Création et validation de signatures sémantiques : application à la mesure de similarité sémantique et à la substitution lexicale

Mokhtar Boumedyen Billami, Núria Gala

Création et validation de signatures sémantiques : application à la mesure de similarité sémantique et à la substitution lexicale

Mokhtar Boumedyen Billami[1] Núria Gala[2]
(1) LIF-CNRS UMR 7279, Aix Marseille Université
(2) LPL-CNRS UMR 7309, Aix Marseille Université
mokhtar.billami@lif.univ-mrs.fr, nuria.gala@univ-amu.fr

RÉSUMÉ

L'intégration de la notion de similarité sémantique entre les unités lexicales est essentielle dans différentes applications de Traitement Automatique des Langues (TAL). De ce fait, elle a reçu un intérêt considérable qui a eu comme conséquence le développement d'une vaste gamme d'approches pour en déterminer une mesure. Ainsi, plusieurs types de mesures de similarité existent, elles utilisent différentes représentations obtenues à partir d'informations soit dans des ressources lexicales, soit dans de gros corpus de données ou bien dans les deux. Dans cet article, nous nous intéressons à la création de signatures sémantiques décrivant des représentations vectorielles de mots à partir du réseau lexical JeuxDeMots (JDM). L'évaluation de ces signatures est réalisée sur deux tâches différentes : mesures de similarité sémantique et substitution lexicale. Les résultats obtenus sont très satisfaisants et surpassent, dans certains cas, les performances des systèmes de l'état de l'art.

ABSTRACT

Creating and validating semantic signatures : application for measuring semantic similarity and lexical substitution.

The integration of semantic similarity between lexical units is at present essential in various applications of natural language processing (NLP). It received a considerable amount of research interest, which in its turn led to a vast range of approaches for mesuring semantic similarity. However, there are several types of measuring similarity that use various representations from lexical resources or large corpus of data or both. In this paper, we are interested in creating semantic signatures that describing vectorial representations of words with the lexical network JEUXDEMOTS (JDM). The evaluation of these signatures is carried out on two different tasks : measuring semantic similarity and lexical substitution. The results are very significant and in some cases they surpass performances of the state-of-the-art systems.

MOTS-CLÉS : similarité sémantique, sémantique lexicale, ressources lexicales, substitution lexicale.

KEYWORDS: semantic similarity, lexical semantic, lexical resources, lexical substitution.

1 Introduction

Mesurer la similarité sémantique entre les mots est essentielle pour de nombreuses applications du traitement automatique des langues telles que la substitution lexicale (Fabre *et al.*, 2014; McCarthy & Navigli, 2009), la simplification lexicale (Biran *et al.*, 2011) ou l'enrichissement sémantique de

requêtes (Voorhees, 1994). Les mesures de similarité sémantique sont également essentielles pour les sens ou les textes. Pour les sens, elles peuvent être utilisées pour la désambiguïsation sémantique (Navigli, 2009) ou l'alignement et l'intégration de différentes ressources lexicales (Matuschek & Gurevych, 2013). Pour les textes, elles permettent par exemple d'évaluer la qualité des sorties des systèmes de traduction automatique (Lavie & Denkowski, 2009) ou de recherche d'information (Otegi *et al.*, 2015). Des travaux existants ont montré qu'il est possible d'ajuster ou d'étendre des approches utilisées pour un niveau de granularité à un autre. Par exemple, les mesures au niveau du mot ont été ajustées pour mesurer la similarité entre les textes (Corley & Mihalcea, 2005) alors que les mesures au niveau du sens ont été étendues au niveau du mot en supposant que la similarité entre deux mots est celle de leurs sens les plus proches (Budanitsky & Hirst, 2006). Dans cet article, nous nous intéressons spécifiquement à la création de représentations sémantiques pour les mots.

Le travail présenté décrit une méthode qui s'appuie tout d'abord sur les propriétés individuelles de chaque élément linguistique qui se définit comme étant un nœud dans le réseau lexical JeuxDeMots (Lafourcade, 2007), cela afin de modéliser des nœuds liés à travers une représentation sémantique. Ce qui nous ramène par la suite à comparer les éléments linguistiques en termes de leurs représentations. Ces dernières sont typées, pondérées et appelées des signatures sémantiques. Elles peuvent être utilisées, par exemple, pour déterminer si la similarité sémantique entre FOURNAISE et FOUR est plus forte que la similarité entre FOURNAISE et INSTRUMENT ou pour savoir si le synonyme PRIX du mot INTÉRÊT représente un substitut pertinent par rapport au synonyme AVANTAGE dans un contexte décrivant le sens *finance* pour le mot INTÉRÊT.

D'une manière générale, une signature sémantique peut être considérée comme une forme spéciale de représentation du modèle d'espace vectoriel (VSM, *Vector Space Model*) (Turney & Pantel, 2010). De la même façon que la représentation d'un élément linguistique à base du modèle VSM, le poids associé à une dimension dans une signature sémantique indique la pertinence ou l'importance de cette dimension pour l'élément linguistique. La différence principale est dans la manière dont les poids sont calculés. Dans une représentation à base du modèle VSM, chaque dimension correspond habituellement à un mot individuel dont le poids est souvent calculé sur les bases de la statistique des cooccurrences, tandis que dans une signature sémantique un élément linguistique est représenté comme une distribution de probabilité sur toutes les entités du réseau lexical utilisé (dans notre cas, JeuxDeMots) où les poids sont estimés sur la base des propriétés structurelles de ce réseau.

Après avoir présenté dans la section 2 les travaux liés à la création de représentations vectorielles pour les mots, nous décrivons dans la section 3 le réseau lexical JeuxDeMots et la méthode utilisée pour la génération des signatures sémantiques. Par la suite, dans la section 4, nous présentons les résultats et l'efficacité de l'utilisation de nos signatures par rapport à d'autres modèles vectoriels de représentation sémantique, avant de terminer, à la section 5, par une conclusion et quelques perspectives.

2 Travaux antérieurs

Mesurer la similarité sémantique entre les mots est un domaine qui a reçu beaucoup d'attention depuis plusieurs années. Des jeux de données ont été construits pour l'évaluation de la similarité, que ce soit pour l'anglais (Rubenstein & Goodenough, 1965) ou pour le français et l'allemand (Joubarne & Inkpen, 2011). Nous pouvons distinguer deux grandes catégories d'approches pour construire des représentations : (1) à base de modèles distributionnels (Baroni *et al.*, 2014; Turney & Pantel, 2010) ; (2) à base de ressources lexicales (Camacho-Collados *et al.*, 2016; Wu & Giles, 2015; Zesch

