Normalisation de termes complexes par sémantique distributionnelle guidée par une ontologie

Arnaud Ferré[1, 2]
(1) MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-en-Josas, France
(2) LIMSI, CNRS, Université Paris-Saclay, 91405 Orsay, France
arnaud.ferre@universite-paris-saclay.fr

RESUME

Nous proposons dans cet article une méthode semi-supervisée originale pour la création de représentations vectorielles pour des termes (complexes ou non) dans un espace sémantique pertinent pour une tâche de normalisation de termes désignant des entités dans un corpus. Notre méthode s'appuie en partie sur une approche de sémantique distributionnelle, celle-ci générant des vecteurs initiaux pour chacun des termes extraits. Ces vecteurs sont alors plongés dans un autre espace vectoriel construit à partir de la structure d'une ontologie. Pour la construction de ce second espace vectoriel ontologique, plusieurs méthodes sont testées et comparées. Le plongement s'effectue par entraînement d'un modèle linéaire. Un calcul de distance (en utilisant la similarité cosinus) est enfin effectué pour déterminer la proximité entre vecteurs de termes et vecteurs de concepts de l'ontologie servant à la normalisation. La performance de cette méthode a atteint un rang honorable, ouvrant d'encourageantes perspectives.

ABSTRACT

Normalization of complex terms with distributional semantics guided by an ontology

We propose in this paper an original and semi-supervised method for computing continuous vector representations for terms (complex or non-complex) in a semantically structured space relevant to a task of normalization of terms of designating entities in a corpus. Our method partially based on a distributional semantics approach, which generates initial vectors for each of the extracted terms. Then, these vectors are embedded into another vector space constructed from the structure of the ontology. For the construction of this second ontological vector space, several methods are tested and compared. This embedding is carried out by training a multivariate linear regression. Finally, we apply a distance calculation (using cosine similarity) to determine the proximity between vector of terms and vector of concepts used for the normalization. The performance of this method reaches an honorable rank, opening up encouraging perspectives.

MOTS-CLES : TAL, extraction d'information, étiquetage par une ontologie, espace vectoriel, sémantique distributionnelle, modèle linéaire

KEYWORDS: NLP, information extraction, ontology-based tagging, vector space, distributional semantics, multivariate linear regression

1 Introduction

La normalisation consiste à annoter des mots ou des combinaisons de mots avec une ou plusieurs catégories sémantiques. La normalisation rencontre plusieurs difficultés, comme la variabilité importante de la morphologie des termes, qu'ils soient représentés par un mot ou par plusieurs (Nazarenko et al., 2006). Les termes multi-mots présentant des structures morphosyntaxiques variées et des imbrications complexes, tels que les groupes nominaux complexes (*complex noun phrase*) sont particulièrement difficiles à étiqueter par des catégories. En effet, la morphologie des labels des catégories sémantiques n'est pas nécessairement proche de la morphologie des termes à annoter. Or, dans les textes de la littérature spécialisée, les groupes nominaux complexes se retrouvent en abondance (Maniez, 2007). Une approche par similarité morphologique entre terme et étiquette sémantique apparaît donc limitée pour effectuer cette tâche (Golik et al., 2011). Une autre difficulté vient du nombre important de catégories sémantiques souvent à utiliser, rendant une approche par classification supervisée coûteuse en annotation manuelle (e.g. plus de trois millions de concepts dans le metathesaurus biomédical de l'UMLS de catégories pour le thésaurus biomédical UMLS (McCray, 1989)).

Une ontologie informatique est une représentation formelle et partielle de propriétés générales d'un ensemble de connaissances, et représente donc un objet manipulable par un programme informatique. Dans le cas de cette article, nous n'emprunterons aux ontologies que la notion de concept (e.g. un concept labélisé 'Chien' qui représente l'abstraction de tous les animaux domestiques de l'espèce *Canis lupus*) et celle de relation hiérarchique (e.g. le concept 'Labrador' hérite du concept 'Chien' qui hérite lui-même du concept 'Animal', etc.). Les concepts d'une ontologie peuvent alors être utilisés pour représenter des catégories sémantiques de façon formelle et structurée.

Une alternative aux approches par classification supervisée ou aux approches morphologiques consiste à calculer la proximité sémantique entre des termes par sémantique distributionnelle. C'est une approche fondée sur la corrélation entre la similarité de sens et la similarité de distribution des unités sémantiques (mot, combinaison de mots, phrase, documents, …) (Firth, 1957; Harris, 1954). Une unité sémantique peut être représentée par un vecteur construit à partir de la distribution des informations de contexte dans lesquels elle est trouvée. La proximité des vecteurs dans cet espace est alors transposable à une proximité sémantique (Fabre et al., 2015). Il existe aujourd'hui de nombreuses méthodes de génération de tels espaces vectoriels, tel que Word2Vec (Mikolov et al., 2013), mais celles-ci se concentrent habituellement sur les jeux de données massifs (Fabre et al., 2014) dans lesquels l'information est relativement répétée.

