Approche supervisée à base de cellules *LSTM* bidirectionnelles 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, l'utilisation des réseaux de neurones est encore peu présente et très récente. Cette direction est pourtant très prometteuse, tant les résultats obtenus par ces premiers systèmes arrivent systématiquement en tête des campagnes d'évaluation, malgré une marge d'amélioration qui semble encore importante. Nous présentons dans cet article une nouvelle architecture à base de réseaux de neurones pour la désambiguïsation lexicale. Notre système est à la fois moins complexe à entraîner que les systèmes neuronaux existants et il obtient des résultats état de l'art sur la plupart des tâches d'évaluation de la désambiguïsation lexicale en anglais. L'accent est porté sur la reproductibilité de notre système et de nos résultats, par l'utilisation d'un modèle de vecteurs de mots, de corpus d'apprentissage et d'évaluation librement accessibles.

ABSTRACT

LSTM Based Supervised Approach for Word Sense Disambiguation

In word sense disambiguation, there are still few usages of neural networks. This direction is very promiseful however, the results obtained by these first systems being systematically in the top of the evaluation campaigns, with an improvement gap which seems still high. We present in this paper a new architecture based on neural networks for word sense disambiguation. Our system is at the same time less difficult to train than existing neural networks, and it obtains state of the art results on most evaluation tasks in English. The focus is on the reproducibility of our systems and our results, through the use of a word embeddings model, training corpora and evaluation corpora freely accessible.

MOTS-CLÉS : Désambiguïsation lexicale, Approche supervisée, LSTM, Réseau neuronal.

KEYWORDS: Word Sense Disambiguation, Supervised Approach, LSTM, Neural Network.

1 Introduction

La Désambiguïsation Lexicale (DL) est une tâche centrale en Traitement Automatique des Langues (TAL) qui vise à attribuer le sens le plus probable à un mot donné dans un document, à partir d'un inventaire prédéfini de sens.

Il existe une multitude d'approches pour la DL, dont les approches supervisées, qui utilisent des méthodes d'apprentissage automatique couplées à de grandes quantités de données manuellement annotées, les approches à base de connaissances, qui se basent sur des ressources lexicales telles que

des dictionnaires, des thésaurus ou des réseaux lexicaux par exemple, les approches semi-supervisées, non-supervisées, ou encore les approches à base de graphes ou de similarités. Pour un état de l'art plus complet, le lecteur est invité à lire par exemple Navigli (2009).

Depuis la création des campagnes d'évaluation pour les systèmes de DL telles que SensEval/SemEval, les approches supervisées se retrouvent systématiquement dans les premières places en terme de scores obtenus (Chan *et al.*, 2007; Zhong & Ng, 2010; Iacobacci *et al.*, 2016). Alors que l'on voit se multiplier les utilisations de techniques d'apprentissage à base de réseaux de neurones dans la plupart des champs de recherche du TAL, comme par exemple pour la représentation vectorielle des mots (Mikolov *et al.*, 2013; Pennington *et al.*, 2014; Bojanowski *et al.*, 2017), la traduction automatique (Sutskever *et al.*, 2014; Cho *et al.*, 2014) ou l'étiquetage morpho-syntaxique (Andor *et al.*, 2016), on retrouve aussi des approches supervisées à base de réseaux de neurones pour la désambiguïsation lexicale, et ce sont ces méthodes qui obtiennent aujourd'hui les résultats état de l'art (Yuan *et al.*, 2016; Kågebäck & Salomonsson, 2016; Raganato *et al.*, 2017b).

Dans cet article, nous présentons une nouvelle approche supervisée de DL à base de réseaux de neurones, qui s'appuie sur les modèles existants et qui obtient des résultats état de l'art sur la plupart des tâches d'évaluation de la DL en anglais tout en étant moins complexe et difficile à mettre en place. De plus, nous utilisons pour la première fois l'ensemble des corpus annotés avec des sens provenant de la base lexicale *WordNet* (Miller, 1995) qui existent à ce jour, ce qui permet à notre système d'être plus robuste car plus généralisable à de nouvelles données.

En effet, les systèmes supervisés de l'état de l'art sont généralement uniquement entraînés sur le SemCor (Miller *et al.*, 1993), mais une demi-douzaine d'autres corpus annotés en sens et de grande taille existent. Notre équipe les a tous regroupés dans une ressource libre nommée UFSAC [1] (Vial *et al.*, 2017). Par soucis de comparaison avec les systèmes état de l'art, nous avons évalué notre approche à la fois en utilisant tous les corpus UFSAC disponibles, mais aussi en nous restreignant uniquement au SemCor.

Dans un premier temps nous allons présenter les architectures des systèmes neuronaux de DL de l'état de l'art, avec leurs avantages et inconvénients respectifs dans la section 2, ainsi que l'architecture que nous proposons dans la section 3. Ensuite nous décrirons le protocole expérimental que nous avons suivi pour évaluer notre système dans la section 4 puis nous détaillerons les résultats dans la section 5. Enfin nous présenterons un travail préliminaire d'amélioration de notre système de manière totalement non supervisée dans la section 6 et enfin nous conclurons dans la section 7.