et al., 2008). Les modèles distributionnels sont le paradigme prédominant pour la modélisation des mots. Ce paradigme repose sur l'hypothèse suivante : « les mots dont les distributions sont similaires sont sémantiquement proches » (Harris, 1954). Les techniques les plus connues reposent sur l'analyse statistique des données textuelles (ADT) en prenant en considération les cooccurrences pour la création des représentations vectorielles de mots. Les modèles classiques considèrent le contexte comme un sac de mots (Deerwester *et al.*, 1990; Salton *et al.*, 1975) tandis que les modèles les plus sophistiqués tiennent compte, par exemple, des dépendances syntaxiques (Lin, 1998). Les poids dans les vecteurs à base de cooccurrences sont habituellement calculés à base de TF-IDF, TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (Jones, 1972) ou de la PMI, POINTWISE MUTUAL INFORMATION (Evert, 2005). Une nouvelle catégorie de modèles distributionnels a vu le jour récemment : des réseaux de neurones sont ici utilisés pour apprendre directement un contexte d'un mot donné, ou un mot à partir de son contexte. Les informations obtenues sont alors modélisées comme un vecteur continu. Le modèle le plus utilisé est celui de Mikolov et ses collaborateurs (Mikolov *et al.*, 2013), appelé souvent Word2Vec. Il y a deux types pour ce modèle : le premier repose sur une architecture fondée sur les sacs-de-mots continus (*continuous bag of words* ou CBOW), le deuxième repose sur une architecture fondé sur les SKIP-GRAM. Le type CBOW cherche à prédire un mot selon un contexte alors que le type SKIP-GRAM cherche à prédire un contexte sachant un mot. Un modèle compétitif au Word2Vec a été proposé par Pennington *et al.* (2014) et développé à Stanford, appelé GLOVE. Ce dernier n'utilise pas une architecture neuronale, il repose sur une utilisation de cooccurrences entre les termes (le nombre de fois qu'un terme apparaît en concomitance avec un autre). En croisant les probabilités de cooccurrences, il se veut capable de reproduire le même fonctionnement de Word2Vec. Ces modèles sont souvent appelés *word embeddings* (ou *plongements lexicaux*).

Les approches à base de ressources lexicales se décomposent elles-mêmes en deux catégories. Celles de la première catégorie tiennent compte de l'ensemble des sens possibles d'un mot donné, tandis que celles de la deuxième catégorie supposent que la similarité entre deux mots peut être calculée en fonction de la similarité de leurs sens les plus proches, ce qui permet d'appliquer directement toute mesure au niveau des sens pour comparer les couples de mots (Camacho-Collados *et al.*, 2015). Les approches de la première catégorie sont nombreuses, parmi elles, les plus grandes ressources collaboratives telles que Wikipédia (Wu & Giles, 2015) et le Wiktionnaire (Zesch *et al.*, 2008) ont été exploitées. C'est dans cette première catégorie d'approches que nous nous situons également.

3 Méthodologie

Cette section présente le réseau lexical JeuxDeMots, la méthode utilisée pour la génération de signatures sémantiques ainsi que les techniques pour mesurer la similarité entre les éléments linguistiques couverts par JeuxDeMots.

3.1 Réseau lexical JeuxDeMots

JeuxDeMots est un réseau lexical contributif où les acteurs clés sont de *simples internautes* qui jouent à travers une interface présentée sous forme d'un jeu en ligne [1]. La base lexicale de ce réseau est en constante évolution, sa structure s'appuie sur les notions de nœuds et de relations entre nœuds.

1. JeuxDeMots est accessible à l'adresse `http://jeuxdemots.org`

Chaque nœud représente une unité lexicale décrivant un terme (mot ou expression polylexicale, appelée aussi *MultiWord Expression* (MWE)). Les relations entre les nœuds sont typées et pondérées. Certaines de ces relations correspondent à des fonctions lexicales portant sur le vocabulaire lui-même (comme la relation d'*idée associée* et de *synonymie*) ou sur des relations sémantiques hiérarchisées (comme la relation d'*hyperonymie* évoquant des termes génériques et d'*hyponymie* évoquant des termes spécifiques). Il existe un autre type de relation décrit dans JeuxDeMots portant sur la prédiction de ce que peut faire un sujet ou ce qui peut être fait avec un objet. Nous supposons que l'intégration de ce dernier type de relation à un système de désambiguïsation sémantique peut être très intéressante.

La validation de la qualité des données collectées pour la construction de la base lexicale est fournie par les joueurs. Plus précisément, des relations proposées d'une manière anonyme par un joueur sont validées par d'autres joueurs, tout autant anonymement. Les relations entre les unités lexicales sont pondérées. La pondération s'effectue de la façon suivante : plus une instance d'une relation est proposée, plus son poids est important, ceci tout en respectant certaines règles du jeu qui n'acceptent pas les relations tabou. Pour la création de signatures sémantiques, nous utilisons la base lexicale datant de Janvier 2017 [2]. Cette base contient 67 603 805 instances de relations, 1 348 507 termes ayant au moins une relation sortante ($terme_A \rightarrow terme_B$) et 991 975 termes ayant au moins une relation entrante ($terme_A \leftarrow terme_B$).

3.2 Création de signatures sémantiques

Nous construisons différentes signatures pour chaque entrée lexicale dans JeuxDeMots. Une signature peut dépendre d'une seule relation r comme elle peut dépendre d'une combinaison de relations appartenant à l'ensemble R. Les dimensions dans une signature sont les nœuds liés par relations sortantes à l'entrée lexicale. Dans le cas où $|R| = 1$, le poids d'une dimension indique l'importance de cette dimension par rapport à la seule relation r pour une signature S. Si $|R| \geqslant 2$ alors le poids d'une dimension est la somme des poids de cette dimension sur l'ensemble des relations appartenant à R. Dans la ressource, les pondérations sont dans $\mathbb{R}$, c'est-à-dire qu'elles peuvent être positives comme elles peuvent être négatives. Nous prenons en considération seulement les pondérations positives (restriction à $\mathbb{R}^+$). La taille d'une signature n'est pas fixe et peut varier selon les liens sortants (instances des relations) de l'entrée lexicale vers les autres nœuds du réseau. Une signature peut être vu comme un vecteur où les dimensions inexistantes sont des dimensions ayant une valeur nulle.

Nous nous intéressons aux relations listées ci-dessous. Chacune d'elle est décrite, d'une part, avec un exemple et, d'autre part, avec son nombre d'instances dans le réseau lexical :

- **Synonyme** : termes décrivant un sens identique ou proche – relation lexicale (p.ex. *outil* $\rightarrow$ *instrument*) – 746 128 instances
- **Acception** : quels sont les termes évoquant les sens possibles de la cible ? – relation associative (p.ex. *outil* $\rightarrow$ *dispositif*) – 93 899 instances
- **Domaine** : domaines auxquels peut appartenir la cible – relation sémantique (p.ex. *outil* $\rightarrow$ *mécanique*) – 670 055 instances
- **Agent** : que peut faire ce sujet ? – relation prédicative (p.ex. *outil* $\leftarrow$ *fonctionner*) – 1 055 271 instances
- **Patient** : que peut-on faire avec cet objet ? – relation prédicative (p.ex. *outil* $\leftarrow$ *fabriquer*) – 65 397 instances

2. `http://www.jeuxdemots.org/JDM-LEXICALNET-FR/?C=M;O=D`

— **Hyperonyme (Générique)** : termes associés aux génériques de la cible – relation sémantique (p.ex. *outil* → *matériel*) – 3 155 763 instances
— **Hyponyme (Spécifique)** : termes associés aux spécifiques de la cible – relation sémantique (p.ex. *outil* → *marteau*) – 712 125 instances
— **Idée associée** : association libre – relation associative (p.ex. *outil* → *bricolage*) – 17 768 142 instances [3]

Nous construisons, d'une part, une signature pour chaque relation. D'autre part, cinq autres signatures sont construites en utilisant une combinaison de relations. La première utilise une combinaison de deux relations, à savoir, *Synonyme* et *Acception*. La deuxième combine deux relations, à savoir, *Hyperonyme* et *Hyponyme*. La troisième combine aussi deux relations, à savoir, *Agent* et *Patient*. La quatrième ne privilège aucune relation, toutes les relations ont un coefficient égal à 1, tandis que la cinquième, donne une importance supérieure à certaines relations :

• **Signature par combinaison de relations sans coefficient** : nous utilisons toutes les relations listées ci-dessus dont le poids de chaque dimension pour chaque relation est représenté avec un même coefficient.

