Compression de vocabulaire de sens grâce aux relations sémantiques pour la désambiguïsation lexicale

Loïc Vial Benjamin Lecouteux Didier Schwab
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
{loic.vial, benjamin.lecouteux, didier.schwab}@univ-grenoble-alpes.fr

RÉSUMÉ

En Désambiguïsation Lexicale (DL), les systèmes supervisés dominent largement les campagnes d'évaluation. La performance et la couverture de ces systèmes sont cependant rapidement limités par la faible quantité de corpus annotés en sens disponibles. Dans cet article, nous présentons deux nouvelles méthodes qui visent à résoudre ce problème en exploitant les relations sémantiques entre les sens tels que la synonymie, l'hyperonymie et l'hyponymie, afin de compresser le vocabulaire de sens de WordNet, et ainsi réduire le nombre d'étiquettes différentes nécessaires pour pouvoir désambiguïser tous les mots de la base lexicale. Nos méthodes permettent de réduire considérablement la taille des modèles de DL neuronaux, avec l'avantage d'améliorer leur couverture sans données supplémentaires, et sans impacter leur précision. En plus de nos méthodes, nous présentons un système de DL qui tire parti des récents travaux sur les représentations vectorielles de mots contextualisées, afin d'obtenir des résultats qui surpassent largement l'état de l'art sur toutes les tâches d'évaluation de la DL.

ABSTRACT

Sense Vocabulary Compression through Semantic Knowledge for Word Sense Disambiguation

In Word Sense Disambiguation (WSD), supervised approaches are predominant in evaluation campaigns. The limited quantity of such corpora however restricts the coverage and the performance of these systems. In this article, we present two new methods that tackle this problem by exploiting the semantic relationships between senses such as synonymy, hypernymy and hyponymy, in order to compress the sense vocabulary of WordNet, and thus reduce the number of different sense tags that must be observed to disambiguate all words of the lexical database. Our methods greatly reduce the size of neural WSD models, with the benefit of improving their coverage without additional training data, and without impacting their precision. In addition to our methods, we present a neural WSD system which relies on the recent advances in contextualized word embeddings in order to achieve results that significantly outperform the state of the art on all WSD evaluation tasks.

MOTS-CLÉS : désambiguïsation lexicale, compression de vocabulaire, relations sémantiques.

KEYWORDS: Word Sense Disambiguation, Vocabulary Compression, Semantic Relationships.

1 Introduction

La Désambiguïsation Lexicale (DL) est une tâche qui vise à clarifier un texte en assignant à chacun de ses mots l'étiquette de sens la plus appropriée depuis un inventaire de sens prédéfini.

Il existe diverses approches pour la DL, telles que les approches à base de connaissances, qui

s'appuient sur des dictionnaires, des bases de données lexicales ou des graphes de connaissances couplés à des algorithmes tels que les mesures de similarité lexicale (Lesk, 1986) ou des mesures basées sur les graphes (Moro *et al.*, 2014), ou les méthodes supervisées, qui exploitent des corpus annotés en sens comme données d'apprentissage pour entraîner un classifieur multi-classe tel qu'un SVM (Chan *et al.*, 2007; Zhong & Ng, 2010), ou plus récemment un réseau neuronal (Kågebäck & Salomonsson, 2016). Les méthodes supervisées sont de loin les plus représentées car elles offrent généralement les meilleurs résultats dans les campagnes d'évaluation (par exemple (Navigli *et al.*, 2007)). Les classifieurs état de l'art combinaient jusqu'à récemment des caractéristiques précises telles que les parties du discours et les lemmes des mots voisins, (Zhong & Ng, 2010), mais ils sont maintenant remplacés par des réseaux de neurones récurrents qui apprennent leur propre représentation des mots (Raganato *et al.*, 2017; Le *et al.*, 2018; Vial *et al.*, 2019).

Cependant, une des limitations majeures des systèmes supervisés est la quantité limitée de corpus manuellement annotés en sens. En effet, le SemCor (Miller *et al.*, 1993), qui est le plus grand corpus manuellement annoté en sens disponible, contient 33 760 labels de sens différents, ce qui correspond à seulement environ 16% de l'inventaire de sens de WordNet [1] (Miller *et al.*, 1990), la base de données lexicale de référence largement utilisée en DL. De nombreux travaux tentent de résoudre ce problème via la création de nouveaux corpus annotés en sens, générés soit automatiquement (Pasini & Navigli, 2017), semi-automatiquement (Taghipour & Ng, 2015), ou bien par *crowdsourcing* (Yuan *et al.*, 2016), mais dans nos travaux, nous cherchons à résoudre ce problème en tirant parti des relations sémantiques présentes entre les sens de WordNet comme l'hyperonymie, l'hyponymie, l'antonymie, la méronymie, etc. Notre méthode est basée sur les observations suivantes :

1. Un sens et ses sens voisins dans le graphe des relations sémantiques de WordNet véhiculent tous une même idée ou concept, à des niveaux d'abstraction différents.
2. Dans certains cas, un mot peut être désambiguïsé en utilisant seulement les sens voisins de ses sens, et pas nécessairement ses sens propres.
3. Par conséquent, nous n'avons pas besoin de connaître tous les sens de WordNet pour désambiguïser tous les mots de WordNet.

Par exemple, considérons le mot « souris » et deux de ses sens : la souris *d'ordinateur* et la souris *l'animal*. Les notions plus générales comme « être vivant » (hyperonyme de souris/animal) et « appareil électronique » (hyperonyme de souris/ordinateur), permettent déjà de distinguer les deux sens, et toutes les notions plus spécialisées telles que « rongeur » ou « mammifère » sont, elles, superflues. En regroupant ces étiquettes de sens ensemble, on peut bénéficier de tous les autres exemples mentionnant un appareil électronique ou un être vivant dans un corpus d'entraînement, même si le mot « souris » n'est pas mentionné spécifiquement, pour désambiguïser le mot « souris ».