La question qui nous intéresse ici est : comment utiliser la sémantique distributionnelle pour normaliser les termes par une ontologie ? C'est-à-dire comment relier l'information distributionnelle aux concepts d'une ontologie ? Et comment répondre à cette question pour des relativement petits corpus et un grand nombre de concepts ? Cet article propose une méthode s'appuyant sur la génération de vecteurs dans un espace vectoriel des termes (EVT) ainsi que sur l'étude et la comparaison de vecteurs générés pour former un espace vectoriel des concepts de l'ontologie (EVO), puis sur le plongement de l'EVT dans l'EVO.

2 Matériel

Les données utilisées sont celles de la tâche de catégorisation Bacteria Biotope (tâche 3) de BioNLP Shared Task en 2016 (Deléger et al., 2016). Les documents sont des références de PubMed, composées de titres et de résumés d'articles scientifiques dans le domaine de la biologie. La tâche consiste, étant donné les entités du corpus dénotant les habitats bactériens (i.e. des termes extraits), à leur assigner un concept de l'ontologie. Le corpus Bacteria Biotope est divisé en trois : le corpus d'entraînement, le corpus de développement et le corpus de test. Dans le corpus d'entraînement et le corpus de développement les concepts des entités sont données : elles nous ont servi à entraîner notre méthode. Le corpus de test est celui pour lequel les concepts sont à prédire : il nous servira à évaluer notre méthode pour la tâche de normalisation. Chacun de ces corpus a été annoté manuellement. Voici un résumé de leurs caractéristiques :

	BB			
	Entr.	Dév.	Test	**Total**
Documents	71	36	54	**161**
Mots	16 295	8 890	13 797	**38 982**
Entités	747	454	720	**1 921**
Entités distinctes	476	267	478	**1 125**
Concepts	825	535	861	**2 221**
Concepts distincts	210	122	177	**329**

TABLE 1 : Statistiques descriptives du corpus BB

En plus de ce corpus, un corpus élargi du même domaine a été utilisé pour générer des représentations vectorielles de chaque mot. Il est composé de 100 000 phrases venant de titres et de résumés d'articles scientifiques dans le domaine de la biologie disponibles sur PubMed. Cela représente un corpus de taille relativement moyenne, qui contient également une majorité de mots non-outils avec une faible fréquence d'apparition (cf. TABLE 2).

Nombre mots uniques non-outils	209 345
Nombre de mots	4 826 058
Nombre de phrases	100 250

TABLE 2 : Statistiques descriptives du corpus élargi

3 Méthode

La méthode présentée dans cet article pour normaliser des termes d'un corpus peut se décomposer en 3 étapes principales : la première étape consiste à générer un espace vectoriel dans lequel les termes du corpus sont représentés (i.e. par des vecteurs) à partir d'une méthode de sémantique distributionnelle ; la seconde étape consiste à générer un autre espace vectoriel dans lequel les concepts de l'ontologie sont représentés ; enfin la troisième étape consiste à déterminer une transformation linéaire permettant de projeter les vecteurs de termes du corpus d'entraînement issu

du premier espace vectoriel dans le second en cherchant à minimiser la distance entre les projections et les vecteurs de concepts associés aux termes, permettant ainsi de prédire des projections du corpus de test et de proposer pour chaque terme un concept en prenant le concept le plus proche dans l'espace (cf. FIGURE 1 et FIGURE 2).