• **Signature par combinaison de relations avec coefficient** : toutes les relations listées ci-dessus sont utilisées dont le poids de chaque dimension pour chaque relation est multiplié par un coefficient différent : {"*Domaine*" : 6, "*Acception*" : 5, "*Hyperonyme*" : 4, "*Hyponyme*" : 4, "*Synonyme*" : 3, "*Agent*" : 2, "*Patient*" : 2, "*Idée associée*" : 1}.

Les signatures sont par la suite normalisées. Plusieurs normes ont été présentées dans la littérature telles que la norme euclidienne, la norme 1 qui présente la somme des valeurs absolues des dimensions ou la norme infinie. La norme 1 est la norme adéquate au sens où nous voulons garder l'aspect de distribution de probabilité. Pour la normalisation de nos signatures, nous avons choisi d'utiliser la norme infinie. Elle se définit par la fonction : $\|\vec{X}\|_\infty = max(|x_1|, |x_2|, \ldots, |x_n|)$. Elle a l'avantage de comparer proportionnellement tous les poids des dimensions à la dimension possédant le poids maximal. Cette dimension aura une valeur égale à 1 et toutes les autres dimensions auront une valeur appartenant à l'intervalle $]0, 1]$. Un seuil est fixé à 0.01 pour la validation des dimensions des signatures. Toutes les dimensions ayant une valeur inférieure au seuil sont ignorées.

Si nous supposons que la fonction $Sim(A, B, R)$ retourne la valeur normalisée du poids de la dimension B dans la signature S_A du terme A construite à partir de l'ensemble des relations appartenant à R et si nous prenons la signature de l'entrée lexicale *intérêt* avec comme relations seulement la relation *Synonyme*, la fonction Sim(*intérêt, prix, Synonyme*) retourne une valeur de 0.98 et la fonction Sim(*intérêt, avantage, Synonyme*) retourne une valeur de 0.93.

3.3 Similarité entre les signatures sémantiques

Une fois que nous avons obtenu une signature sémantique pour chaque entrée lexicale se trouvant dans JeuxDeMots, nous pouvons calculer la similarité entre deux unités lexicales en comparant leurs signatures sémantiques correspondantes. Nous adoptons deux techniques pour cette comparaison : (1) Cosinus ; (2) Weighted Overlap (Pilehvar *et al.*, 2013). Il existe une autre classe de mesures statistiques qui s'appuient plutôt sur le classement absolu ou relatif des entités dans les vecteurs. Les exemples type pour ces mesures sont la corrélation de Spearman (ρ) et de Pearson (r), qui calculent

3. Il s'agit de la relation contenant le plus grand nombre d'instances puisqu'elle englobe tous les termes faisant penser à une entrée lexicale donnée.

la dépendance statistique du classement de deux variables. Pearson mesure la corrélation linéaire de deux variables en fonction des différences dans leurs valeurs, tandis que Spearman considère le classement relatif des valeurs des deux variables. Par la suite, pour la comparaison de nos résultats avec ceux d'une liste de référence (gold standard), nous utilisons la corrélation de Pearson comme mesure principale pour l'évaluation sur la tâche de mesures de similarité sémantique (cf. Section 4.2).

• **Cosinus** : cette mesure permet de calculer la similarité entre deux signatures S_1 et S_2 en traitant chacune d'elle comme un vecteur puis en calculant le rapport entre le produit scalaire et la norme des deux vecteurs. Cette fonction se présente comme suit :

$$Sim_{Cos}(S_1, S_2) = \frac{S_1 \,.\, S_2}{\|S_1\| \, \|S_2\|} \tag{1}$$

• **Weighted Overlap (WO)** : cette mesure calcule la similarité entre deux listes ordonnées en comparant le classement des dimensions. Nous supposons que les éléments de chaque liste sont classés selon leur poids d'importance, du plus fort vers le plus faible. Soit D l'ensemble des dimensions non nulles qui apparaissent à la fois dans les deux signatures S_1 et S_2. Soit $r_d(S)$ la fonction qui renvoie le rang de la dimension d dans la signature S. La fonction WO calcule la similarité entre les deux signatures comme suit :

$$Sim_{WO}\ (S_1, S_2) = \frac{\sum_{d \in D} \left(r_d(S_1) + r_d(S_2)\right)^{-1}}{\sum_{i=1}^{|D|} (2i)^{-1}} \tag{2}$$

Le dénominateur est un facteur de normalisation qui garantit une valeur maximale de 1. La fonction retourne une valeur minimale de 0, cette valeur se produit lorsqu'il n'y a pas de chevauchement entre les deux signatures, c'est-à-dire $|D| = 0$. Elle retourne la valeur de 1 lorsqu'il y a une parfaite correspondance au niveau du classement des dimensions.

3.4 Fonctions d'activation

Soient A et B deux termes décrits dans JeuxDeMots, soit R un ensemble de relations contenant au moins une relation r. La valeur $A[B]$ est le poids de B dans la signature normée S_A du terme A. De même pour $B[A]$ qui présente le poids de A dans la signature normée S_B du terme B. Une fonction d'activation est décrite comme suit :

$$Act_i : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{N}^+ \rightarrow \mathbb{R}^+$$

Nous utilisons quatre fonctions d'activation, que nous appelons aussi configurations, pour comparer nos signatures sémantiques. Les deux fonctions Act_3 et Act_4 numérotées en (5) et (6), respectivement, prennent une forme similaire à la fonction d'activation proposée par Lafourcade (2011, 21-22). Chaque fonction est utilisée par la suite dans la section 4 pour l'évaluation. Les fonctions Act_1 et Act_2 numérotées en (3) et (4), respectivement, retournent une valeur de 1 si $A \in S_B$ ou $B \in S_A$.

$$Act_1(A, B, R)\ = \begin{cases} 1\ si\ A \in S_B\ ou\ B \in S_A \\ Sim_{Cos}(S_A, S_B)\ sinon \end{cases} \tag{3}$$

$$Act_2(A, B, R) = \begin{cases} 1 \; si \; A \in S_B \; ou \; B \in S_A \\ Sim_{WO}(S_A, S_B) \; sinon \end{cases} \quad (4)$$

La différence entre Act_1, Act_2 et Act_3, Act_4 décrites ci-dessous est dans la manière de calculer la similarité au cas où $A \in S_B$ ou $B \in S_A$. Si nous reprenons l'exemple des mots *intérêt* et *prix* avec *Synonyme* comme relation, nous avons $S_{intérêt}[prix] = 0.98$ et $S_{prix}[intérêt] = 0.27$. Les fonctions Act_1 et Act_2 retournent une valeur égale à 1 puisque *prix* $\in$ Synonymes$_{intérêt}$ comme *intérêt* $\in$ Synonymes$_{prix}$ tandis que la fonction Act_3 retourne $max(0.98, Sim_{Cos}(S_{intérêt}, S_{prix}))$ et la fonction Act_4 retourne $max(0.98, Sim_{WO}(S_{intérêt}, S_{prix}))$.

$$Act_3(A, B, R) = max(A[B],\ B[A], Sim_{Cos}(S_A, S_B)) \quad (5)$$

$$Act_4(A, B, R) = max(A[B],\ B[A], Sim_{WO}(S_A, S_B)) \quad (6)$$

4 Évaluation

Après avoir obtenu les signatures sémantiques des unités lexicales, restait à en évaluer la qualité. Nous présentons tout d'abord en section 4.1 les modèles auxquels nous nous comparons. Par la suite, les sections 4.2 et 4.3 décrivent les deux tâches qui nous ont servi à l'évaluation : mesures de similarité sémantique et substitution lexicale, respectivement.