Contributions : Dans cet article, nous émettons l'hypothèse que seul un sous-ensemble des sens de WordNet peut être considéré pour pouvoir désambiguïser tous les mots de la base lexicale. Par conséquent, nous proposons deux méthodes différentes pour construire ce sous-ensemble que nous appelons méthodes de compression de vocabulaire de sens. En utilisant ces techniques, nous sommes en mesure d'améliorer considérablement la couverture des systèmes de DL supervisés, en éliminant quasiment le besoin d'une stratégie de repli habituellement employée pour les mots jamais observés pendant l'entraînement. Nous présentons des résultats qui surpassent l'état de l'art de façon significative sur toutes les tâches d'évaluation de la DL, et nous fournissons à la communauté notre outil ainsi que nos meilleurs modèles pré-entraînés, sur un dépôt GitHub dédié [2].

1. https://wordnet.princeton.edu/documentation/wnstats7wn
2. https://github.com/getalp/disambiguate

2 Désambiguïsation lexicale neuronale

Plusieurs avancées récentes ont été réalisées dans la création de nouvelles architectures neuronales pour les systèmes supervisés de désambiguïsation lexicale. Ces systèmes atteignent des performances état de l'art et certains peuvent intégrer des sources de connaissances externes. Dans cette section, nous donnons un aperçu de ces travaux.

2.1 Approches basées sur un modèle de langue

Dans ce type d'approches, initié par Yuan *et al.* (2016) et réimplémenté par Le *et al.* (2018), le composant principal est un modèle de langue neuronal capable de prédire un mot en tenant compte des mots qui l'entourent, grâce à un réseau neuronal entraîné sur une quantité massive de données non annotées (100 milliards de mots pour Yuan *et al.* (2016) et 1,8 milliards pour Le *et al.* (2018)).

Une fois le modèle de langue entraîné, il est utilisé pour produire des vecteurs de sens en moyennant les vecteurs de mots prédits par le modèle à l'endroit où ces mots sont annotés avec un sens particulier.

Au moment du test, le modèle de langue est utilisé pour prédire un vecteur en fonction du contexte environnant, et le sens le plus proche du vecteur prédit est attribué à chaque mot.

Ces systèmes ont l'avantage de contourner le problème de l'absence de données annotées en sens en concentrant le pouvoir d'abstraction offert par les réseaux neuronaux récurrents sur un modèle de langue de bonne qualité et entraîné de manière non supervisée. Cependant, ces méthodes souffrent toujours du manque de corpus annotés en sens étant donné qu'ils restent indispensables pour la création des vecteurs de sens.

2.2 Approches basées sur un classifieur linéaire et la fonction *softmax*

Dans ces systèmes, le réseau neuronal principal classifie et attribue directement un sens à chaque mot donné en entrée à l'aide d'une distribution de probabilité calculée par la fonction *softmax*. Les annotations en sens sont simplement considérées comme des balises placées sur chaque mot, à la manière d'une tâche d'étiquetage en parties du discours par exemple.

On peut distinguer deux branches distinctes de ces types de réseaux neuronaux :

1. Ceux dans lesquels il y a un réseau neuronal (ou classifieur) distinct et spécifique à chaque lemme du dictionnaire (Iacobacci *et al.*, 2016; Kågebäck & Salomonsson, 2016). Chaque classifieur est capable de gérer un lemme particulier avec ses sens. Par exemple, l'un des classifieurs est spécialisé dans le choix entre les quatre sens possibles du nom « souris ». Ce type d'approche est particulièrement adapté aux tâches de *lexical sample*, où seul un petit nombre de mots distincts et très ambigus doivent être annotés dans plusieurs contextes. Mais ils nécessiteraient plusieurs milliers de réseaux différents [3] pour pouvoir aussi être utilisés dans les tâches de désambiguïsation lexicales *all words*, dans lesquelles tous les mots d'un document doivent être annotés en sens.
2. Ceux dans lesquels il y a un seul réseau neuronal, plus grand et capable de gérer tous les lemmes du lexique, qui attribuent à un mot un sens issu de l'ensemble de tous les sens de l'inventaire de sens utilisé (Raganato *et al.*, 2017; Vial *et al.*, 2019).

L'avantage de la première branche est que pour désambiguïser un mot, il est beaucoup plus facile de

3. L'ensemble de WordNet contient par exemple 26 896 mots polysémiques (`https://wordnet.princeton.edu/documentation/wnstats7wn`)

limiter notre choix à l'un de ses sens possibles que de chercher parmi tous les sens de tous les mots du lexique. Pour se donner une idée, le nombre moyen de sens des mots polysémiques dans WordNet est d'environ 3, alors que le nombre total de sens en considérant tous les mots est 206 941. [4]

La seconde approche a cependant une propriété intéressante : tous les sens résident dans le même espace vectoriel et partagent donc des caractéristiques dans les couches cachées du réseau. Cela permet au modèle de prédire un sens identique pour deux mots différents (synonymes), mais aussi de prédire un sens pour un mot non présent dans le dictionnaire (néologisme, faute d'orthographe, etc.).

Enfin, dans deux articles récents, Luo *et al.* (2018a,b) ont proposés une amélioration de ce type d'architectures, en calculant une attention entre le contexte d'un mot cible et les définitions de ses différents sens. Ainsi, leur travail est le premier à incorporer les connaissances de WordNet dans un système de désambiguïsation neuronal.

3 Compression de vocabulaire de sens

Les systèmes supervisés neuronaux état de l'art tels que Yuan *et al.* (2016); Raganato *et al.* (2017); Le *et al.* (2018); Luo *et al.* (2018a,b); Vial *et al.* (2019) sont tous confrontés aux mêmes limitations :

1. La quantité de données annotés manuellement en sens étant très limitée, il se peut qu'un mot cible ne soit jamais observé pendant l'entraînement. Dans ce cas, le système ne peut pas être en mesure de l'annoter, et une stratégie de repli est généralement effectuée (par exemple utiliser le premier sens du mot dans WordNet).
2. Pour la même raison, un mot peut être observé, mais pas tous ses sens. Dans ce cas, le système va être capable d'annoter ce mot, mais si le sens attendu n'a jamais été observé, le résultat sera faux, quelle que soit l'architecture sous-jacente du système supervisé.
3. L'empreinte mémoire des modèles neuronaux ainsi que leur temps d'entraînement et d'exécution augmentent avec la quantité de données d'apprentissage et le nombre d'étiquettes de sens différentes prises en compte, nombre qui monte jusqu'à 206 941 si l'on considère toutes les étiquettes de sens de WordNet.