3.1 Génération de l'espace vectoriel des termes (EVT)

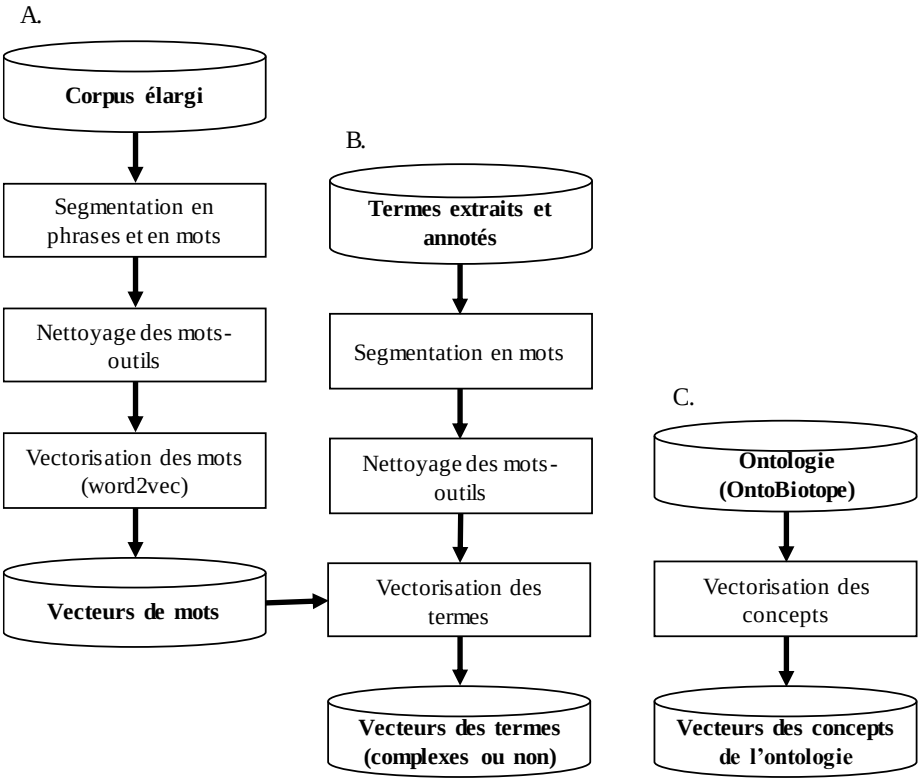

FIGURE 1 : A. Processus de création de vecteurs de mots. B. Processus de création des vecteurs de termes C. Processus de création de vecteurs de concepts

L'espace vectoriel des termes (EVT) est obtenu en générant un vecteur pour chacun des mots du corpus élargi qui comprend également les corpus d'entraînement et de développement. Pour cela, nous avons utilisé l'outil Word2Vec (Mikolov et al., 2013) en prenant pour contexte d'un mot, tous les mots contenus dans la phrase. Pour avoir suffisamment de données à entraîner pour la génération de vecteurs de mots, et aussi pour éviter de prendre en compte des fautes de frappes ou des erreurs, il est habituellement conseillé d'utiliser Word2Vec en ne prenant pas en compte les mots n'apparaissant qu'une ou deux fois dans tout le corpus. Notre corpus contenant beaucoup de mots d'intérêt à faible fréquence, nous avons fait le choix de ne pas appliquer ce seuil de fréquence. Après quelques tests de performance, une dimension de 200 a été choisie pour les vecteurs de sorties (cf. FIGURE 1A), ce qui est du même ordre que ce qui est conseillé habituellement (Mikolov et al., 2013). Pour calculer des représentations vectorielles pour des termes extraits (cf. FIGURE 1B), on peut alors commencer par les segmenter en mots. Pour chaque mot non-outil, on récupère le vecteur issu du calcul précédent. Puis on calcule le vecteur moyen :

$$v_{t_k} = \sum_{i=1}^{n_k} v_{m_i^k}/n_k \qquad (1)$$

Où v_{t_k} est le vecteur associé au terme t_k, n_k est le nombre de mots non-outils du terme t_k, $v_{m_i^k}$ est le vecteur du mot m_i^k issu de Word2Vec, et le terme t_k est tel que : $\forall i \in [1, n_k], m_i^k \in t_k$

3.2 Génération de l'espace vectoriel ontologique (EVO)

La caractéristique commune à tous les différents EVO que nous présentons ici est qu'ils ont une dimension égale au nombre de concepts différents dans l'ontologie. Pour construire les vecteurs de concepts, on initialise des vecteurs nuls possédant autant de dimension que de concepts dans l'ontologie. Chaque valeur du vecteur correspond donc à un des concepts de l'ontologie. On a donc :

$$\forall k \in [\![1, n]\!], v_{c_k} = \left(w_{c_k}^0, \dots, w_{c_k}^i, \dots, w_{c_k}^n \right) \qquad (2)$$

où v_{c_k} est le vecteur associé au concept c_k, c_k est le concept associé à la dimension k, n est le nombre de concepts dans l'ontologie et $w_{c_k}^i$ est la valeur du vecteur v_{c_k} pour la dimension i.