4.1 Systèmes existants

Nous comparons nos représentations vectorielles avec deux modèles provenant des systèmes de l'état de l'art. Le premier repose sur des représentations de sens tandis que le deuxième repose sur des représentations de mots dans un espace vectoriel continu (*word embeddings*).

– **NASARI** : il s'agit d'un modèle sémantique distributionnel représentant les sens avec des vecteurs pondérés (Camacho-Collados *et al.*, 2015). Le calcul des pondérations repose sur des mesures statistiques utilisées principalement pour l'extraction de termes. NASARI intègre deux ressources : (1) le réseau sémantique multilingue BabelNet (Navigli & Ponzetto, 2012) ; (2) l'encyclopédie Wikipédia. Il permet de fournir un vecteur pour chaque sens et chaque article. Ce modèle propose des représentations sémantiques seulement pour les noms communs et les entités nommées. Nous utilisons l'algorithme proposé par Camacho-Collados *et al.* (2015, 570) pour mesurer la similarité sémantique entre deux mots. Cet algorithme propose une mesure égale à 1 si les deux mots sont synonymes [4]. Il intègre WO comme mesure de similarité entre sens. Nous utilisons une deuxième instance de cet algorithme avec la mesure Cosinus.

– **Word Embeddings** : il s'agit de modèles sémantiques distributionnels permettant de projeter les mots dans un espace dans lequel les relations sémantiques entre ces mots peuvent être observées ou mesurées. La technique des *word embeddings* consiste à projeter les mots d'une langue (contenu

4. Nous utilisons la version 3.7 de BabelNet comme base de synonymes. `http://babelnet.org/download`

dans une fenêtre graphique définie) dans un espace de représentation vectorielle. Chaque mot est représenté par un vecteur , à n dimensions, qui correspond à une projection du mot dans un espace où les distances modélisent les relations inter-mots. Cette projection permet de tirer profit des mots selon leurs sens dans une région de l'espace sémantique proche. Par exemple, *Paris* et *Londres* peuvent partager l'idée de *capitale*. Nous utilisons un jeu de vecteurs basé sur une variante de l'algorithme GLOVE (Pennington *et al.*, 2014) proposée par l'équipe Alpage pour le français et que nous appellerons par la suite DEPGLOVE [5]. Nous utilisons Cosinus comme mesure de similarité.

4.2 Similarité sémantique

Nous évaluons la qualité de nos signatures sémantiques par rapport à un jugement humain. Pour cela, nous calculons la corrélation linéaire entre les scores retournés par nos fonctions d'activation et les annotations humaines. L'idée est de voir si les scores obtenus sont fortement corrélés avec ceux donnés par les humains. Nous avons opté pour l'utilisation de la liste de référence RG-65 pour le français (Joubarne & Inkpen, 2011) qui présente une traduction avec un autre jugement humain pour le jeu de données RG-65 créé pour l'anglais (Rubenstein & Goodenough, 1965). Le but de la création de ce dernier consistait à étudier la similarité sémantique et contextuelle pour un ensemble de 65 paires de noms communs évaluées sur une échelle de 0 (non liés) à 4 (complètement liés). Le RG-65 que nous utilisons a fait appel à 18 évaluateurs humains qui ont le français comme langue maternelle. Ce jeu de données présente une traduction en langue française du RG-65 pour l'anglais par utilisation d'une combinaison du dictionnaire Larousse français-anglais, Le Grand dictionnaire terminologique (maintenu par l'Office québécois de la langue française), un couple de locuteurs natifs et un traducteur humain. Nous avons pris la liste des paires telle qu'elle est fournie par Joubarne & Inkpen (2011) [6]. Parmi les 65 paires traduites, la traduction directe pour chaque mot de deux paires a retourné le même mot. Il s'agit de la paire *(cock, rooster)* traduite en *(coq, coq)* et de la paire *(cemetery, graveyard)* traduite en *(cimetière, cimetière)*. Ces deux paires ne sont pas utilisées pour l'évaluation. Le tableau 1 décrit l'ensemble de noms communs se trouvant dans RG-65. Nous présentons dans le tableau 2 les résultats de corrélation obtenus par utilisation de toutes les signatures construites ; quant à la liste des 63 paires, avec les scores du jugement humain et les scores retournés automatiquement par utilisation de nos signatures, elle figure dans l'annexe A.

asile	*bois*	*coussin*	*fournaise*	*grimace*	*moine*	*outil*	*signature*
asylum	*cimetière*	*dîner*	*frère*	*grue*	*monticule*	*périple*	*sorcier*
auto	*colline*	*esclave*	*fruit*	*instrument*	*nourriture*	*refuge*	*sourire*
autographe	*coq*	*ficelle*	*garçon*	*joyau*	*oiseau*	*rivage*	*trip*
automobile	*corde*	*forêt*	*gars*	*magicien*	*oracle*	*sage*	*verre*
bijou	*côte*	*four*	*goblet*	*midi*	*oreiller*	*serf*	*voyage*

TABLE 1 – Les 48 mots du vocabulaire correspondant au jeu de données RG-65

Le vocabulaire du jeu de données n'a pas une couverture parfaite sur l'ensemble des signatures. Par exemple, les mots *asylum* et *goblet* ne sont associés à aucun nœud dans le réseau lexical et n'ont aucune signature. La raison est qu'ils représentent une mauvaise traduction en français ou que la traduction n'a jamais eu lieu. Le nom *asylum* peut se traduire par *refuge* ou *asile* et le nom *goblet* s'écrit *gobelet* en français. Nous appelons *asylum* et *goblet* des OOV (*Out-Of-Vocabulary*). Il y a seulement 46 mots

5. http ://alpage.inria.fr/depglove/process.pl
6. http ://www.site.uottawa.ca/ mjoub063/wordsims.htm

sur 48 ayant des signatures pour les types suivants : {*synonyme, combinaison de synonyme avec acception (r_synonyme_acception), idée associée, combinaison de toutes les relations sans coefficient (r_traits_égaux), combinaison de toutes les relations avec coefficient (r_traits_avec_coeff)*}.