Afin de résoudre ces problèmes, nous proposons deux nouvelles méthodes permettant de regrouper ensemble des étiquettes de sens qui se réfèrent à des concepts similaires, tout en nous assurant que ces groupes de sens permettent toujours de discriminer les différents sens de tous les mots du lexique, afin de retrouver l'étiquette de sens originale pour un mot au moment de le désambiguïser. En conséquence, le vocabulaire de sens, c'est à dire le nombre total d'étiquettes de sens dans notre inventaire de sens diminue, le système est capable de mieux généraliser, et sa couverture augmente.

3.1 Des sens aux *synsets* : une première compression de vocabulaire de sens à travers la synonymie

Dans la base de données lexicale WordNet (Miller *et al.*, 1990), les sens sont organisés en ensembles de synonymes appelés *synsets*. Un *synset* est concrètement un groupe d'un ou plusieurs sens qui ont la même définition et donc la même signification. Par exemple, les premiers sens des mots « *eye* », « *optic* » et « *oculus* » appartiennent tous au même *synset* dont la définition est « l'organe de la vue ».

La conversion des étiquettes de sens (« Xème sens du mot N ») aux étiquettes de *synsets* (« *synset* numéro Y »), illustré dans la figure 1, est ainsi une façon de compresser le vocabulaire qui est

4. https://wordnet.princeton.edu/documentation/wnstats7wn

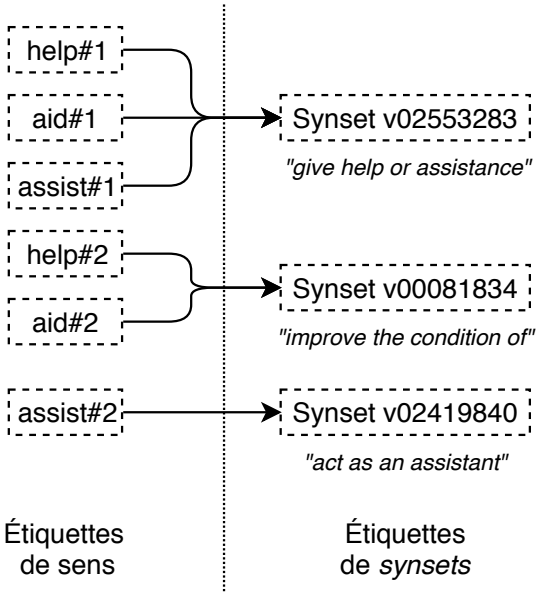

FIGURE 1 – Conversion des étiquettes de sens vers des étiquettes de *synsets*, appliqué aux deux premiers sens des mots « *help* », « *aid* » et « *assist* ». Le nombre de sens différents dans notre vocabulaire passe ainsi de six à trois.

déjà appliquée dans plusieurs travaux (Yuan *et al.*, 2016; Le *et al.*, 2018; Vial *et al.*, 2019) sans être toujours explicitement précisée. Cette méthode contribue pourtant clairement à améliorer la couverture des systèmes supervisés. En effet, si le verbe « *aid* » annoté avec son premier sens est observé dans les données d'apprentissage, le contexte autour du mot cible peut être aussi utile pour annoter ultérieurement les verbes « *assist* » ou « *help* » avec la même étiquette de *synset*.

En allant plus loin, on peut trouver d'autres informations dans WordNet qui peuvent aider à mieux généraliser. La première nouvelle méthode que nous proposons repose ainsi sur ce même principe de regroupement de sens, mais en exploitant les relations d'hyperonymie et d'hyponymie entre les sens.

3.2 Compression de vocabulaire de sens à travers les relations d'hyperonymie, d'hyponymie et d'instance

Selon Polguère (2003), l'hyperonymie et l'hyponymie sont deux relations sémantiques qui correspondent à un cas particulier d'inclusion de sens : l'hyponyme d'un terme est une spécialisation de ce terme, alors que son hyperonyme est une généralisation. Par exemple, une « souris » est un type de « rongeur » qui est à son tour un type d'« animal ». Dans WordNet, ces relations lient presque tous les noms ensemble allant de la racine générique, le nœud « entité » aux feuilles les plus spécifiques, par exemple « souris à pattes blanches ». Si l'on prend aussi en compte la relation d'instance, qui fonctionne de la même manière mais qui lie les entitées nommées aux noms courants (par exemple « Einstein » est une instance de « physicien »), tous les noms de WordNet font partie de cette même hiérarchie.

Ces relations sont également présentes sur plusieurs verbes : ainsi, par exemple, « additionner » est une manière de « calculer » qui est à son tour une manière de « raisonner ».

Pour la DL, tout comme le regroupement des synonymes en *synsets* aide à mieux généraliser, nous faisons l'hypothèse que le regroupement des sens faisant partie d'une même hiérarchie d'hyperonymie va aussi aider à mieux généraliser, et que les concepts les plus spécialisés de WordNet sont souvent superflus. En effet, si l'on considère un sous-ensemble de WordNet qui ne comprend que le mot « souris », avec son premier sens (le petit rongeur), son quatrième sens (le dispositif électronique), et tous leurs hyperonymes, tel qu'illustré dans la figure 2, on voit que les concepts « artefact » et « être vivant » suffisent à différencier les deux sens, et toutes les étiquettes plus spécialisées pourrait être ramenés à ces deux concepts. Ainsi, non seulement le vocabulaire de sens, c'est à dire le nombre

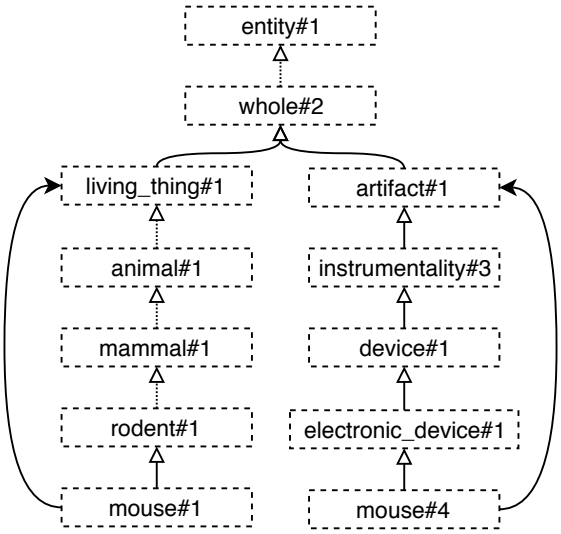

FIGURE 2 – Compression de vocabulaire utilisant la hiérarchie d'hyperonymie, appliquée au premier et quatrième sens du mot « *mouse* ». Les lignes en pointillés indiquent que des nœuds ont été omis pour la clarté.

d'étiquettes de sens dans notre inventaire, sera réduit, mais en plus tous les autres « êtres vivants » donneront des exemples qui pourront ensuite permettre de différencier les deux sens de souris.