2 Architectures neuronales pour la désambiguïsation lexicale

Parmi les approches neuronales pour la DL, on retrouve notamment trois travaux majeurs : le modèle de Kågebäck & Salomonsson (2016), le modèle de Yuan *et al.* (2016) et celui de Raganato *et al.* (2017b).

Kågebäck & Salomonsson (2016) sont les premiers à mettre en œuvre un réseau de neurones à base de vecteurs de mots et de cellules récurrentes de type *LSTM* pour prédire le sens d'un mot cible. Dans leurs travaux, un modèle n'est capable de prédire le sens que d'un seul lemme du dictionnaire, et donc chaque lemme a son modèle propre de classification qui est entraîné séparément. Leur système est évalué sur les tâches de *lexical sample* des campagnes d'évaluation SensEval 2 et SensEval 3 dans

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

lesquelles plusieurs instances d'un faible nombre de lemmes distincts sont à annoter en sens, mais il n'est pas évalué sur les tâches de désambiguïsation lexicale *all words* où tous les mots d'un document doivent être annotés en sens.

Le principal avantage de leur modèle est donc sa petite taille. En effet la couche de sortie de leur réseau est de la taille du nombre de sens pour le lemme cible, le nombre de sens moyen pour les mots polysémiques dans WordNet étant d'environ 3 [2]. Les couches cachées de cellules *LSTM* sont elles aussi très petites, avec seulement deux couches de taille 74 chacune. Il est cependant peu aisé d'entraîner ce système à annoter tous les mots d'un document car chaque lemme doit avoir son propre modèle.

Dans le modèle de Yuan *et al.* (2016), un réseau neuronal à base de cellules *LSTM* est utilisé comme modèle de langue, pour prédire un mot d'une séquence en fonction de son contexte. Un apprentissage supervisé sur des corpus annotés en sens est ensuite effectué pour que leur système apprenne à distinguer les différents sens d'un mot en fonction des mots prédits par leur modèle de langue. Dans un second temps, les auteurs proposent une méthode de propagation de labels pour augmenter leurs données annotées en sens et obtenir ainsi leurs meilleurs résultats. Cette méthode consiste à chercher dans des corpus non annotés de nouvelles phrases, proches des phrases de leur corpus annoté, en se basant sur une mesure de similarité cosinus entre les représentations vectorielles de ces phrases. Les annotations en sens sont ensuite propagées de la phrase initialement annotée vers l'autre phrase.

Dans cet article, les auteurs comparent les performances de différents modèles, entraînés sur le SemCor ou l'OMSTI, avec et sans leur propagation de labels, et obtiennent des résultats état de l'art sur la plupart des tâches. Le principal problème de leur approche est la reproductibilité des résultats, en effet leur modèle de langue est entraîné sur un corpus privé d'actualités (*news*) d'une taille de 100 milliards de mots, et ils ont utilisé pour leur propagation de labels des phrases prises aléatoirement sur le Web, sans en spécifier la source plus précisément.

Enfin, l'architecture de leur modèle de langue ne permet de prédire le sens que d'un seul mot à la fois pour une séquence donnée, parce que le mot cible doit être remplacé par un symbole spécial avant d'être donné en entrée de leur réseau. Il est donc nécessaire d'exécuter leur modèle pour chaque mot d'une phrase afin de tous les annoter.

Raganato *et al.* (2017b) proposent également un modèle à base de *LSTM* mais qui apprend directement à prédire un label pour chacun des mots donnés en entrée. Le label à prédire fait partie d'un ensemble comprenant tous les sens possibles dans un dictionnaire ainsi que tous les mots observés pendant l'entraînement. Ils augmentent ensuite leur modèle avec une couche d'attention, et ils effectuent un entraînement multi-tâches dans lequel leur réseau prédit à la fois un sens ou un mot, un label de partie du discours, et un label sémantique.

Cette architecture est la seule qui permet d'annoter tous les mots d'une séquence en une passe et l'entraînement de leur modèle s'est effectué sur le SemCor uniquement. Leur réseau associe à un mot en entrée un label appartenant à l'ensemble des sens de leur inventaire de sens ainsi que l'ensemble des mots observés pendant l'entraînement. Cette approche permet à leur modèle d'apprendre à prédire un label de sens lorsque le mot est annoté dans le corpus d'entraînement, et un label de mot lorsque le mot n'est pas annoté (si c'est un mot outil par exemple). L'inconvénient de leur approche est qu'elle n'est pas applicable lorsque l'on veut réaliser l'apprentissage sur un corpus partiellement annoté en sens. En effet pour ce type de corpus, leur modèle va apprendre à "recopier" des mots non annotés alors qu'ils sont potentiellement porteurs de sens.