Nous présentons dans cet article quatre méthodes différentes pour construire les vecteurs de concepts :

1. La méthode « One-Hot » : Chaque concept est représenté par un vecteur différent, mais aucune information ne permet de représenter la structure de l'ontologie. Ici, seule la valeur associée au concept courant est non-nulle. En utilisant l'équation (2), on a donc :
$$\forall i, k \in [\![1, n]\!], w_{c_k}^i = \begin{cases} 1 \; si \; i = k \\ 0 \; sinon \end{cases} \qquad (3)$$

2. La méthode « Ancestry » : Chaque vecteur de concept possède l'information des vecteurs qui sont ses ancêtres, mais pas l'information de l'organisation de sa lignée. Pour cela, en plus de la valeur associée au concept qui est mise à 1 comme dans la méthode précédente, on va également mettre à 1 tous les parents (directs ou non) du concept courant jusqu'à la racine. En utilisant l'équation (2), on a donc :
$$\forall i, k \in [\![1, n]\!], w_{c_k}^i = \begin{cases} 1 & si \; i = k \\ 1 & si \; c_i \; ancêtre \; de \; c_k \\ 0 & sinon \end{cases} \qquad (4)$$

3. La méthode « Ancestry+ » : S'appuyant sur la méthode précédente, elle vise à éloigner chaque vecteur de concept des vecteurs de ses concepts fils et à l'inverse de le rapprocher des vecteurs de ses concepts parents. L'objectif de cette modification est que, pour un concept devant annoter correctement un terme, il est préférable d'annoter par un concept parent plutôt que par un concept fils (e.g. si le terme 'le chien du voisin' échoue à être annoté par un concept 'Chien', il vaut mieux qu'il soit annoté par un concept 'Animal' que 'Labrador'). D'après les critères d'évaluation de la tâche de normalisation, cette caractéristique pourrait donc avoir une influence positive. Pour cela, au lieu de mettre la valeur associée au concept à 1, celle-ci va prendre une valeur égale à la distance du concept de la racine. Si on définit la fonction $dist$ suivante :
$$\forall i, j \in [\![1, n]\!], dist(c_i, c_j) = distance \; de \; c_i \; à \; c_j \qquad (5)$$
$$Telle \; que : \forall k \in [\![1, n]\!], dist(c_k, c_k) = 0 \; et \; dist(c_k, parent \; de \; c_k) = 1 \qquad (6)$$

   On a alors :

$$w_{c_k}^i = \begin{cases} dist(c_k, racine) + 1 & si\ i = k \\ 1 & si\ c_i\ ancêtre\ de\ c_k \\ 0 & sinon \end{cases} \quad (7)$$

4. La méthode « Centered Node » : Cette méthode, proposée dans (Eidoon et al., 2007) permet d'augmenter l'information portée par chaque vecteur sur la structure de l'ontologie. Plus précisément, chaque vecteur de concept porte l'information de sa distance à chaque concept de sa lignée (ses ancêtres comme ses descendants). Pour cela, la valeur du concept courant est là aussi mise à 1. De façon symétrique (i.e. vers la racine ou vers les feuilles), plus l'on va s'éloigner de la valeur du concept courant en restant sur la lignée du concept, plus ses valeurs associées aux fils et aux parents vont diminuer. En utilisant la fonction définie en (5) :

$$w_{c_k}^i = \begin{cases} 1 & si\ i = k \\ 1/(dist(c_i, c_k) + 1) & si\ c_i\ ancêtre\ ou\ descendant\ de\ c_k \\ 0 & sinon \end{cases} \quad (8)$$

3.3 Entraînement et prédiction

L'objectif de la phase d'entraînement est de déterminer une transformation de l'EVT vers l'EVO qui minimise toutes les distances entre les vecteurs des termes résultants dans l'EVO et les vecteurs des concepts associés. Dans cet article, une transformation linéaire est étudiée car nous faisons l'hypothèse qu'il y a une certaine similitude de répartition entre les vecteurs de termes dans l'EVT et les vecteurs de concepts associés dans l'EVO. Autrement dit, une transformation non-linéaire pourrait fortement déformer la répartition des vecteurs de termes dans l'EVO pour s'adapter aux données d'entraînement peu nombreuses et ne recouvrant qu'une faible partie des annotations pouvant être détectées dans les textes ciblés. Cet entraînement vise à obtenir les meilleurs paramètres pour approximer l'équation matricielle suivante :

$$Y = X.B + U \quad (9)$$

où Y est une matrice formée d'une série de vecteurs de concepts, X est une matrice formée d'une série de vecteurs de termes (où la ième ligne de X représente le vecteur d'un terme qui est annoté par un concept qui a pour vecteur la ième ligne de Y), B est la matrice contenant les paramètres qui sont à estimer, et U est une matrice contenant une distribution gaussienne multivariée. Cet entraînement est réalisé sur les corpus d'entraînement et de développement (cf. FIGURE 2A).