En considérant maintenant tout le vocabulaire de WordNet, l'objectif de notre méthode est ainsi de faire correspondre chaque sens à son ancêtre le plus haut dans sa hiérarchie d'hyperonymie, avec les contraintes suivantes : Premièrement, cet ancêtre doit permettre de discriminer tous les différents sens du mot cible. Deuxièmement, nous devons conserver les hyperonymes qui sont indispensables pour discriminer les sens des autres mots du dictionnaire. Par exemple, en prenant tout WordNet en considération, nous ne pouvons pas faire correspondre « souris#1 » à « être vivant#1 », parce qu'une étiquette plus spécifique, « animal#1 » est nécessaire pour distinguer les deux sens du mot « proie » (un sens décrit une personne et l'autre un animal).

Notre méthode fonctionne donc en deux étapes :

1. Nous marquons comme « nécessaires » les enfants du premier ancêtre commun de chaque paire de sens de chaque mot de WordNet.
2. Nous faisons correspondre chaque sens à son ancêtre le plus bas dans sa hiérarchie d'hyperonymie ayant été précédemment marqué comme « nécessaire ».

En conséquence, les sens les plus spécifiques de l'arbre qui ne sont pas indispensables pour distinguer un mot de l'inventaire lexical seront automatiquement supprimés du vocabulaire. En d'autres termes, l'ensemble de sens qui reste dans le vocabulaire est le plus petit sous-ensemble de tous les *synsets* qui sont nécessaires pour distinguer chaque sens de chaque mot de WordNet, en considérant seulement les liens d'hyperonymie, d'hyponymie et d'instance.

3.3 Compression de vocabulaire de sens à travers l'ensemble des relations sémantiques de WordNet

En plus de l'hyperonymie, de l'hyponymie et de la relation d'instance, WordNet contient plusieurs autres relations entre *synsets*, telles que la méronymie (X fait partie de Y, ou X est un membre de Y) et son opposé l'holonymie, l'antonymie (X est le contraire de Y) et son opposé la similarité, etc.

Nous proposons ainsi une deuxième nouvelle méthode de compression du vocabulaire de sens, qui prend en compte toutes les relations sémantiques offertes par WordNet, afin de former des groupes de *synsets* proches.

Par exemple, en utilisant toutes les relations sémantiques disponibles, nous pourrions former un groupe contenant « physicien », « physique » (domaine), « Einstein » (instance), « astronome » (hyponyme), mais aussi d'autres sens connexes tels que « photon », car c'est un méronyme de « rayonnement », qui est un hyponyme de « énergie », qui appartient au même domaine de « physique », etc.

Notre méthode fonctionne en construisant ces groupes de manière itérative. Soit S l'ensemble des *synsets* de WordNet et C l'ensemble des groupes de *synsets* que l'on cherche à construire, on initialise d'abord C comme des singletons contenant chacun un *synset* différent.

$$S = \{s_0, s_1, ..., s_n\} \qquad C = \{c_0, c_1, ..., c_n\} \qquad C = \{\{s_0\}, \{s_1\}, ..., \{s_n\}\}$$

Ensuite, à chaque étape, on trie C par taille de groupes, et on sélectionne le plus petit groupe c_x ainsi que le plus petit groupe relié à c_x, c_y. On considère qu'un groupe c_a est relié à un groupe c_b si un *synset* $s_a \in c_a$ est relié à un *synset* $s_b \in c_b$ par n'importe quel lien sémantique. On fusionne c_x et c_y ensemble, si et seulement si l'opération permet toujours de discriminer les différents sens de tous les mots de la base lexicale. Si c'est le cas, on valide la fusion et on passe à l'étape suivante. Si ce n'est pas le cas, on annule la fusion et on essaye avec un autre groupe relié à c_x. S'il est impossible de fusionner un groupe avec c_x, alors on essaye avec le plus petit groupe suivant, et si aucune fusion n'est possible pour aucun des groupes, l'algorithme s'arrête.

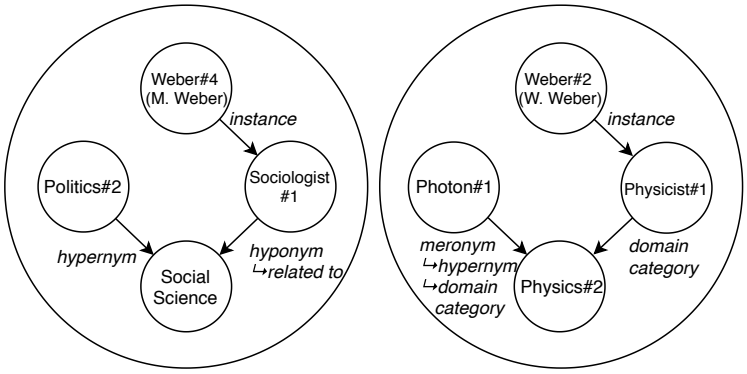

FIGURE 3 – Exemple de groupes de sens pouvant résulter de notre méthode, si on ne considère que deux sens du nom « Weber » et seulement certaines relations.

Dans la figure 3, nous montrons un ensemble possible de groupes qui pourraient résulter de notre méthode.

