can We Neglect Function Words in Word Embedding?*, In C.-Y. LIN, N. XUE, D. ZHAO, X. HUANG & Y. FENG, Eds., *Natural Language Understanding and Intelligent Applications: 5^{th} CCF Conference on Natural Language Processing and Chinese Computing (NLPCC 2016) and 24^{th} International Conference on Computer Processing of Oriental Languages (ICCPOL 2016)*, p. 541–548. Springer International Publishing: Kunming, China.

TANG J., QU M., WANG M., ZHANG M., YAN J. & MEI Q. (2015). LINE: Large-scale Information Network Embedding. In *WWW 2015*, p. 1067–1077.

WARD G. (1996). Moby Thesaurus. Moby Project.

XU C., BAI Y., BIAN J., GAO B., WANG G., LIU X. & LIU T.-Y. (2014). RC-NET: A General Framework for Incorporating Knowledge into Word Representations. In *CIKM 2014*, p. 1219–1228.

YAN S., XU D., ZHANG B., J. ZHANG H., YANG Q. & LIN S. (2007). Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(1), 40–51.

YIH W.-T., ZWEIG G. & PLATT J. (2012). Polarity Inducing Latent Semantic Analysis. In *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, p. 1212–1222, Jeju Island, Korea.

YU M. & DREDZE M. (2014). Improving Lexical Embeddings with Semantic Knowledge. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, p. 545–550.

ZHANG J., SALWEN J., GLASS M. & GLIOZZO A. (2014). Word Semantic Representations using Bayesian Probabilistic Tensor Factorization. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, p. 1522–1531, Doha, Qatar.