La matrice obtenue nous permet de concevoir une fonction de transformation linéaire, afin de permettre de prédire de nouveaux vecteurs associés aux termes du corpus d'essai exprimé dans l'EVO :

$$f: \begin{pmatrix} EVT \rightarrow EVO \\ v_{\text{term}} \ \rightarrow \ v'_{\text{term}} = f(v_{\text{term}}) \end{pmatrix} \quad (10)$$

Où v_{term} est un vecteur de terme dans l'EVT et v'_{term} est le vecteur résultant du même terme projeté dans l'EVO. Pour satisfaire aux exigences de la tâche d'évaluation, le vecteur de concept le plus proche (en terme de similarité cosinus) de v'_{term} est choisi pour le terme annoté (cf. FIGURE 2B).

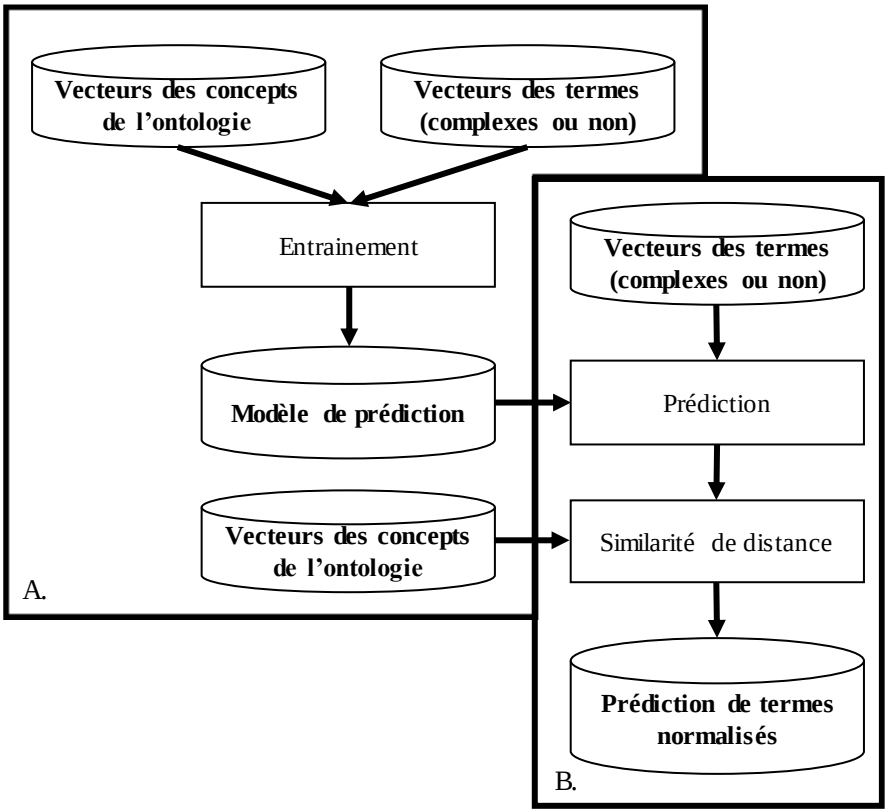

FIGURE 2 : A. Processus d'entraînement pour déterminer une transformation de l'EVT vers l'EVO ; B. Processus de prédiction des concepts associées aux termes extraits

4 Résultats

Dans cette section, nous présentons les résultats intermédiaires obtenus pour les deux espaces vectoriels générés, ainsi que les résultats finaux pour la tâche de normalisation en fonction de la méthode de génération de l'EVO utilisée.

4.1 Espace vectoriel des termes (EVT) généré

En dépit de la faible fréquence d'apparition des mots du corpus élargi (cf. TABLE 2), les vecteurs de mots obtenus semblent présenter des proximités relativement cohérente du point de vue de la similarité sémantique des termes associés. De plus, la méthode utilisée pour former des vecteurs pour les termes complexes semblent elle aussi pouvoir conserver cette cohérence. Les cas où la tête sémantique est commune dans ces termes semble permettre de retrouver une similarité cohérente (cf. TABLE 3).

cell	similarité
HCE cell	0,9999
13C-labeled cell	0,9998
parietal cell	0,9989
Schwann cell	0,9965
CD8+ T cell	0,9770
PMN cell	0,9669
macrophage cell	0,9473
J774 cell line	0,9230
eukaryotic cell	0,9114
macrophage-like cell line J774	0,9086

TABLE 3 : Termes les plus proches du terme 'cell' dans l'EVT

Il semble également que des différences morphologiques n'empêchent pas l'agglomération de vecteurs de termes proches (cf. TABLE 4), ce qui était une des propriétés désirées.