Cette méthode produit des groupes significativement plus grands que celle s'appuyant sur les hyperonymes. En effet, en moyenne, un groupe contient 5 *synsets* avec cette dernière, alors qu'il en contient 17 avec cette méthode. De plus, cette méthode, contrairement à la précédente, est également stochastique, parce qu'à chaque fois qu'on ordonne les groupes par taille, l'algorithme de tri place les groupes de même taille dans un ordre aléatoire. Cependant, comme nous réordonnons les groupes après chaque fusion, les groupes sont de taille assez équilibrés, et nous avons observé que la taille finale du vocabulaire (c.-à-d. le nombre de groupes) se situe toujours entre 11 000 et 13 000,

Dans la suite, on considère un ensemble C généré après que l'algorithme se soit arrêté après 105 775 étapes de fusion (générant ainsi 11 885 groupes de sens).

La table 1 montre l'effet de la compression de vocabulaire via les synonymes (sens vers *synsets*), de notre première nouvelle méthode utilisant les hyperonymes, ainsi que de notre deuxième nouvelle méthode utilisant toutes les relations de WordNet, sur la taille du vocabulaire de sens de WordNet, et sur la couverture du SemCor. Comme nous pouvons le constater, la taille du vocabulaire diminue considérablement grâce à nos méthodes, et la couverture d'un même corpus est nettement améliorée.

Méthode de compression	Taille du vocabulaire	Taux de compression	Couverture du SemCor
Référence	206 941	référence	16%
Synonymes	117 659	43%	22%
Hyperonymes	39 147	81%	32%
Toutes relations	11 885	94%	39%

TABLE 1 – Résultats de nos deux méthodes de compression de vocabulaire sur la taille du vocabulaire et la couverture du SemCor. La couverture correspond au ratio du nombre d'étiquettes de sens différentes observables dans le corpus sur le nombre total d'étiquettes (taille du vocabulaire).

4 Protocole expérimental

Afin d'évaluer nos méthodes de compression de vocabulaire de sens, nous les avons appliquées à un système neuronal de DL état de l'art similaire à celui de Vial *et al.* (2019) (voir la section 2.2). Notre réseau de neurones prend ainsi en entrée directement les mots sous forme vectorielle, à partir d'un modèle de vecteurs de mots pré-entraîné, il repose ensuite sur une ou plusieurs couches cachées, puis sur une couche de sortie, qui associe à chaque mot une distribution de probabilité sur tous les sens du vocabulaire utilisé, à l'aide de la fonction *softmax*.

4.1 Détails de l'architecture

En entrée de notre réseau, nous avons utilisé les vecteurs contextualisés BERT (Devlin *et al.*, 2018). Nous avons utilisé le modèle pour l'anglais « bert-large-cased » qui est pré-entraîné sur BookCorpus (Zhu *et al.*, 2015) et Wikipedia, et qui produit des vecteurs de dimension 1 024.

Pour les couches cachées, nous avons appliqué 6 couches d'encodeurs *Transformer* (Vaswani *et al.*, 2017), avec les même paramètres que le modèle « base » de l'article original (8 têtes d'attention, dimension cachée de 2 048, et régularisation *dropout* à 0,1). Les couches *Transformer* sont basés sur le mécanisme d'auto-attention, et nous les avons utilisés à la place des cellules récurrentes plus classiques comme des *LSTM* ou des *GRU*, parce que plusieurs travaux récents ont montré leur plus grande efficacité dans une multitude de tâches, par exemple en traduction automatique (Vaswani *et al.*, 2017; Ott *et al.*, 2018) et en modélisation de la langue (Devlin *et al.*, 2018).

De plus, étant donné que les vecteurs retournés par BERT encodent directement les positions des mots, il n'est pas nécessaire d'avoir une récurrence au niveau des couches cachées. Ainsi, nous n'ajoutons pas de vecteurs de positions supplémentaires en entrée de notre encodeur.

Pour tous les autres paramètres du modèle, comme le nombre de phrase par mini-lot, et la méthode d'optimisation, nous avons utilisé les mêmes paramètres que Vial *et al.* (2019).

4.2 Entraînement du modèle

Nous avons comparé nos méthodes sur deux ensembles de corpus d'entraînement : le SemCor (Miller *et al.*, 1993), le plus grand corpus annoté en sens utilisé pour l'apprentissage de la plupart des systèmes supervisés de DL, et la concaténation du SemCor et du WordNet Gloss Tagged [5]. Ce dernier est un corpus distribué dans WordNet depuis sa version 3.0, et il contient les définitions de tous les sens de WordNet, annoté manuellement ou semi-automatiquement en sens. Nous avons utilisé les versions de

5. http://wordnetcode.princeton.edu/glosstag-files/glosstag.shtml

ces corpus fournies avec la ressource UFSAC 2.1 [6] (Vial *et al.*, 2018).

Système	SE2	SE3	SE07 17	SE13	SE15	ALL	SE07 07
SemCor, référence	91,15	96,76	97,58	91,06	94,78	93,23	92,84
SemCor, hyperonymes	98,03	99,19	99,78	99,15	98,39	98,75	98,85
SemCor, toutes relations	99,56	99,84	100	100	98,92	99,67	99,69
SemCor+WNGT, référence	97,81	98,92	99,34	97,63	99,34	98,26	98,45
SemCor+WNGT, hyperonymes	99,74	99,95	100	99,76	99,91	99,83	99,91
SemCor+WNGT, toutes relations	100	100	100	100	99,91	99,99	100
Nombre de mots à annoter	2282	1850	455	1644	1022	7253	2261

TABLE 2 – Couverture (en %) des corpus d'évaluation en fonction du corpus d'apprentissage et de l'utilisation de notre méthode de compression de vocabulaire.

Nous avons choisi d'ajouter uniquement le WNGT en plus du SemCor à nos données d'entraînement, et pas tous les corpus de la ressource UFSAC 2.1, parce que c'est le seul, avec le SemCor, dans lequel à la fois tous les mots sont annotés en sens, l'inventaire de sens utilisé par les annotateurs est directement WordNet, les annotations ne sont pas entièrement automatiques, et la ressource est libre. Nous avons ainsi cherché à utiliser seulement des données de la meilleure qualité possible, pour éviter d'ajouter du bruit et/ou de rallonger le temps d'entraînement de nos modèles.