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

3 Architecture proposée

Notre approche est, comme pour Raganato *et al.* (2017b), de considérer la désambiguïsation lexicale comme un problème de classification dans lequel un label est assigné à chaque mot. Cependant, nous simplifions leur modèle en considérant un label comme appartenant uniquement à l'ensemble de tous les sens possibles de notre inventaire de sens. L'architecture de notre réseau de neurones, illustrée par la figure 1 repose ainsi sur 3 couches de cellules :

— La couche d'entrée, qui prend directement les mots sous une forme vectorielle construite séparément de notre système. On pourra utiliser ici n'importe quelle base de vecteurs de mots pré-entraînés telle que Word2Vec (Mikolov *et al.*, 2013) ou GloVe (Pennington *et al.*, 2014).

— La couche cachée, composée de cellules LSTM (Hochreiter & Schmidhuber, 1997) bidirectionnelles. Ces cellules dites "à mémoire" aussi appelées cellules "récurrentes" permettent de calculer une sortie en considérant non seulement l'élément courant de la séquence, mais aussi l'historique passé des cellules précédentes. Ces cellules sont communément utilisées pour l'apprentissage automatique sur des séquences, que ce soit sur du texte écrit (Sutskever *et al.*, 2014) ou de la parole (Chan *et al.*, 2016).

— La couche de sortie, qui génère pour chacun des mots en entrée, une distribution de probabilité sur tous les sens possibles du dictionnaire, à l'aide d'une fonction softmax classique.

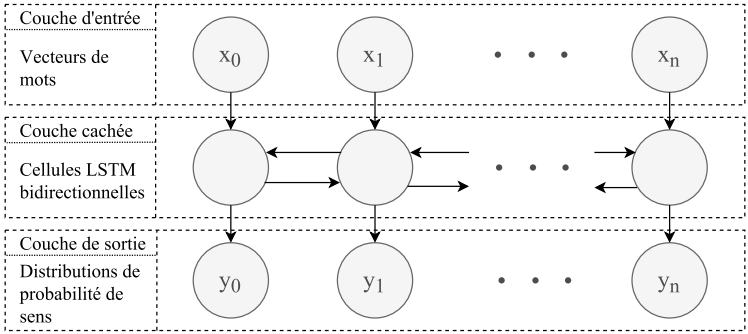

FIGURE 1 – Architecture de notre réseau de neurones pour la DL.

La fonction de coût à minimiser pendant la phase d'apprentissage est l'entropie croisée entre la couche de sortie et un vecteur de type *one-hot*, pour lequel toutes les composantes sont à 0 sauf à l'index du sens cible où elle est à 1. On cherche ainsi à minimiser la fonction $H(p,q) = -\sum_x p(x) \log q(x)$, où x est est une composante du vecteur de la couche de sortie, p est la distribution de probabilité réelle et q la sortie de notre réseau de neurones. Comme toutes les valeurs de la distribution réelle sont à 0 sauf à l'index du sens correct, pour un exemple donné, on cherche ainsi à minimiser la formule $-\log q(s)$, où s est l'index du sens à prédire.

Notre modèle prédit toujours un sens en sortie pour chaque mot en entrée, même pour les mots outils ou les mots qui n'ont pas été annotés dans le corpus d'entraînement, cependant, dans ces cas là, nous avons un symbole spécial <*skip*> nous permettant d'ignorer les prédictions faites par le modèle et de ne pas en tenir compte lors de la phase de rétro-propagation durant l'entraînement.

Contrairement à l'approche proposée par Raganato *et al.* (2017b), notre modèle peut ainsi apprendre non seulement sur des données entièrement annotées, comme c'est le cas avec le SemCor (Miller

et al., 1993) par exemple, mais également sur des données partiellement annotées, comme l'OMSTI (Taghipour & Ng, 2015) ou le DSO (Ng & Lee, 1997), dans lesquelles un seul mot est annoté par phrase. Il est en effet capable d'apprendre à prédire les sens de tous les mots d'une séquence en même temps, et à la fois d'en ignorer certains éléments. L'entraînement se retrouve aussi moins complexe à réaliser que pour Raganato *et al.* (2017b) car la taille de la couche de sortie est beaucoup plus petite : le nombre de sens différents dans la version 3.0 de *WordNet* est de 117 659 [3], alors qu'une taille de vocabulaire typique pour des modèles de vecteurs de mots en anglais contient au minimum 400 000 mots et plus généralement plus de 1 000 000 de mots [4] [5].

Notre architecture est aussi très différente de celles de Yuan *et al.* (2016) ou de Kågebäck & Salomonsson (2016), notamment car leurs architectures ne permettent pas d'annoter tous les mots en entrée de leurs modèles en une seule passe, mais seulement indépendamment les uns des autres.

4 Protocole expérimental

Pour évaluer notre système de DL à base de réseaux de neurones, nous avons tiré parti de notre précédent travail (Vial *et al.*, 2017) dans lequel nous proposions une ressource contenant tous les corpus anglais annotés en sens *WordNet* connus à ce jour, et nous avons entraîné notre modèle sur 6 de ces corpus : le SemCor (Miller *et al.*, 1993), le DSO (Ng & Lee, 1997), le WordNet Gloss Tagged (Miller, 1995), l'OMSTI (Taghipour & Ng, 2015), le MASC (Ide *et al.*, 2008) et l'Ontonotes (Hovy *et al.*, 2006). Nous avons utilisé le corpus de la tâche 13 de SemEval 2015 (Moro & Navigli, 2015) comme corpus de développement durant l'apprentissage, pour éviter le surapprentissage de nos données d'entraînement. Enfin, nous avons évalué le modèle ayant obtenu le meilleur score F1 de DL sur notre corpus de développement, sur les corpus de SensEval 2 (Edmonds & Cotton, 2001), SensEval 3 (Snyder & Palmer, 2004), les tâches 7 et 17 de SemEval 2007 (Navigli *et al.*, 2007; Pradhan *et al.*, 2007), et enfin la tâche 12 de SemEval 2013 (Navigli *et al.*, 2013).