Types de signature	Couverture (%)	$\mathbf{Act_1}$	$\mathbf{Act_2}$	$\mathbf{Act_3}$	$\mathbf{Act_4}$
r_synonyme	**95.83**	**0.88**	0.85	**0.89**	0.85
r_acception	77.08	0.80	0.80	0.79	0.80
r_synonyme_acception	**95.83**	**0.88**	**0.88**	0.87	**0.87**
r_domaine	81.25	0.43	0.39	0.40	0.32
r_agent	58.33	0.23	0.35	0.23	0.35
r_patient	60.42	0.51	0.47	0.51	0.47
r_agent_patient	72.92	0.25	0.33	0.25	0.33
r_hyperonyme	85.42	0.46	0.51	0.43	0.48
r_hyponyme	87.5	0.45	0.45	0.44	0.44
r_hyperonyme_hyponyme	89.58	0.44	0.48	0.43	0.47
r_idée_associée	**95.83**	**0.88**	**0.88**	0.82	0.81
r_traits_avec_coeff	**95.83**	0.86	0.86	0.81	0.81
r_traits_égaux	**95.83**	0.87	0.87	0.85	0.85

TABLE 2 – Corrélations de Pearson obtenues selon différentes signatures avec différentes configurations

Nous constatons que les signatures à base de synonymie permettent d'obtenir les meilleures corrélations (il est à noter que cette relation propose des éléments lexicaux qui vont au-delà de la définition stricte de synonymie). Nous obtenons une corrélation de 0.89 par utilisation de la troisième fonction d'activation et 0.88 par utilisation de la première fonction. La combinaison de la relation de *Synonyme* avec *Acception* ne retourne pas une meilleure corrélation (Act_1 et Act_3) par rapport à la simple utilisation de la relation de *Synonyme*. La raison est que nous obtenons par cette combinaison des signatures avec un plus grand nombre de dimensions. Le chevauchement entre les signatures n'est pas plus grand que l'utilisation d'une seule relation. Il en est de même pour la combinaison des relations *Hyperonyme* et *Hyponyme*. Cependant, ces deux dernières relations ne permettent pas d'obtenir une meilleure couverture. JeuxDeMots (à ce jour) ne propose aucun terme générique ou spécifique pour les mots suivants : {*asylum, autographe, goblet, périple, trip*}. Pour la relation *Idée associée*, la corrélation obtenue reste relativement supérieure par rapport aux autres relations ou combinaison de relations hors *r_synonyme* et *r_synonyme_acception*.

Nous utilisons les signatures construites avec les types suivants pour la comparaison avec les autres modèles décrits dans la section 4.1, à savoir : *r_traits_avec_coeff, r_traits_égaux, r_idée_associée et r_synonyme*. Le tableau 3 présente les résultats de corrélation obtenus pour cette comparaison. Il est à noter que même les modèles NASARI et DEPGLOVE ne permettent pas d'avoir une couverture parfaite. Par exemple, NASARI ne fournit aucune représentation de sens pour le nom *trip* et DEPGLOVE ne fournit aucun vecteur dans son espace vectoriel continu pour les noms *asylum* et *goblet*. Cela nous amène à mettre à l'écart trois paires contenant au moins l'un de ces noms [7] pour permettre une comparaison sur un même ensemble de paires couvertes par tous les modèles. Dans le tableau 3, la comparaison s'effectue par utilisation de la première fonction d'activation.

Les résultats obtenus montrent clairement que nos signatures permettent d'avoir une corrélation

7. Rappelons que deux paires ont été déjà ignorées à cause de la traduction qui a retourné le même mot pour les deux éléments de la paire, ce qui nous ramène à garder 60 paires au final.

Système	Corrélation de Pearson (r)	Corrélation de Spearman (ρ)
r_traits_avec_coeff	0.86	**0.85**
r_traits_égaux	0.87	**0.85**
r_idée_associée	**0.88**	0.83
r_synonyme	**0.88**	0.75
DEPGLOVE	0.48	0.50
NASARI$_{COS}$	0.80	0.77
NASARI$_{WO}$	0.82	0.78

TABLE 3 – Corrélations de Pearson et Spearman obtenues selon différentes signatures avec utilisation de la première configuration, comparaison avec les résultats obtenus par NASARI et DEPGLOVE sur un ensemble de 60 paires couvertes par tous les systèmes

largement supérieure aux systèmes à base de corpus. Une corrélation de Pearson de 0.88 est obtenue sur les 60 paires pour la signature typée avec la relation *Idée associée* contre 0.82 pour NASARI à base de la mesure *Weighted Overlap* ou 0.48 pour DEPGLOVE. Pour la corrélation de Spearman (ρ), nous avons obtenu une valeur de 0.85 pour *r_traits_avec_coeff* et *r_traits_égaux*, 0.83 pour *r_idée_associée* et 0.75 pour *r_synonyme* contre une valeur de 0.78 pour NASARI$_{\text{WO}}$, 0.77 pour NASARI$_{\text{COS}}$ et 0.50 pour DEPGLOVE.

Joubarne & Inkpen (2011) ont utilisé deux mesures de similarité sémantique à base de corpus : (1) POINTWISE MUTUAL INFORMATION (PMI) ; (2) SECOND ORDER CO-OCCURRENCE POINTWISE MUTUAL INFORMATION (SOC-PMI). Le principe de la PMI est d'estimer si l'apparition simultanée de deux mots A et B est supérieure à la probabilité d'apparition *à priori* des deux mots indépendamment ; quant à la SOC-PMI, il s'agit d'un même principe en tenant compte des mots communs apparaissant dans le voisinage de A et B selon une fenêtre contextuelle définie à la base. Les corrélations de Pearson obtenues entre ces mesures et les 18 évaluateurs humains pour l'ensemble des 63 paires sont de 0.29 pour la PMI et de 0.17 pour la SOC-PMI.

4.3 Substitution lexicale

La substitution lexicale est une tâche qui, ces dernières années, a reçu un intérêt majeur au sein de la communauté du traitement automatique des langues. D'abord, une première campagne d'évaluation, SemEval 2007, a vu le jour pour l'anglais (McCarthy & Navigli, 2009) ; ensuite une adaptation de cette dernière a été présentée pour le français (Fabre *et al.*, 2014) dans l'atelier de la Sémantique Distributionnelle [8] (SemDis), basée sur des données issues du corpus français FRWAC (Baroni *et al.*, 2009). Le principe est de remplacer un mot-cible par un substitut potentiel tout en gardant le même sens du mot-cible par rapport à un contexte donné.

La substitution lexicale a un double intérêt pour l'évaluation de la similarité sémantique : (1) elle représente une évaluation extrinsèque pour laquelle la similarité sémantique à un rôle prépondérant pour que des différences la concernant puissent être observées vis-à-vis de la tâche de substitution ; (2) le niveau contextuel est pris en compte. Cette tâche se décompose elle-même en deux sous-tâches : (a) génération de candidats substituts pour le mot-cible à remplacer ; (b) choix de l'un des candidats en fonction du contexte. Le jeu d'évaluation fourni dans SemDis comporte 30 unités lexicales (10 noms,

8. https ://www.irit.fr/semdis2014/fr/task1.html

10 verbes et 10 adjectifs). Pour chaque mot-cible, 10 phrases différentes ont été proposées (300 phrases au total). Pour chaque phrase, il est possible de fournir jusqu'à 10 substituts au maximum classés par ordre décroissant de préférence. Les données ont été fournies par la suite avec des annotations manuelles. Le tableau 4 décrit les mots-cibles à substituer.