Nous avons entraîné chaque modèle sur 20 passes de nos données d'entraînement. Au début de chaque passe, nous avons mélangé toutes les phrases aléatoirement, et à la fin de chaque passe, nous avons évalué notre modèle sur un jeu de développement, et nous avons conservé celui qui a obtenu le meilleur score F1 de DL. Le corpus de développement est constitué de 4 000 phrases prises aléatoirement du WNGT pour le système entraîné sur le SemCor seul, et de 4 000 phrases extraites aléatoirement de nos données d'entraînement pour les autres.

Nous avons ainsi entraînés trois systèmes :

1. un système « référence » dont le vocabulaire de sens est celui de tous les *synsets* vus pendant l'entraînement (utilisant ainsi la compression classique via les synonymes) ;
2. un système « hyperonymes » entraîné dans les mêmes conditions, mais avec notre première méthode de compression du vocabulaire via les hyperonymes, les hyponymes et les instances appliquée sur le corpus d'entraînement ;
3. un système « toutes relations » qui applique cette fois-ci sur le corpus d'entraînement notre deuxième méthode de compression de vocabulaire via toutes les relations sémantiques de WordNet.

	Nombre de paramètres	
Système	SemCor	SemCor+WNGT
Référence	77,15M	120,85M
Hyperonymes	63,44M	79,85M
Toutes relations	55,16M	60,27M

TABLE 3 – Nombre de paramètres d'un modèle en fonction du corpus d'apprentissage et de notre méthode de compression de vocabulaire.

Tous les entraînements ont été effectués sur un seul GPU Titan X de Nvidia. Dans la table 3, nous montrons le nombre de paramètres des différents modèles, en fonction du corpus d'entraînement et de notre méthode de compression du vocabulaire. Comme nous pouvons le voir, ce nombre est réduit par un facteur de 1,2 à 2 grâce à nos méthodes de compression.

6. https://github.com/getalp/UFSAC

4.3 Résultats

Nous avons évalué nos modèles sur tous les corpus d'évaluation de la DL de l'anglais des campagnes d'évaluation SensEval/SemEval, c'est-à-dire les corpus d'évaluation « grain fin » de SensEval 2 (Edmonds & Cotton, 2001), SensEval 3 (Snyder & Palmer, 2004), SemEval 2007 (tâche 17) (Pradhan *et al.*, 2007), SemEval 2013 (Navigli *et al.*, 2013) et SemEval 2015 (Moro & Navigli, 2015), ainsi que le corpus « ALL » constitué de leur concaténation. Nous avons également comparé nos résultats sur la tâche « gros grain » de SemEval 2007 (tâche 7) (Navigli *et al.*, 2007).

Pour chaque évaluation, nous avons entraîné 8 modèles indépendant, et nous donnons le score obtenu par un système « ensemble » qui moyenne leurs prédictions à l'aide d'une moyenne géométrique.

Les scores obtenus par nos systèmes en comparaison avec les meilleurs systèmes de l'état de l'art et l'étalon du premier sens sont présents dans le tableau 4, et le tableau 2 montre la couverture de nos systèmes sur les tâches d'évaluation.

	SE2	SE3	SE07	SE13	SE15	ALL (concat. tâches précédentes)					SE07
Système			17			noms	verbes	adj.	adv.	total	07
Étalon du premier sens	65,6	66,0	54,5	63,8	67,1	67,7	49,8	73,1	80,5	65,5	78,9
UFSAC+1M (Vial *et al.*, 2019)	74,6	69,4	60,7	69,8	74,2	-	-	-	-	†71,1	85,0
HCAN (Luo *et al.*, 2018a)	72,8	70,3	-	68,5	72,8	72,7	58,2	77,4	84,1	71,1	-
LSTMLP (Yuan *et al.*, 2016)	73,8	71,8	63,5	69,5	72,6	†73,9	-	-	-	†71,5	83,6
SemCor, référence	77,2	76,5	70,1	74,7	77,4	78,7	65,2	79,1	85,5	76,0	87,7
SemCor, hyperonymes	77,5	77,4	69,5	76,0	78,3	79,6	65,9	79,5	85,5	76,7	87,6
SemCor, toutes relations	76,6	76,9	69,0	73,8	75,4	77,2	66,0	80,1	85,0	75,4	86,7
SemCor+WNGT, référence	**79,7**	76,1	**74,1**	78,6	80,4	80,6	68,1	82,4	86,1	78,3	90,4
SemCor+WNGT, hyperonymes	**79,7**	77,8	73,4	**78,7**	**82,6**	81,4	68,7	83,7	85,5	**79,0**	90,4
SemCor+WNGT, toutes relations	79,4	**78,1**	71,4	77,8	81,4	80,7	68,6	82,8	85,5	78,5	**90,6**

TABLE 4 – Scores F1 (%) sur les tâches de DL de l'anglais des campagnes d'évaluation SensEval/SemEval. La tâche « ALL » est la concaténation de SE2, SE3, SE07 17, SE13 et SE15. La stratégie de repli est appliquée sur les mots dont aucun sens n'a été observé pendant l'entraînement. Les scores en **gras** sont à notre connaissance les meilleurs résultats obtenus sur la tâche. Les scores prefixés par une obélisque (†) ne sont pas fournis par les auteurs mais sont déduits de leurs autres scores.

Concernant les résultats présentés dans la table 4, nous observons que nos systèmes qui utilisent nos méthodes de compression de vocabulaire, que ce soit à travers les relations d'hyperonymie ou à travers toutes les relations obtiennent des scores qui sont globalement équivalents ou légèrement supérieurs aux systèmes « référence » qui n'utilisent pas nos méthodes.

Nos méthodes de compression améliorent cependant grandement la couverture de nos systèmes. En effet, comme nous pouvons le voir dans la table 2, sur un total de 7 253 mots à annoter pour le corpus « ALL », le système de référence entraîné sur le SemCor n'est pas capable d'annoter 491 d'entre eux, alors qu'avec la compression du vocabulaire à travers les hyperonymes, ce nombre descend à 91, et 24 avec la compression à travers toutes les relations.