younger ones	similarité
children less than five years of age	0,8087
children less than 2 years of age	0,8060
children less than two years of age	0,7995

TABLE 4 : Termes les plus proches du terme 'younger ones' dans l'EVT

Néanmoins, la cooccurrence de certains mots semble agglomérer certains termes de concept différent. Deux mots apparaissant fréquemment dans des contextes communs se retrouvent alors avec des vecteurs similaires. Cette similarité persiste alors également lors du calcul des vecteurs de termes. C'est par exemple le cas pour les termes relatifs au poisson et ceux relatifs aux fermes de poissons (cf. TABLE 5). Ces représentations sont moins satisfaisantes car elles ne permettent pas de différencier les concepts sous-jacents.

fish	similarité
fish farming	0,9875
fish farm	0,9170
disease-free fish farm	0,9124
fish farm sediments	0,8683
healthy fish	0,8145

TABLE 5 : Termes les plus proches du terme 'fish' dans l'EVT

4.2 Espace vectoriel ontologique (EVO) généré

On peut estimer la qualité des vecteurs de concepts créés en observant la cohérence entre la proximité vecteur/vecteur (en terme de similarité cosinus) et leur position dans l'ontologie.

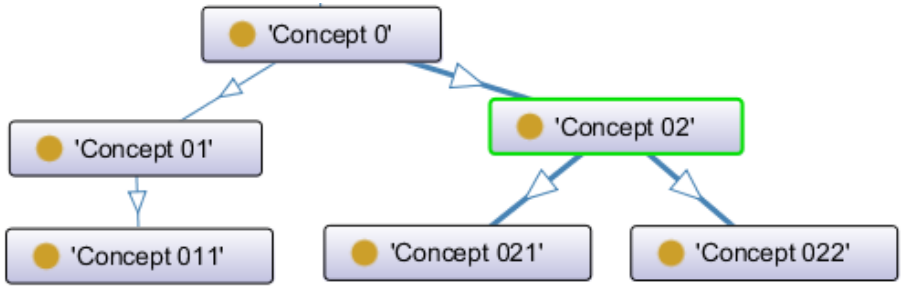

FIGURE 3 : Représentation conceptuelle d'une ontologie (figure générée par le logiciel Protégé (Gennari et al., 2003))

Pour cela, nous pouvons étudier ce qu'il se produit sur une ontologie abstraite (cf. FIGURE 3) :

1. La méthode « One-Hot » : Cette méthode génère des vecteurs de concepts tous orthogonaux entre eux. La similarité cosinus entre 2 vecteurs de concepts distincts est donc égale à 0, qu'il y ait ou non une relation hiérarchique qui les relient (cf. TABLE 6). Cette méthode ne permet donc pas de représenter la structure de l'ontologie.

Concept 02	**similarité**
Concept 02	1,00
Concept 021	0,00
Concept 022	0,00
Concept 0	0,00
Concept 01	0,00
Concept 011	0,00

TABLE 6 : Proximité du 'Concept 02' aux concepts voisins pour la méthode One-Hot

2. La méthode « Ancestry » : A partir de l'étude des vecteurs de concepts les plus proches d'un vecteur de concept quelconque, il semble ici possible de retrouver les concepts parents et les enfants de celui-ci (cf. TABLE 7). Ensuite, ce sont ses concepts parents puis ses concepts frères qui se retrouvent les plus proches. Il semble alors possible de retrouver l'ensemble de la structure de l'ontologie grâce à l'ensemble des vecteurs de concepts générés par cette méthode.

Concept 02	**similarité**
Concept 02	1,00
Concept 021	0.82
Concept 022	0.82
Concept 0	0.71
Concept 01	0.50
Concept 011	0.41

TABLE 7 : Proximité du 'Concept 02' aux concepts voisins pour la méthode Ancestry

3. La méthode « Ancestry+ » : Comme attendu, on observe bien un changement dans la proximité des vecteurs de concepts (cf. TABLE 8). Un concept se retrouve plus proche de ses parents que de ses enfants, puis relativement plus éloignés, on retrouve le concept frère plus le concept neveu. On observe également un écartement des concepts entre eux : en terme de

similarité cosinus, les concepts les plus proches se retrouvent à une distance relativement importante vis-à-vis de la méthode précédente.

Concept 02	similarité
Concept 02	1,00
Concept 0	0.45
Concept 021	0.40
Concept 022	0.40
Concept 01	0.20
Concept 011	0.13

TABLE 8 : Proximité du 'Concept 02' aux concepts voisins pour la méthode Ancestry+

4. La méthode « Centered Node » : On retrouve avec cette méthode l'ordre de proximité de la méthode Ancestry (i.e. un concept est d'abord plus proche de ses fils puis de ses parents). Néanmoins, les concepts fils et parents d'un concept courant semblent être à une distance relativement similaire du concept courant. Cette distance semble relativement faible également. Par contre, les concepts frères et neveux se retrouvent beaucoup plus éloignés (cf. TABLE 9).