Pour comparer l'architecture que nous proposons avec l'état de l'art, et notamment Raganato *et al.* (2017b) et Yuan *et al.* (2016) qui utilisent uniquement le SemCor comme corpus d'apprentissage supervisé, nous avons aussi évalué notre approche en limitant l'apprentissage du modèle à ce corpus.

Dans certains corpus, les mots peuvent être annotés avec plusieurs sens *WordNet*, soit parce que l'annotateur a trouvé qu'ils étaient tous applicables, ou bien parce que les sens ont été initialement annotés avec un autre dictionnaire puis convertis en sens *WordNet* (c'est le cas du MASC par exemple). Dans ce cas nous supprimons toutes les annotations pour ne garder au final que les annotations qui ne contiennent qu'un seul sens dans notre corpus d'apprentissage.

En entrée de notre réseau, nous avons utilisé les vecteurs de GloVe (Pennington *et al.*, 2014) pré-entrainés sur Wikipedia 2014 et Gigaword 5 disponibles librement [6]. La taille des vecteurs est de 300, la taille du vocabulaire est de 400 000 et tous les mots sont mis en minuscules. Nous avons choisi ces vecteurs pour la petite taille de leur modèle pré-entraîné et pour sa qualité par rapport aux tâches de similarité de mots et d'analogie de mots. Ce sont aussi ces vecteurs qui sont utilisés en entrée du réseau décrit par Kågebäck & Salomonsson (2016).

3. https://wordnet.princeton.edu/documentation/wnstats7wn
4. https://nlp.stanford.edu/projects/glove/
5. https://fasttext.cc/docs/en/english-vectors.html
6. https://nlp.stanford.edu/projects/glove/

Pour la couche cachée de neurones récurrents, nous avons choisi des cellules *LSTM* de taille de 1000 par direction (donc 2000 au total). C'est à peu près la taille qui est utilisée dans Raganato *et al.* (2017b) (chaque *LSTM* est de taille 1024) et Yuan *et al.* (2016) (une seule couche de taille 2048).

Enfin, entre la couche cachée et la couche de sortie, nous avons appliqué une régularisation de type *Dropout* (Srivastava *et al.*, 2014) à 50%, une méthode classique qui vise à empêcher le surapprentissage pendant l'entraînement afin de rendre le modèle plus robuste.

Cette configuration permet de reproduire aisément nos résultats. En effet, en plus du modèle de vecteurs de mots pré-entraîné, tous les corpus utilisés sont libres d'accès et dans un format unifié [7]. La seule exception est le corpus DSO qui est payant, il ne contient cependant qu'approximativement 8% des mots annotés dans nos corpus d'apprentissage, avec seulement 121 noms et 70 verbes différents.

Les paramètres utilisés pour l'apprentissage sont les suivants :

— La méthode d'optimisation est Adam (Kingma & Ba, 2014), avec les mêmes paramètres par défaut tels que décrits dans leur article, c'est à dire $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ et $\epsilon = 10^{-8}$;

— la taille de mini-lots utilisée est de 30 ;

— les phrases sont tronquées à 50 mots, pour faciliter l'entraînement tout en minimisant la perte d'informations (moins de 5% des mots annotés dans nos données d'entraînement sont perdus) ;

— les séquences sont remplies de vecteurs nuls depuis la fin de façon à ce qu'elles aient toutes la même taille au sein d'un mini-lot.

Nous avons construit notre réseau neuronal à l'aide de l'outil *PyTorch* [8] et nous avons effectué l'apprentissage pendant 20 *epochs*. Une *epoch* correspondant à une passe complète sur nos données d'entraînement. Nous avons évalué périodiquement (tous les 2000 mini-lots et à la fin de chaque *epoch*) notre modèle sur le corpus de développement, et nous avons conservé uniquement le modèle ayant obtenu le plus grand score F1 de désambiguïsation.

Pour réaliser la désambiguïsation d'une séquence de mots en utilisant le réseau entraîné, la méthode suivante est utilisée :

1. Chaque mot est d'abord transformé en vecteur à l'aide du modèle de vecteurs de mots, puis donné en entrée au réseau.
2. En sortie, une distribution de probabilité sur tous les sens observés pendant l'apprentissage est retournée pour chaque élément de la séquence. Nous assignons le sens le plus probable en suivant cette distribution, parmi les sens possibles du mot dans *WordNet*, en fonction de son lemme et de sa partie du discours. Ces deux informations étant systématiquement données pendant les campagnes d'évaluation de la DL.
3. Si aucun sens n'est assigné, une stratégie de repli est effectuée. La plus courante et celle que nous utilisons ici est d'assigner au mot son sens le plus fréquent dans *WordNet*.