Lors de l'ajout du WordNet Gloss Tagged aux données d'entraînement, seulement 12 mots ne peuvent pas être annotés avec le système « hyperonymes », et avec le système « toutes relations », plus qu'un

Corpus d'entraînement	Vecteurs de mots pré-entraînés	Ensemble	Scores F1 sur la tâche "ALL" (%)					
			Référence		Hyperonymes		Toutes relations	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
SemCor+WNGT	BERT	Oui	78,27	-	79,00	-	78,48	-
SemCor+WNGT	BERT	Non	76,97	±0,38	77,08	±0,17	76,52	±0,36
SemCor+WNGT	ELMo	Oui	75,16	-	74,65	-	70,58	-
SemCor+WNGT	ELMo	Non	74,56	±0,27	74.36	±0,27	68,77	±0.30
SemCor+WNGT	GloVe	Oui	72,23	-	72,74	-	71,42	-
SemCor+WNGT	GloVe	Non	71,93	±0,35	71,79	±0,29	69,60	±0,32
SemCor	BERT	Oui	76,02	-	76,73	-	75,40	-
SemCor	BERT	Non	75,06	±0,26	75,59	±0,16	73,91	±0,33
SemCor	ELMo	Oui	72,55	-	73,09	-	69,43	-
SemCor	ELMo	Non	72,21	±0,13	72,83	±0,24	68,74	±0,29
SemCor	GloVe	Oui	70,77	-	71,18	-	68,44	-
SemCor	GloVe	Non	70,51	±0,16	70,77	±0,21	67,48	±0,55
Système « élève » (Vial *et al.*, 2019)								
SemCor+UFSAC+1M News 2016	GloVe	Oui	71,1					
HCAN (Luo *et al.*, 2018a)								
SemCor+WordNet glosses	GloVe	Non	71,1					
LSTMLP (Yuan *et al.*, 2016)								
SemCor+1K (private)	private	Non	71,5					

TABLE 5 – Étude des hyperparamètres sur la tâche "ALL" (concaténation des corpus de toutes les tâches de désambiguïsation lexicale à granularité fine de SensEval/SemEval). Pour les systèmes qui n'utilisent pas l'ensemble, nous montrons la moyenne des scores ($\bar{x}$) de huit modèles entraînés séparément, avec l'écart type (σ).

seul mot (l'adjectif monosémique « cytotoxic ») ne peut pas être annoté parce que son sens n'a pas été vu pendant l'entraînement. Si nous prenons en compte uniquement les mots polysémiques, le système basé sur la compression à travers toutes les relations et entraîné sur le SemCor n'est pas capable d'annoter seulement un seul mot (l'adverbe « eloquently »). Avec le WNGT en plus, il a une couverture de 100%.

Par rapport aux autres travaux, nous obtenons des résultats surpassant significativement l'état de l'art dans toutes les tâches, notamment grâce à l'ajout du WordNet Gloss Tagged aux données d'entraînement, et des vecteurs BERT en entrée de notre système.

4.4 Étude des hyperparamètres

Afin de mieux comprendre l'origine de nos scores, nous étudions l'impact de nos principaux paramètres sur les résultats. En plus du corpus d'entraînement et de la méthode de compression du vocabulaire, nous avons choisi deux paramètres qui nous différencient de l'état de l'art : le modèle de vecteurs de mots pré-entraînés, et la méthode d'ensemble, et nous les avons fait varier.

Pour le modèle de vecteurs de mots, nous avons expérimenté avec BERT (Devlin *et al.*, 2018) comme pour nos résultats principaux, mais aussi avec ELMo (Peters *et al.*, 2018) et GloVe (Pennington *et al.*, 2014). Pour ELMo, nous avons utilisé le modèle entraîné sur Wikipedia et les données monolingues de WMT 2008-2012. [7] Pour GloVe, nous avons utilisé le même modèle que Luo *et al.* (2018a) et Vial

7. https://allennlp.org/elmo

et al. (2019) entraîné sur Wikipedia 2014 et Gigaword 5. [8] Comme les représentations vectorielles de GloVe n'encodent pas la position des mots (un mot a la même représentation quelque soit sa position ou son contexte), nous avons réutilisé une couche de cellules *LSTM* bidirectionnelles de taille 1 000 par direction pour les couches cachées (comme Vial *et al.* (2019)).

Pour la méthode d'ensemble, nous avons expérimenté soit en l'utilisant, comme dans nos résultats principaux, c'est à dire en moyennant les prédictions de 8 modèles entraînés séparément, ou bien en donnant la moyenne et l'écart type des scores des 8 modèles évalués individuellement.

Comme nous pouvons le voir dans la table 5, le corpus d'entraînement supplémentaire (WNGT) et encore plus l'utilisation de BERT en tant que vecteurs de mots ont tous les deux un impact majeur sur nos résultats et conduisent à des scores supérieurs à l'état de l'art. L'utilisation de BERT au lieu de ELMo ou GloVe améliore respectivement le score d'environ 3 et 5 points dans chaque expérience, et l'ajout du WNGT aux données de d'entraînement l'améliore encore d'environ 2 points. Enfin, l'utilisation d'ensembles ajoute environ 1 point au score F1 final.

Enfin, à travers les scores obtenus par les modèles individuels (sans ensemble), nous pouvons observer sur les écarts-types que la méthode de compression du vocabulaire par les hyperonymes n'a jamais d'impact significatif sur le score final. Cependant, la méthode de compression via toutes les relations semble avoir un impact négatif sur les résultats dans certains cas (en utilisant GloVe et ELMo particulièrement, et en utilisant le SemCor seul comme corpus d'entraînement).

5 Conclusion

Dans cet article, nous avons présenté deux nouvelles méthodes qui améliorent la couverture et la capacité de généralisation des systèmes de DL supervisés, en réduisant le nombre d'étiquettes de sens différentes dans WordNet afin de ne conserver que celles qui sont essentielles pour différencier les sens de tous les mots présents dans la base lexicale. À l'échelle de l'ensemble de la base de données lexicale, nous avons montré que ces méthodes permettaient de réduire le nombre total d'étiquettes de sens différentes dans WordNet à seulement 6% de sa taille originale, et que la couverture d'un même corpus d'entraînement est ensuite plus que doublée.