Noms	**Verbes**	**Adjectifs**
affection, capacité, couverture, débit, direction, don, espace, intérêt, montée, vaisseau	*arrêter, commander, entraîner, éplucher, essuyer, faucher, fonder, interpréter, maintenir, taper*	*aisé, compris, grossier, hermétique, incorrect, mince, modeste, obscur, riche, vaseux*

TABLE 4 – Les 30 mots-cibles pour la tâche de substitution lexicale

Pour la première sous-tâche qui consiste à générer des candidats substituts, nous prenons les signatures construites à base de la relation de *Synonyme*. Pour une entrée lexicale donnée, les dimensions de sa signature représentent des substituts potentiels. Nous avons fait le choix de présélectionner les candidats en tenant compte seulement des synonymes ayant un poids d'importance supérieur ou égal à la valeur de 0.8, cela afin de tenir compte seulement des termes représentant des synonymes stricts.

Pour la deuxième sous-tâche, nous ne nous intéressons pas à développer un modèle sophistiqué de substitution lexicale mais plutôt à comparer l'utilisation de nos représentations sémantiques avec un modèle utilisant un algorithme comme celui décrit par Ferret (2014). Cet algorithme consiste à mesurer la similarité entre chaque candidat substitut et l'ensemble de mots pleins de la phrase contenant le mot-cible à remplacer, hors ce dernier. Par la suite, nous faisons appel à cet algorithme par *Sub_Lex*. Afin d'avoir l'ensemble de mots pleins, nous avons réalisé une analyse morpho-syntaxique avec l'outil Talismane [9] (Urieli, 2013) sur l'ensemble des phrases du corpus SemDis.

4.3.1 Mesures d'évaluation

Il s'agit de mesures utilisées dans SemEval 2007 pour la tâche de substitution lexicale, à savoir la mesure *best* et la mesure *oot (out of ten)* [10].

- **best** : le système est évalué par rapport à la première substitution proposée. Le meilleur score renvoie le substitut choisi majoritairement par les annotateurs.
- **oot** *(out of ten)* : le système est évalué par rapport à tous les substituts proposés (dans la limite de 10). Le meilleur score obtenable correspond au nombre maximum de réponses couvertes par les annotateurs.

4.3.2 Résultats d'expérimentation

Nous avons testé les quatre configurations sur trois signatures à base des types suivants : {*idée associée, combinaison de toutes les relations sans coefficient, combinaison de toutes les relations avec coefficient*}. Le tableau 5 présente les résultats obtenus.

Par rapport aux systèmes décrits dans la section 4.1, nous avons comparé l'utilisation de nos représentations avec DEPGLOVE seulement. Nous ne pouvons pas appliquer l'algorithme proposé par

9. http ://redac.univ-tlse2.fr/applications/talismane.html

10. Pour comprendre mieux le fonctionnement de ces mesures, nous renvoyons le lecteur à consulter le travail de Fabre *et al.* (2014, 201).

Système	best				oot			
	Nom	*Adj.*	*Verbe*	*Total*	*Nom*	*Adj.*	*Verbe*	*Total*
JDM_TraitsÉgaux_FctAct3	.078	.100	.059	**.079**	.261	.323	.339	.308
JDM_TraitsÉgaux_FctAct2	.068	**.111**	.053	.077	.269	.322	.332	.308
JDM_TraitsÉgaux_FctAct4	.071	.098	.063	.077	.269	.325	.331	.308
JDM_TraitsIdéeAssociée_FctAct2	**.081**	.092	.054	.076	.277	**.346**	.317	.313
JDM_TraitsÉgaux_FctAct1	.075	.099	.053	.076	.258	.322	.340	.307
JDM_TraitsAvecCoeff_FctAct1	.077	.094	.056	.076	.259	.326	**.341**	.309
JDM_TraitsAvecCoeff_FctAct3	.078	.075	**.070**	.074	.260	.322	.340	.307
JDM_TraitsIdéeAssociée_FctAct3	.060	.095	.063	.073	.270	.323	.318	.304
JDM_TraitsAvecCoeff_FctAct2	.066	.096	.052	.071	.264	.324	.334	.307
JDM_TraitsIdéeAssociée_FctAct4	.067	.093	.046	.069	**.283**	.344	.317	**.315**
JDM_TraitsIdéeAssociée_FctAct1	.060	.097	.047	.068	.268	.326	.318	.304
JDM_TraitsAvecCoeff_FctAct4	.058	.066	.058	.061	.261	.327	.334	.307
baseline_jdmsyn	.029	.051	.006	.029	.247	.303	.258	.269
DEPGLOVE	.017	.033	.053	.034	.242	.280	.331	.284
Proxteam_JDM_Syn	.110	.106	.075	.097	.398	.429	.379	.402
CEA_list-word_cos_sent ***(Sub_Lex)***	.075	.074	.076	.075	.195	.245	.268	.236
Proxteam_AxeParaProx_JDM_Syn	.055	.054	.087	.065	.311	.396	.363	.357
Alpage_WoDiS	.054	.072	.061	.063	.191	.211	.213	.205
Proxteam_LM	.052	.040	.061	.051	.233	.166	.237	.212
baseline_Campagne	.044	.040	.052	.045	.294	.336	.344	.325
CEA_list-fredist_cos_sent ***(Sub_Lex)***	.032	.028	.060	.040	.181	.225	.303	.236
CEA_list-isc_cos_w2	.030	.041	.041	.037	.243	.281	.329	.284
CEA_list-isc_cos_sent ***(Sub_Lex)***	.025	.034	.040	.033	.233	.287	.340	.287
CEA_list-isc_l2_sent ***(Sub_Lex)***	.004	.012	.015	.010	.163	.230	.300	.231

TABLE 5 – Résultats pour la tâche de substitution lexicale selon différentes signatures avec différentes configurations, comparaison avec les résultats obtenus en utilisant DEPGLOVE et ceux des systèmes ayant participé à l'atelier SemDis

Ferret (2014) en utilisant NASARI car ce dernier propose des représentations vectorielles de sens seulement pour les noms. Son utilisation, dans ce cas, permet d'évaluer seulement les noms et réduit le contexte en comparant un candidat substitut seulement avec les noms du contexte.

Les résultats du tableau 5 sont classés par catégorie grammaticale et par ordre décroissant du score *best* sur le l'ensemble des mots à substituer (*Total*) du corpus SemDis. Le tableau regroupe différents systèmes sur trois parties : (1) *Sub_Lex* à base de nos signatures ; (2) *Sub_Lex* à base de DEPGLOVE ; (3) systèmes présentés dans SemDis utilisant le même algorithme que le notre (notés avec *Sub_Lex*) ainsi que les autres systèmes de la campagne utilisant un algorithme différent. Nous utilisons une baseline, *baseline_jdmsyn*, consistant à renvoyer les 10 premiers synonymes d'un mot-cible par ordre d'importance depuis les signatures à base de la relation de *Synonyme*. Durant l'atelier SemDis, une baseline a été proposée, *baseline_Campagne* . Elle consiste d'abord à sélectionner dans le dictionnaire DicoSyn (Ploux & Victorri, 1998) l'ensemble des synonymes pour un mot-cible en ne prenant que les mots simples en compte, puis de prendre les dix premiers synonymes selon un ordre de fréquence décroissant dans le corpus FRWAC.