Concept 02	similarité
Concept 02	1,00
Concept 021	0.76
Concept 022	0.76
Concept 0	0.74
Concept 01	0.15
Concept 011	0.11

TABLE 9 : Proximité du 'Concept 02' aux concepts voisins pour la méthode Centered-Node

4.3 Normalisation

Nous allons examiner dans cette section la performance d'un système qui repose sur les espaces vectoriels construits comme expliqué dans les sections précédentes. Rappelons qu'étant donné un terme extrait du texte, sa représentation vectorielle dans l'EVT est obtenue selon l'équation (1), puis son vecteur dans l'EVO est obtenu par la fonction définie en (10) à partir de l'équation (9). Le concept de l'ontologie qui est assigné au terme examiné est alors celui dont le cosinus entre son vecteur et celui de la représentation vectorielle du terme dans l'EVO est maximal.

Pour évaluer la performance des systèmes participants à la tâche de catégorisation de Bacteria Biotope de BioNLP-ST 2016, une mesure de similarité sémantique est implémentée sur le site du challenge BioNLP-ST 2016. La mesure utilisée est celle définie par Wang et al. en 2007 (Wang et al., 2007), avec le paramètre de poids à 0,65. Avec cette mesure, nous pouvons calculer une *baseline* en attribuant à tous les termes le concept « bacteria habitat », qui est la racine de la hiérarchie de l'ontologie OntoBiotope. La mesure trouvée est alors de 32%. Deux équipes ont participé à cette tâche de BioNLP-ST 2016 et ont obtenu les résultats rapportés dans la TABLE 10.

Equipe	Score
BOUN	0,62
LIMSI	0,44
Baseline	0,32

TABLE 10 : Performance dans la tâche d'assignation de catégorie de Bacteria Biotope dans BioNLP-ST 2016

On observe une performance globale de nos méthodes au-dessus de la *baseline* (cf. TABLE 10 et TABLE 11). Les 2 meilleures méthodes sont Ancestry et Centered-Node, avec des scores très proches. La moins bonne méthode est Ancestry+, derrière celle qui porte le moins d'information One-Hot.

Méthode	Score
Ancestry	**0,60**
Centered-Node	0,59
One-Hot	0,55
Ancestry+	0,51

TABLE 11 : Performance de notre système pour la tâche d'assignation de catégorie de Bacteria Biotope dans BioNLP-ST 2016, en fonction des méthodes de génération d'EVO utilisée

Notre méthode globale avec la méthode de représentation de l'EVO Ancestry a donc obtenu un résultat de 60%, bien au-dessus de la baseline, et très proche de la première équipe (Tiftikci et al., 2016). Ce score est significativement au-dessus de la méthode du LIMSI (Grouin, 2016), méthode basée sur une approche morphologique.

5 Discussions

La qualité globale des espaces vectoriels générés n'a été évaluée que via la tâche de normalisation présentée. Il serait pertinent d'examiner les effets du système avec d'autres tâches. Nous prévoyons d'effectuer cela dans de futurs travaux.

Les résultats inférieures à la méthode One-Hot de la méthode Ancestry+ peut laisser supposer que la réorganisation de l'espace opéré par cette méthode est soit non-pertinente, soit trop brutale (cf. répartition en distance des concepts dans la TABLE 8). A l'inverse, la One-Hot reste une méthode aux résultats honorables alors qu'elle n'apporte aucune information sur la structure de l'ontologie. Ce résultat montre qu'avec l'ajout de l'information sur la structure de l'ontologie, les représentations Ancestry et Centered-Nodeont ont pu apporter un gain de 4-5 points dans la tâche. Reste à savoir s'il existe des représentations qui permettraient d'améliorer encore plus ces résultats, ou si l'on se rapproche d'un certain palier.

Un des plafonds de la méthode présentée dans cet article est dû au fait que, pour cette tâche de normalisation, un terme peut être normalisé par plusieurs concepts de l'ontologie (ex : le terme « school age children with wheezing illness » devrait être normalisé par le concept <OBT:002307: pediatric patient> ainsi que le concept <OBT:002187: patient with disease>).