Nous avons entraîné un système de DL neuronal état de l'art et nous avons montré que nos méthodes permettaient de réduire la taille des modèles par un facteur de 1,2 à 2 et de largement augmenter leur couverture, sans dégrader leurs performances. Au final, nous obtenons une couverture de 99,99% sur l'ensemble des tâches d'évaluation (soit un seul mot manquant sur les 7 253) lorsque l'on entraîne notre système sur le SemCor uniquement, et 100% lorsque l'on ajoute le WordNet Gloss Tagged aux données d'entraînement. On élimine ainsi quasiment le besoin d'une méthode de repli pour désambiguïser n'importe quel mot du vocabulaire de WordNet.

Notre méthode combinée avec les récentes avancées en terme de vecteurs de mots pré-entraînés permet à notre système de surpasser nettement l'état de l'art dans toutes les tâches d'évaluation de la DL de l'anglais, avec une bien meilleure couverture.

Pour finir, bien que nous ayons appliqué nos méthodes uniquement sur l'anglais dans cet article, elles peuvent facilement s'appliquer à d'autres langues en utilisant toujours WordNet comme inventaire de sens. Par exemple, elles peuvent s'appliquer à la désambiguïsation d'une langue moins bien dotée en utilisant la méthode de Hadj Salah *et al.* (2018). Cependant, du fait que les autres langues ont très peu de ressources annotées manuellement en sens (que ce soit avec l'inventaire de sens WordNet ou un autre), l'évaluation des systèmes de DL pour d'autres langues que l'anglais est limité.

8. https://nlp.stanford.edu/projects/glove/

Références

CHAN Y. S., NG H. T. & ZHONG Z. (2007). Nus-pt : Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, p. 253–256, Stroudsburg, PA, USA : Association for Computational Linguistics.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding.

EDMONDS P. & COTTON S. (2001). Senseval-2 : Overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, SENSEVAL '01, p. 1–5, Stroudsburg, PA, USA : Association for Computational Linguistics.

HADJ SALAH M., VIAL L., BLANCHON H., ZRIGUI M., LECOUTEUX B. & SCHWAB D. (2018). Traduction automatique de corpus en anglais annotés en sens pour la désambiguïsation lexicale d'une langue moins bien dotée, l'exemple de l'arabe. In *25e conférence sur le Traitement Automatique des Langues Naturelles*, Rennes, France.

IACOBACCI I., PILEHVAR M. T. & NAVIGLI R. (2016). Embeddings for word sense disambiguation : An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 897–907, Berlin, Germany : Association for Computational Linguistics.

KÅGEBÄCK M. & SALOMONSSON H. (2016). Word sense disambiguation using a bidirectional lstm. In *5th Workshop on Cognitive Aspects of the Lexicon (CogALex)* : Association for Computational Linguistics.

LE M., POSTMA M., URBANI J. & VOSSEN P. (2018). A deep dive into word sense disambiguation with lstm. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 354–365 : Association for Computational Linguistics.

LESK M. (1986). Automatic sense disambiguation using mrd : how to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC '86*, p. 24–26, New York, NY, USA : ACM.

LUO F., LIU T., HE Z., XIA Q., SUI Z. & CHANG B. (2018a). Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 1402–1411 : Association for Computational Linguistics.

LUO F., LIU T., XIA Q., CHANG B. & SUI Z. (2018b). Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2473–2482 : Association for Computational Linguistics.

MILLER G. A., BECKWITH R., FELLBAUM C., GROSS D. & MILLER K. (1990). Wordnet : An on-line lexical database. *International Journal of Lexicography*, **3**, 235–244.

MILLER G. A., LEACOCK C., TENGI R. & BUNKER R. T. (1993). A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, p. 303–308, Stroudsburg, PA, USA : Association for Computational Linguistics.

MORO A. & NAVIGLI R. (2015). Semeval-2015 task 13 : Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, p. 288–297, Denver, Colorado : Association for Computational Linguistics.

MORO A., RAGANATO A. & NAVIGLI R. (2014). Entity linking meets word sense disambiguation : a unified approach. *TACL*, **2**, 231–244.

NAVIGLI R., JURGENS D. & VANNELLA D. (2013). SemEval-2013 Task 12 : Multilingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, p. 222–231.

NAVIGLI R., LITKOWSKI K. C. & HARGRAVES O. (2007). Semeval-2007 task 07 : Coarse-grained english all-words task. In *SemEval-2007*, p. 30–35, Prague, Czech Republic.

OTT M., EDUNOV S., GRANGIER D. & AULI M. (2018). Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation : Research Papers*, p. 1–9, Belgium, Brussels : Association for Computational Linguistics.

PASINI T. & NAVIGLI R. (2017). Train-o-matic : Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 78–88 : Association for Computational Linguistics.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543.

PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

POLGUÈRE A. (2003). *Lexicologie et sémantique lexicale*. Les Presses de l'Université de Montréal.

PRADHAN S. S., LOPER E., DLIGACH D. & PALMER M. (2007). Semeval-2007 task 17 : English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, p. 87–92, Stroudsburg, PA, USA : Association for Computational Linguistics.

RAGANATO A., DELLI BOVI C. & NAVIGLI R. (2017). Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 1167–1178 : Association for Computational Linguistics.

SNYDER B. & PALMER M. (2004). The english all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

TAGHIPOUR K. & NG H. T. (2015). One Million Sense-Tagged Instances for Word Sense Disambiguation and Induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, p. 338–344, Beijing, China : Association for Computational Linguistics.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Eds., *Advances in Neural Information Processing Systems 30*, p. 5998–6008. Curran Associates, Inc.

VIAL L., LECOUTEUX B. & SCHWAB D. (2018). UFSAC : Unification of Sense Annotated Corpora and Tools. In *Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

VIAL L., LECOUTEUX B. & SCHWAB D. (2019). Approche supervisée à base de cellules LSTM bidirectionnelles pour la désambiguïsation lexicale. *Traitement Automatique des Langues*.

YUAN D., RICHARDSON J., DOHERTY R., EVANS C. & ALTENDORF E. (2016). Semi-supervised word sense disambiguation with neural models. In *COLING 2016*.

ZHONG Z. & NG H. T. (2010). It makes sense : A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, p. 78–83, Stroudsburg, PA, USA : Association for Computational Linguistics.

ZHU Y., KIROS R., ZEMEL R., SALAKHUTDINOV R., URTASUN R., TORRALBA A. & FIDLER S. (2015). Aligning books and movies : Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision (ICCV)*.