Il apparaît clairement dans le tableau 5 que l'utilisation des signatures à base de combinaison

des différentes relations avec un même coefficient [11] rend performant l'algorithme implémenté. D'autre part, cet algorithme surpasse la baseline par utilisation de nos trois signatures sur toutes les configurations. Pour les systèmes proposés par Ferret (2014), à savoir, les quatre systèmes décrits dans le tableau 5 notés avec CEA_list-.* et (*Sub_Lex*), l'utilisation de nos configurations et signatures sémantiques reste globalement meilleure que l'utilisation de ses représentations sémantiques à base du modèle neuronal SKIP-GRAM dont l'une des différences secondaires, hors le modèle, avec DEPGLOVE est dans le corpus utilisé pour l'entraînement des *word embeddings*.

Il existe un seul système parmi tous les systèmes décrits dans Fabre *et al.* (2014) surpassant les performances globales de ce que nous proposons. Il retourne un *best* de .097 contre notre meilleur système (.079) (cf. *Proxteam_JDM_Syn* représenté dans le tableau 5). La qualité est dans l'algorithme utilisé. Ce dernier repose sur des balades aléatoires dans des graphes construits à partir de corpus et différentes ressources lexicales. Il est à noter que les résultats obtenus pour cette tâche dépendent à la fois des ressources et des algorithmes utilisés.

5 Conclusion et perspectives

Dans cet article, nous avons décrit une approche à base du réseau lexical JeuxDeMots permettant de créer des signatures sémantiques pour des mots. Nous avons évalué ces signatures sur deux tâches différentes : mesures de similarité sémantique en utilisant le jeu de données RG-65 et la substitution lexicale en utilisant le corpus SemDis. Pour cette deuxième tâche, nous avons utilisé un algorithme classique consistant à mesurer la similarité sémantique entre chaque candidat substitut et l'ensemble des mots pleins du contexte contenant le mot-cible à remplacer, hors ce dernier. Notre approche repose sur l'utilisation de plusieurs relations définies dans la ressource JeuxDeMots. Nous avons utilisé quatre fonctions différentes pour mesurer la similarité sémantique et nous avons démontré que les résultats obtenus en utilisant notre approche surpassent les résultats obtenus en utilisant les systèmes de l'état de l'art comme GLOVE ou NASARI.

Comme perspectives de ce travail, nous envisageons d'utiliser nos signatures sémantiques pour la tâche de désambiguïsation sémantique en tenant compte de l'hypothèse qu'un sens peut être représenté par un ensemble de synonymes désambiguïsés. Nous supposerons que la similarité entre deux sens est celle de leurs synonymes les plus proches. D'un autre côté, en comparaison avec un algorithme exhaustif qui consiste à comparer chaque sens candidat d'un mot-cible avec chaque sens de chaque mot du contexte, il serait possible de réduire le contexte tout en gardant une cohérence au niveau de la désambiguïsation. Tout cela nous rendra capable par la suite de comparer, par exemple, un mot à un sens d'un autre mot. Finalement, nous envisageons d'ajouter une étape de désambiguïsation sémantique avant la substitution lexicale pour réduire le nombre de candidats substituts en ne gardant que ceux susceptibles d'avoir le même sens que celui du mot-cible par rapport à un contexte donné.

Remerciements

Nous tenons à remercier tout particulièrement Mathieu Lafourcade pour la mise à disposition du réseau lexical JeuxDeMots ainsi que tous les joueurs ayant participé de près ou de loin à sa construction.

11. Il s'agit des systèmes *JDM_TraitsÉgaux_FctActi* avec $\mathbf{i} \in \{1, 2, 3, 4\}$.

Références

BARONI M., BERNARDINI S., FERRARESI A. & ZANCHETTA E. (2009). The WaCky wide web : a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, **43**(3), 209–226.

BARONI M., DINU G. & KRUSZEWSKI G. (2014). Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, **1**, 238–247.

BIRAN O., BRODY S. & ELHADAD N. (2011). Putting It Simply : A Context-aware Approach to Lexical Simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : Short Papers*, volume 2 of *HLT '11*, p. 496–501, Stroudsburg, PA, USA.

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

CAMACHO-COLLADOS J., PILEHVAR M. T. & NAVIGLI R. (2015). NASARI : a Novel Approach to a Semantically-Aware Representation of Items. In R. MIHALCEA, J. Y. CHAI & A. SARKAR, Eds., *HLT-NAACL*, p. 567–577 : The Association for Computational Linguistics.

CAMACHO-COLLADOS J., PILEHVAR M. T. & NAVIGLI R. (2016). NASARI : Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artif. Intell.*, **240**, 36–64.

CORLEY C. & MIHALCEA R. (2005). Measuring the Semantic Similarity of Texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, p. 13–18, Stroudsburg, PA, USA : Association for Computational Linguistics.

DEERWESTER S. C., DUMAIS S. T., LANDAUER T. K., FURNAS G. W. & HARSHMAN R. A. (1990). Indexing by Latent Semantic Analysis. *JASIS*, **41**(6), 391–407.

EVERT S. (2005). The statistics of word cooccurrences : word pairs and collocations. *Ph.D. thesis, Universität Stuttgart*.

FABRE C., HATHOUT N., HO-DAC L.-M., MORLANE-HONDÈRE F., MULLER P., SAJOUS F., TANGUY L. & VAN DE CRUYS T. (2014). Présentation de l'atelier SemDis 2014 : sémantique distributionnelle pour la substitution lexicale et l'exploration de corpus spécialisés. In *Actes de l'atelier SemDis 2014, 21e Conférence sur le Traitement Automatique des Langues Naturelles*, p. 196–205, Marseille, France.

FERRET O. (2014). Utiliser un modèle neuronal générique pour la substitution lexicale. In *Actes de l'atelier SemDis 2014, 21e Conférence sur le Traitement Automatique des Langues Naturelles*, p. 218–227, Marseille, France.

HARRIS Z. (1954). Distributional structure. *Word*, **10**(23), 146–162.

JONES K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, **28**, 11–21.

JOUBARNE C. & INKPEN D. (2011). Comparison of Semantic Similarity for Different Languages Using the Google n-gram Corpus and Second-Order Co-occurrence Measures. In *Advances in Artificial Intelligence - 24th Canadian Conference on Artificial Intelligence, Canadian AI 2011, St. John's, Canada, May 25-27, 2011. Proceedings*, p. 216–221.

LAFOURCADE M. (2007). Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *SNLP'07 : 7th International Symposium on NLP*, Pattaya, Chonburi, Thailande.

LAFOURCADE M. (2011). *Lexique et analyse sémantique de textes - structures, acquisitions, calculs, et jeux de mots.* Mémoire d'habilitation à diriger les recherches, Université Montpellier 2, LIRMM.

LAVIE A. & DENKOWSKI M. J. (2009). The METEOR Metric for Automatic Evaluation of Machine Translation. *Machine Translation*, **23**(2-3), 105–115.

LIN D. (1998). Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 17th International Conference on Computational Linguistics*, volume 2 of *COLING '98*, p. 768–774, Stroudsburg, PA, USA : Association for Computational Linguistics.

MATUSCHEK M. & GUREVYCH I. (2013). Dijkstra-WSA : A Graph-Based Approach to Word Sense Alignment. *Transactions of the Association for Computational Linguistics*, **1**, 151–164.

MCCARTHY D. & NAVIGLI R. (2009). The English Lexical Substitution Task. *Language Resources and Evaluation*, **43**(2), 139–159.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. *CoRR*, **abs/1301.3781**.