Le processus d'apprentissage est forcément stochastique, en effet non seulement les poids du modèle sont initialisés aléatoirement par la bibliothèque sous-jacente, mais le corpus d'apprentissage est également mélangé à chaque début d'*epoch*. Nous avons entraîné ainsi 8 modèles séparément pour chacun de nos tests, puis nous avons utilisé une moyenne géométrique sur toutes les prédictions faites par ces modèles pour obtenir la distribution de sens finale que nous avons utilisée pour réaliser une désambiguïsation. C'est une pratique couramment utilisée (par exemple (Sutskever *et al.*, 2014)) car elle permet non seulement d'avoir un système moins sensible au bruit et donc plus robuste, mais

7. https://github.com/getalp/UFSAC
8. http://pytorch.org/

aussi un système de meilleure qualité. En effet, un modèle peut être individuellement bloqué dans un minimum local pendant l'entraînement et avoir un très bon score sur le corpus de développement, mais être incapable de généraliser, alors qu'il est improbable que ce problème arrive à l'ensemble de modèles.

5 Résultats

Nous avons évalué notre modèle sur tous les corpus d'évaluation communément utilisés en DL, à savoir les tâches de DL des campagnes d'évaluation SensEval/SemEval. Les scores obtenus par notre système comparés à ceux des systèmes semblables de l'état de l'art à base de réseaux de neurones (Yuan *et al.*, 2016; Raganato *et al.*, 2017b), ainsi que l'étalon du sens le plus fréquent, et du meilleur système précédant l'utilisation des réseaux de neurones en DL (Iacobacci *et al.*, 2016) se trouvent dans la table 1.

Système	SE2	SE3	SE07 (07)	SE07 (17)	SE13 (12)	SE15 (13)
Notre système (SemCor)	68.27	69.95	80.14	60.51	62.97	*69.72
Notre système (SemCor + repli)	73.71	**71.68**	**83.99**	**61.98**	67.58	*72.74
Notre système (UFSAC)	72.54	69.46	82.87	59.85	67.53	*73.56
Notre système (UFSAC + repli)	**73.75**	70.16	83.59	60.00	**68.92**	***73.98**
Yuan *et al.* (2016) (LSTM)	73.6	69.2	82.8	**64.2**	67.0	72.1
Yuan *et al.* (2016) (LSTM + LP)	**73.8**	**71.8**	**83.6**	63.5	**69.5**	**72.6**
Raganato *et al.* (2017b) (BLSTM)	71.4	68.8	-	*61.8	65.6	69.2
Raganato *et al.* (2017b) (BLSTM + att. + LEX + POS)	72.0	69.1	83.1	*64.8	66.9	71.5
Sens le plus fréquent	65.6	66.0	78.89	54.5	63.8	67.1
Iacobacci *et al.* (2016)	68.3	68.2	-	59.1	-	-

TABLE 1 – Scores F1 (%) obtenus par notre système sur les tâches de DL des campagnes d'évaluation SensEval 2 (SE2), SensEval 3 (SE3), SemEval 2007 (SE07) tâches 07 et 17, SemEval 2013 (SE13) tâche 12 et SemEval 2015 (SE15) tâche 13. Les résultats préfixés par un astérisque (*) sont obtenus sur le corpus utilisé pour le développement pendant l'apprentissage. Les résultats affichés en gras sont les meilleurs obtenus par notre système et par les systèmes de l'état de l'art. Les résultats affichés en rouge sont les meilleurs de l'état de l'art.

Pour toutes les tâches, nous avons évalué notre système avec et sans le repli sur le premier sens pour les mots qui n'ont pas été observés pendant l'apprentissage. Nous l'avons aussi évalué dans un premier temps avec un apprentissage sur le SemCor uniquement, et sur les 6 corpus UFSAC combinés dans un second temps.

On remarque d'abord qu'en termes de scores F1 avec le repli, il y a très peu de différences entre le système entraîné sur le SemCor et celui entraîné sur tous les corpus UFSAC. Nos meilleurs résultats sur les tâches de SensEval 3 et de SemEval 2007 sont même obtenus par le système qui est entraîné sur le SemCor uniquement. Le SemCor possède pourtant seulement environ 10% des mots annotés dans UFSAC.

Cependant, lorsque l'on compare les scores de désambiguïsation d'un système avec repli et sans

repli, la différence entre ces deux scores est bien plus grande avec le système entraîné sur le SemCor qu'avec celui entraîné sur UFSAC. Ceci s'explique par la couverture du SemCor qui est moins importante que celle de tous les corpus UFSAC réunis. Pour le système appris sur le SemCor, la couverture est en effet de 91% sur SensEval 2, 97% sur SensEval 3, 93% sur SemEval 2007 (07), 98% sur SemEval 2007 (17), 91% sur SemEval 2013 et 95% sur SemEval 2015. Pour celui appris sur tout UFSAC, la couverture est respectivement de 98%, 99%, 99%, 99%, 98% et 99%.