6 Conclusion et perspectives

L'objectif de cet article était de proposer une approche pertinente et originale pour la création de représentations vectorielles pour des termes (complexes ou non) dans un espace sémantique pertinent pour une tâche d'assignation de concepts à des termes désignant des entités dans un corpus. De plus, il visait à proposer une méthode capable de s'adapter à un corpus spécialisé de petite taille où les termes d'intérêt ont un faible nombre d'occurrences. Les résultats, proches des meilleurs des participants à la tâche de catégorisation de BioNLP-ST 2016, semble ouvrir d'encourageantes perspectives.

Pour de futurs travaux, il serait pertinent d'appliquer des méthodes d'évaluation globale de la qualité des espaces vectoriels générés. En particulier, cela permettrait d'évaluer plus exhaustivement les processus intermédiaires et d'observer avec plus de précision l'impact des modifications sur leurs paramètres internes. De nouvelles méthodes plus élaborées pourraient alors être envisagées pour améliorer les résultats. Par exemple, la méthode utilisée ici pour générer les vecteurs de l'EVT pourrait être améliorée pour prendre en compte la structure syntaxique des termes. Cela pourrait résoudre les problèmes de similarité sémantique entre « fish » et « fish farm » (cf. TABLE 5). Plus généralement, il serait intéressant de comparer les effets d'autres méthodes de génération de vecteurs de termes sur la performance du système présenté.

Appliquer cette même méthode avec une ontologie dont tous les concepts sont définis pourrait être bénéfique. Si de telles ontologies sont rares, il pourrait être tout de même intéressant de tester en ne prenant en compte qu'une partie des concepts dont les intersections avec d'autres ne sont pas définies.

Enfin, malgré la limitation inhérente des méthodes de normalisation basées sur la morphologie des mots, celles-ci pourraient néanmoins être utilisées pour effectuer une pré-normalisation du corpus. En conséquence, on pourrait envisager d'utiliser ces annotations pour entraîner la méthode au lieu d'utiliser une annotation manuelle. Ainsi, cela transformerait cette méthode en une méthode non-supervisée.

Remerciements

This work is supported by the "IDI 2015" project funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02.

Références

DELGER L., CHAIX E., BA M., FERRE A., BESSIERES P., NEDELLEC C. (2016). Overview of the Bacteria Biotope Task at BioNLP Shared Task 2016.

EIDOON Z., YAZDANI N., OROUMCHIAN F. (2007). A Vector Based Method of Ontology Matching (p. 378‑381). IEEE.

FABRE C., HATHOUT N., SAJOUS F., TANGUY L. (2014). Ajuster l'analyse distributionnelle à un corpus spécialisé de petite taille. In *21e Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2014)* (p. 266–279).

FABRE C., LENCI A. (2015). Distributional Semantics Today Introduction to the special issue. *Traitement Automatique des Langues*, *56*(2), 7–20.

FIRTH J. R. (1957). The technique of semantics. *Oxford University Press*. London.

GENNARI J. H., MUSEN M. A., FERGERSON R. W., GROSSO W. E., CRUBÉZY M., ERIKSSON H., … TU S. W. (2003). The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies*, *58*(1), 89–123.

GOLIK W., WARNIER P., NÉDELLEC C. (2011). Corpus-based extension of termino-ontology by linguistic analysis: a use case in biomedical event extraction. In *WS 2 Workshop Extended Abstracts, 9th International Conference on Terminology and Artificial Intelligence* (p. 37–39).

GROUIN C. (2016). Identification of Mentions and Relations between Bacteria and Biotope from PubMed Abstracts. *ACL 2016*, 64.

HARRIS Z. S. (1954). Distributional Structure. WORD, *10*(2‑3), 146‑162.

MANIEZ F. (2007). Prémodification et coordination : quelques problèmes de traduction des groupes nominaux complexes en anglais médical. *ASp*, (51‑52), 71‑94.

MCCRAY A. T. (1989). The UMLS Semantic Network. In *Proceedings/the... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care* (p. 503–507). American Medical Informatics Association.

MIKOLOV T., CHEN K., CORRADO G., DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

NAZARENKO A., NEDELLEC C., ALPHONSE E., AUBIN S., HAMON T., MANINE A.-P. (2006). Semantic annotation in the alvis project. In *International Workshop on Intelligent Information Access (IIIA)* (p. 5–pages).

TIFTIKCI M., SAHIN H., BÜYÜKÖZ B., YAYIKÇI A., OZGÜR A. (2016). Ontology-based Categorization of Bacteria and Habitat Entities using Information Retrieval Techniques. *ACL 2016*, 56.

WANG J. Z., DU Z., PAYATTAKOOL R., YU P. S., CHEN C.-F. (2007). A new method to measure the semantic similarity of GO terms. *Bioinformatics*, *23*(10), 1274‑1281.