NAVIGLI R. (2009). Word Sense Disambiguation : A Survey. *ACM Computing Surveys*, **41**(2), 1–69.

NAVIGLI R. & PONZETTO S. P. (2012). BabelNet : The Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network. *Artif. Intell.*, **193**, 217–250.

OTEGI A., ARREGI X., ANSA O. & AGIRRE E. (2015). Using knowledge-based relatedness for information retrieval. *Knowl. Inf. Syst.*, **44**(3), 689–718.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global Vectors for Word Representation. In *EMNLP*, volume 14, p. 1532–1543.

PILEHVAR M. T., JURGENS D. & NAVIGLI R. (2013). Align, Disambiguate and Walk : A Unified Approach for Measuring Semantic Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria*, volume 1, p. 1341–1351.

PLOUX S. & VICTORRI B. (1998). Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes. *Traitement Automatique des Langues*, **39**(1), 161–182.

RUBENSTEIN H. & GOODENOUGH J. B. (1965). Contextual Correlates of Synonymy. *Commun. ACM*, **8**(10), 627–633.

SALTON G., WONG A. & YANG C. S. (1975). A Vector Space Model for Automatic Indexing. *Commun. ACM*, **18**(11), 613–620.

TURNEY P. D. & PANTEL P. (2010). From Frequency to Meaning : Vector Space Models of Semantics. *J. Artif. Int. Res.*, **37**(1), 141–188.

URIELI A. (2013). *Robust French syntax analysis : reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. PhD thesis, Université de Toulouse II le Mirail.

VOORHEES E. M. (1994). Query Expansion Using Lexical-semantic Relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, p. 61–69, New York, NY, USA.

WU Z. & GILES C. L. (2015). Sense-aware Semantic Analysis : A Multi-prototype Word Representation Model Using Wikipedia. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, p. 2188–2194 : AAAI Press.

ZESCH T., MULLER C. & GUREVYCH I. (2008). Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 2 of *AAAI'08*, p. 861–866 : AAAI Press.

Annexe A

Ci-dessous, la liste des 63 paires de mots avec, d'une part, les scores des évaluateurs humains (Sc_H) et, d'autre part, les scores obtenus par application de la troisième fonction d'activation en utilisant les types de signatures sémantiques suivants : (1) *r_synonyme* (Sc_{r_syn}), (2) r_idée_associée ($Sc_{r\_idée\_associée}$), (3) r_traits_avec_coeff ($Sc_{r\_traits\_avec\_coeff}$) et (4) r_traits_égaux ($Sc_{r\_traits\_égaux}$). La corrélation de Pearson (r) est fournie en entête.

$\mathbf{Mot_A}$	$\mathbf{Mot_B}$	$\mathbf{Sc_H}$	$\mathbf{Sc_{r\_syn}(r = 0.89)}$	$\mathbf{Sc_{r\_idée\_associée}(r = 0.82)}$	$\mathbf{Sc_{r\_traits\_avec\_coeff}(r = 0.81)}$	$\mathbf{Sc_{r\_traits\_égaux}(r = 0.85)}$
autographe	rivage	0.0	0.0	0.0	0.0	0.0
automobile	sorcier	0.0	0.0	0.0022	0.0	9.10E-4
corde	sourire	0.0	0.0	0.0	0.0	0.0
grimace	instrument	0.0	0.0	0.0	0.0	0.0
midi	ficelle	0.0	0.0	0.0	0.0	0.0
refuge	fruit	0.0	0.03	0.0020	0.0098	0.0069
automobile	coussin	0.06	0.0	0.01	0.03	0.02
coq	périple	0.06	0.0	0.0026	4.38E-4	0.0010
monticule	four	0.06	0.0	0.0	0.0	0.0
oiseau	bois	0.06	0.0	0.02	0.0032	0.0077
verre	magicien	0.06	0.0	9.55E-4	0.0011	0.0044
cimetière	bois	0.11	0.0	0.0020	0.01	0.0058
fruit	fournaise	0.11	0.0	0.0	0.0	0.0
grimace	gars	0.11	0.0	0.03	7.84E-4	0.0048
coussin	bijou	0.17	0.0	0.0020	2.04E-4	9.21E-4
forêt	cimetière	0.17	0.0	0.0010	0.04	0.01
moine	esclave	0.17	0.0	0.02	0.0077	0.01
monticule	rivage	0.17	0.0	0.0	0.0	0.0
cimetière	asylum	0.22	–	–	–	–
cimetière	monticule	0.22	0.0	0.0	0.0	0.0
côte	forêt	0.22	0.0	0.02	0.04	0.03
refuge	moine	0.22	0.0	0.0	0.0	0.0
grue	coq	0.28	0.0	0.35	0.69	0.62
rivage	trip	0.28	0.0	0.0	0.0	0.0
garçon	sage	0.29	0.0	0.04	0.08	0.06
auto	voyage	0.33	0.0	0.04	0.03	0.03
rivage	bois	0.33	0.0	0.0076	0.01	0.0097
moine	oracle	0.39	0.0	0.01	0.01	0.01
colline	bois	0.44	0.0	0.01	0.01	0.01
garçon	coq	0.44	0.03	0.13	0.19	0.18
gars	sorcier	0.44	0.0	0.07	0.10	0.08
refuge	cimetière	0.5	0.0	0.01	0.16	0.079
fournaise	instrument	0.56	0.0	0.0	0.0	0.0
magicien	oracle	0.56	0.10	0.1	0.03	0.06
verre	bijou	0.56	0.0	0.01	0.0015	0.0058
nourriture	coq	0.61	0.0	0.03	0.03	0.03
sage	sorcier	0.83	0.05	0.08	0.07	0.08
grue	instrument	0.94	0.0	0.0	0.0012	6.91E-5
oracle	sage	1.28	0.0	0.0079	0.02	0.01
grimace	sourire	1.5	0.03	0.12	0.08	0.09
oiseau	grue	1.65	0.0	1.0	1.0	1.0
serf	esclave	1.89	0.76	0.62	0.77	0.75
frère	gars	2.0	0.0	0.07	0.21	0.16
côte	colline	2.17	0.61	0.16	0.21	0.24
midi	dîner	2.17	0.0	0.40	0.15	0.32
oiseau	coq	2.41	0.40	0.58	1.0	1.0
côte	rivage	2.5	1.0	0.96	0.96	0.97
voyage	périple	2.59	1.0	0.99	1.0	1.0
magicien	sorcier	2.67	0.98	0.59	0.61	0.67
fournaise	four	2.78	0.69	0.59	0.57	0.69
nourriture	fruit	2.78	0.0	0.19	0.50	0.36
frère	moine	2.89	1.0	0.29	0.58	0.55
colline	monticule	2.94	1.0	0.36	0.54	0.54
coussin	oreiller	3.0	0.89	1.0	1.0	1.0
instrument	outil	3.0	0.95	0.51	1.0	1.0
joyau	bijou	3.22	1.0	0.30	0.28	0.39
refuge	asile	3.28	1.0	0.65	1.0	1.0
corde	ficelle	3.33	1.0	0.95	0.37	0.69
verre	goblet	3.39	–	–	–	–
autographe	signature	3.56	1.0	1.0	1.0	1.0
forêt	bois	3.72	0.96	1.0	1.0	1.0
garçon	gars	3.83	0.98	0.37	0.46	0.50
automobile	auto	3.94	1.0	1.0	1.0	1.0