Ces résultats démontrent la grande qualité du SemCor, c'est en effet lorsque sa couverture sur les tâches d'évaluation est la plus proche de 100% que notre système appris sur ce seul corpus obtient les meilleurs résultats. Les autres corpus UFSAC permettent quand même d'annoter un bien plus grand nombre de sens sans stratégie de repli, et nos meilleurs résultats sur SensEval 2, SemEval 2013 et SemEval 2015 sont obtenus avec le système appris sur tous les corpus réunis.

Le premier système de Yuan *et al.* (2016) obtient des résultats comparables aux nôtres mais comme nous l'avons souligné dans la section 2, le caractère privé de leur corpus d'entraînement contenant 100 milliards de mots pour leur modèle de langue rend très difficile la reproductibilité de leurs résultats.

Leur deuxième système (LSTM + LP) ajoute une étape de propagation de labels, dans laquelle ils augmentent automatiquement leurs données d'entraînement annotées en sens, en recherchant dans une grande quantité de textes non annotés des phrases similaires aux phrases annotées, et en portant les labels de sens depuis les phrases annotées, vers les phrases non annotées. Cette méthode apporte de meilleurs résultats sur la plupart des tâches, cependant ils récupèrent, pour leurs données non annotées, 1000 phrases prises aléatoirement sur le Web pour chaque lemme, sans plus de précisions, ce qui rend la reproductibilité des résultats encore plus difficile.

Le système de Raganato *et al.* (2017b) qui est quant à lui très semblable au nôtre obtient des résultats moins élevés malgré une plus grande complexité de leur modèle, et ils utilisent 2 couches de cellules LSTM bidirectionnelles de taille 2048 (1024 par direction), donc un total de 4096 unités cachées, ce qui est deux fois plus que notre modèle.

Pour leur second système (BLTM + att. + LEX + POS), les auteurs ont ajouté une couche d'attention à leur réseau, et ils effectuent de l'apprentissage multi-tâches, c'est à dire que leur réseau apprend à la fois à prédire un label de mot ou de sens, ainsi que la partie du discours (POS) du mot, et son label sémantique dans WordNet (LEX), la tâche est rendue ainsi plus complexe.

En comparaison avec ces autres systèmes, les nôtres obtiennent des scores supérieurs à ceux de Raganato *et al.* (2017b) dans la majorité des cas, malgré une complexité réduite au niveau de l'architecture. Nous obtenons des scores similaires ou légèrement inférieurs à ceux de Yuan *et al.* (2016) mais en utilisant largement moins de données pour l'apprentissage, et surtout des données librement accessibles.

Enfin, on voit que tous les systèmes supervisés à base de réseaux de neurones surpassent le système de Iacobacci *et al.* (2016) là où il a été évalué. Cette approche combinant des classifieurs linéaires de type *SVM* et des traits à base de vecteurs de mots obtenait pourtant des résultats état de l'art avant l'arrivée des systèmes neuronaux.

6 Vers une amélioration non supervisée

Dans cette section, nous présentons une première approche visant l'amélioration de notre système de manière complètement non supervisée, en s'appuyant sur des corpus non annotés en sens. Nous mettons ainsi en avant des pistes qui pourraient être approfondies dans de futurs travaux.

6.1 Approche

L'approche que nous avons suivie est en partie inspirée de la méthode de propagation de labels de Yuan *et al.* (2016), dans laquelle les auteurs transfèrent des annotations de sens de leur corpus manuellement annoté vers des phrases non annotées, pour étendre leurs données d'apprentissage.

Notre approche est aussi et surtout inspirée des méthodes d'apprentissage par transfert et apprentissage par mimétisme telles que Kim & Kim (2017); Buciluǎ *et al.* (2006); Hinton *et al.* (2015), dans lesquelles un ou plusieurs modèles "enseignant" vont transférer leurs connaissances à un modèle "élève" en lui montrant comment effectuer une tâche. L'élève va ainsi apprendre à recopier ce que font les enseignants, observer des exemples dans de nouveaux contextes et ainsi apprendre à mieux généraliser.

Dans le contexte de la DL et donc dans notre approche, les modèles enseignants sont des modèles capables d'annoter n'importe quelle séquence de mots en sens, et le modèle élève sera un nouveau modèle qui va être entraîné sur des données produites par les enseignants.

Plus particulièrement, nous avons utilisé comme modèle enseignant le système de DL qui a obtenu le meilleur score F1 (voir la Table 1) sur notre corpus de développement uniquement (SemEval 2015) afin d'éviter tout biais, c'est à dire celui entraîné sur toutes les données UFSAC avec la stratégie de repli.

Nous avons annoté avec ce système un million de phrases prises sur les données anglaises monolingues des campagnes d'évaluation de la traduction automatique WMT, et plus précisément le premier million de phrases du corpus "News Crawl 2016" accessible sur le site de la campagne d'évaluation WMT17 [9].

Ensuite, nous avons entraîné un nouveau modèle avec la même architecture sur ces données automatiquement annotées, en suivant le même protocole décrit dans la section 4, puis nous avons conservé l'ensemble de poids qui obtenait le meilleur score F1 sur le corpus de développement.

Enfin, nous avons poursuivi l'entraînement de ce modèle initialisé avec cet ensemble de poids mais cette fois ci sur les corpus UFSAC manuellement annotés, toujours pendant 20 *epochs* et en conservant le modèle avec le meilleur score sur le corpus de développement. Cependant pour cette dernière phase, le modèle a convergé très rapidement et obtenu ce meilleur score au bout d'environ une à deux *epoch*, ceci parce qu'il avait été pré-entraîné sur les données automatiquement annotées.

Nous avons réitéré cette dernière étape jusqu'à obtenir 8 modèles différents afin d'évaluer cette méthode, comme pour le système original, en moyennant les prédictions d'un ensemble de modèles.

9. `http://data.statmt.org/wmt17/translation-task/news.2016.en.shuffled.gz`

6.2 Résultats

Nous avons évalué le système “élève” sur les mêmes tâches que pour la section 5, avec et sans repli, et nous avons comparé ses scores avec ceux obtenus par le système “enseignant”, et avec le système état de l’art de Yuan *et al.* (2016). Les résultats sont dans la Table 2.

Système	SE2	SE3	SE07 (07)	SE07 (17)	SE13 (12)	SE15 (13)
Système “élève” (UFSAC + 1M News 2016)	73.03	68.48	84.12	60.95	68.57	*74.13
Système “élève” (UFSAC + 1M News 2016 + repli)	**74.23** (+0.48)	69.19 (-0.97)	**84.83** (+1.24)	**61.10** (+1.10)	**69.95** (+1.03)	***74.55** (+0.57)
Système “enseignant” (UFSAC + repli)	73.75	**70.16**	83.59	60.00	68.92	*73.98
Yuan *et al.* (2016) (LSTM)	73.6	69.2	82.8	64.2	67.0	72.1
Yuan *et al.* (2016) (LSTM + LP)	73.8	71.8	83.6	63.5	69.5	72.6

TABLE 2 – Scores F1 (%) obtenus par le système “élève” sur les tâches de DL des campagnes d’évaluation SensEval 2 (SE2), SensEval 3 (SE3), SemEval 2007 (SE07) tâches 07 et 17, SemEval 2013 (SE13) tâche 12 et SemEval 2015 (SE15) tâche 13. Les résultats préfixés par un astérisque (*) sont obtenus sur le corpus utilisé pour le développement pendant l’apprentissage. La différence entre le système élève (avec repli) et le système enseignant est affichée entre parenthèses. Le meilleur score entre l’élève et l’enseignant est affiché en gras, et le meilleur score de l’état de l’art est affiché en rouge.

Sur toutes les tâches d’évaluation, à part celle de SensEval 3, le système élève obtient ainsi des scores significativement supérieurs à ceux du système enseignant, et il obtient même des scores surpassant l’état de l’art sur les tâches de SensEval 2, SemEval 2007 (07), SemEval 2013 et SemEval 2015.

À travers ces résultats, on peut voir à quel point la mise en place de ce type d’apprentissage par transfert de connaissances peut s’avérer efficace pour la construction d’un système de DL robuste et de bonne qualité. Notre système ainsi entraîné obtient en effet des scores supérieurs à notre système original et à l’état de l’art sur la plupart des tâches d’évaluation, alors que nous avons uniquement utilisé comme ressource supplémentaire un million de phrases en anglais non annotées provenant d’un corpus en libre accès.

Cette approche est un premier pas pour l’amélioration du système de DL basé sur notre architecture neuronale sans utiliser de données annotées manuellement supplémentaires, et elle aide effectivement notre système à mieux généraliser, mais elle souffre encore de défauts évidents, en témoigne la baisse de résultats sur la tâche de SensEval 3.

Parmi les points que nous prévoyons d’améliorer nous souhaitons entre autres :

— une sélection plus fine des données à annoter par le système enseignant, plutôt que de prendre un million de phrases d’un corpus de *news* aléatoires, s’adapter au domaine de la tâche sur laquelle on souhaite s’évaluer ;

— une sélection des annotations produites par le système enseignant, pour éviter de reproduire les erreurs du modèle neuronal qui peuvent être facilement détectées, par exemple à l’aide d’une mesure de confiance basée sur sa couche de sortie.

7 Conclusion

Nous présentons dans cet article une nouvelle architecture de réseau neuronal pour la désambiguïsation lexicale à base de cellules *LSTM*. Les *LSTM* sont des cellules récurrentes largement utilisées dans les réseaux de neurones traitant des séquences tels que les systèmes *sequence-to-sequence* pour la traduction automatique ou les systèmes utilisant un modèle de langue prédisant la prochaine entrée d'une suite de mots. Notre modèle est composé d'une couche d'entrée qui prend une séquence de vecteurs de mots construits séparément, il a ensuite une couche cachée de cellules *LSTM* bidirectionnelles, et enfin il possède une couche de sortie entièrement connectée de la taille du nombre de sens possibles dans le dictionnaire utilisé. Ce modèle se distingue de ceux existants dans l'état de l'art par le fait qu'il permet d'annoter tous les mots d'une séquence donnée en une seule passe, contrairement à Yuan *et al.* (2016) et Kågebäck & Salomonsson (2016), pour lesquels chaque mot et chaque lemme est traité indépendamment. Il est aussi moins complexe et moins difficile à entraîner que celui de Raganato *et al.* (2017b).

Nous avons entraîné un système sur six corpus au format UFSAC (Vial *et al.*, 2017), à savoir le SemCor, le DSO, le WNGT, l'OMSTI, le MASC et l'Ontonotes, mais aussi un système sur le SemCor uniquement, et nous les avons évalués sur les tâches de DL des campagnes d'évaluation SensEval/SemEval. Les résultats montrent que nos systèmes obtiennent des scores équivalents à ceux des meilleurs systèmes neuronaux de l'état de l'art. Seul le système de Yuan *et al.* (2016) augmenté par les données issues de leur propagation de labels obtient des scores plus élevés. Cette augmentation indépendante de leur architecture neuronale est cependant basée sur l'utilisation de grandes quantités de textes pris aléatoirement sur le web, ce qui rend la reproductibilité difficile.

Nous avons ensuite présenté une amélioration de notre système à l'aide d'une approche par transfert de connaissances pour laquelle seulement un million de phrases initialement non annotées étaient ajoutées aux données d'entraînement afin d'obtenir un modèle plus robuste et performant. Nous avons présenté des résultats avec ce système qui surpassent significativement l'état de l'art sur toutes les tâches d'évaluation de la DL hormis deux, et nous avons proposé quelques pistes d'amélioration futures pour continuer dans cette voie.

Les études sur les systèmes à base de réseaux de neurones pour la désambiguïsation lexicale sont encore très récentes en atteste le faible nombre de systèmes existants pour le moment. C'est cependant une direction prometteuse, tant les résultats obtenus par ces nouveaux systèmes ont montré leur qualité sur les campagnes d'évaluation, dépassant les meilleurs systèmes non neuronaux. Dans le même temps, les récents travaux comme Raganato *et al.* (2017a) ou Vial *et al.* (2017) facilitent la création et l'évaluation rigoureuse de nouveaux systèmes de DL, étant donné que toutes les ressources annotées en sens *WordNet* sont disponibles librement et dans un format unifié.

Références

ANDOR D., ALBERTI C., WEISS D., SEVERYN A., PRESTA A., GANCHEV K., PETROV S. & COLLINS M. (2016). Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2442–2452, Berlin, Germany : Association for Computational Linguistics.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, **5**, 135–146.

BUCILUǍ C., CARUANA R. & NICULESCU-MIZIL A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 535–541 : ACM.

CHAN W., JAITLY N., LE Q. V. & VINYALS O. (2016). Listen, attend and spell : A neural network for large vocabulary conversational speech recognition. In *ICASSP*.

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.

CHO K., VAN MERRIENBOER B., BAHDANAU D. & BENGIO Y. (2014). On the properties of neural machine translation : Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 103–111 : Association for Computational Linguistics.

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.

HINTON G., VINYALS O. & DEAN J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

HOVY E., MARCUS M., PALMER M., RAMSHAW L. & WEISCHEDEL R. (2006). Ontonotes : The 90In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume : Short Papers*, NAACL-Short '06, p. 57–60, Stroudsburg, PA, USA : Association for Computational Linguistics.

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.

IDE N., BAKER C., FELLBAUM C., FILLMORE C. & PASSONNEAU R. (2008). Masc : the manually annotated sub-corpus of american english. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco : European Language Resources Association (ELRA). http ://www.lrec-conf.org/proceedings/lrec2008/.

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.

KIM S. W. & KIM H.-E. (2017). Transferring knowledge to smaller network with class-distance loss.

KINGMA D. P. & BA J. (2014). Adam : A method for stochastic optimization. *CoRR*, **abs/1412.6980**.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In C. BURGES, L. BOTTOU, M. WELLING, Z. GHAHRAMANI & K. WEINBERGER, Eds., *Advances in Neural Information Processing Systems 26*, p. 3111–3119. Curran Associates, Inc.

MILLER G. A. (1995). Wordnet : A lexical database. *ACM*, **Vol. 38**(No. 11), p. 1–41.

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.

NAVIGLI R. (2009). Wsd : a survey. *ACM Computing Surveys*, **41**(2), 1–69.

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.

NG H. T. & LEE H. B. (1997). Dso corpus of sense-tagged english.

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.

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., CAMACHO-COLLADOS J. & NAVIGLI R. (2017a). Word sense disambiguation : A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 99–110, Valencia, Spain : Association for Computational Linguistics.

RAGANATO A., DELLI BOVI C. & NAVIGLI R. (2017b). 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*.

SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**(1), 1929–1958.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, p. 3104–3112, Cambridge, MA, USA : MIT Press.

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.

VIAL L., LECOUTEUX B. & SCHWAB D. (2017). *UFSAC : Unification of Sense Annotated Corpora and Tools*. Research report, UGA - Université Grenoble Alpes.

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.