Apprendre des représentations jointes de mots et d'entités pour la désambiguïsation d'entités

Jose G. Moreno[1] Romaric Besançon[2] Romain Beaumont[3] Eva D'hondt[3]
Anne-Laure Ligozat[3,4] Sophie Rosset[3] Xavier Tannier[3,5] Brigitte Grau[3,4]
(1) Université Paul Sabatier, IRIT, 118 Route de Narbonne, F-31062 Toulouse, France
(2) CEA, LIST, Laboratoire Vision et Ingénierie des Contenus., F-91191 Gif-sur-Yvette, France
(3) LIMSI, CNRS, F-91405 Orsay, France, Université Paris-Saclay
(4) ENSIIE
(5) Univ. Paris-Sud
firstname.lastname@{irit.fr[1],cea.fr[2],limsi.fr[3]}

RÉSUMÉ

La désambiguïsation d'entités (ou liaison d'entités), qui consiste à relier des mentions d'entités d'un texte à des entités d'une base de connaissance, est un problème qui se pose, entre autre, pour le peuplement automatique de bases de connaissances à partir de textes. Une difficulté de cette tâche est la résolution d'ambiguïtés car les systèmes ont à choisir parmi un nombre important de candidats. Cet article propose une nouvelle approche fondée sur l'apprentissage joint de représentations distribuées des mots et des entités dans le même espace, ce qui permet d'établir un modèle robuste pour la comparaison entre le contexte local de la mention d'entité et les entités candidates.

ABSTRACT

Combining Word and Entity Embeddings for Entity Linking.

The correct identification of the link between an entity mention in a text and a known entity in a large knowledge base is important in information retrieval or information extraction. However, systems have to deal with ambiguity as numerous entities could be linked to a mention. This paper proposes a novel method for entity disambiguation which is based on the joint learning of embeddings for the words in the text and the entities in the knowledge base. By learning these embeddings in the same space we arrive at a more conceptually grounded model that can be used for candidate selection based on the surrounding context.

MOTS-CLÉS : Liaison d'entité, Embeddings de mots et d'entités, Extraction d'information.

KEYWORDS: Entity Linking, Word and Entity Embeddings, Information Extraction.

1 Introduction

La liaison d'entités (*entity linking*) consiste à relier des mentions d'entités d'un texte à des entités d'une base de connaissance (BC) si elles existent [1] (Ling *et al.*, 2015; Shen *et al.*, 2015), dans le but de normaliser ces mentions. C'est un problème qui se pose, par exemple, pour le peuplement de bases de connaissances par extraction d'information à partir de textes. Cette tâche est parfois

1. et, dans le cas contraire, à regrouper les mentions faisant référence à la même entité.

considérée comme un cas particulier dans un cadre plus général de désambiguïsation de concepts en fonction d'une base de connaissance, qu'il s'agisse d'entités nommées ou d'expressions nominales (par exemple, Wikify (Mihalcea & Csomai, 2007) ou Babelfy (Moro *et al.*, 2014)).

Les méthodes de liaison d'entités sont généralement décomposées en trois sous-tâches (Ji *et al.*, 2014) : 1) l'identification des mentions d'entités dans les textes devant être reliées à la base de connaissances ; 2) la recherche d'entités candidates de la base de connaissances pour chaque mention ; 3) la sélection de l'entité correcte parmi les candidates. L'un des principaux défis de la tâche est la très grande quantité d'entités présentes dans la base, de sorte que chaque mention peut être reliée à différentes entités et la référence correcte ne peut être déduite que du contexte textuel des mentions.

Ainsi, dans la phrase "Koirala n'a pourtant appris la nouvelle que grâce aux tweets de son homologue indien, Narendra Modi.", la mention "Koirala" est ambiguë car elle peut faire référence à l'actrice népalaise "Manisha Koirala", la "famille Koirala", une famille importante dans les milieux politiques du Népal, "Saradha Koirala", une poète d'ascendance népalaise, ou "Sushil Koirala", le premier ministre du Népal entre 2014 et 2015.

Cet article porte principalement sur la dernière étape du processus, la sélection du meilleur candidat. La plupart des approches pour ce problème s'appuient sur les mots et considèrent les mots comme des unités atomiques pour représenter l'information sur la mention et son contexte. Nous proposons une nouvelle approche, plus sémantique, fondée sur l'apprentissage joint des représentations distribuées (*embeddings*) des mots et des entités dans le même espace. Les avantages de l'apprentissage joint sont multiples : 1) Les représentations des mots ainsi créées sont mieux ancrées sémantiquement car leur contexte (pendant l'entraînement) peut contenir des vecteurs de concepts qui permettent de passer outre les variations de surface ; 2) Les représentations des entités sont apprises sur un grand corpus qui les mentionne avec une plus grande fréquence que si elles étaient apprises à partir de la base de connaissance directement ; 3) Comme les représentations des mots et des entités sont apprises dans un même espace, nous pouvons utiliser une mesure de similarité simple pour calculer la proximité d'une mention d'entité et des mots de son contexte dans le texte avec les entités de la BC.

Dans cet article, nous présentons les contributions suivantes :

— Le modèle EAT qui apprend simultanément des représentations distribuées pour les mots et les entités de la base (Section 3) ;

— Un système global de liaison d'entités, intégrant notre modèle EAT (Section 4) ; ce modèle pourrait être intégré comme un trait supplémentaire dans toutes approches supervisées.

Une évaluation utilisant la base Freebase et l'environnement d'évaluation de la campagne TAC 2015 "Entity Discovery and Linking" framework (Section 5) montre des résultats ($P(all) = 0,731$) qui dépassent la *baseline* et sont comparables avec les meilleurs systèmes.

2 État de l'art

Les méthodes d'apprentissage proposées pour la désambiguïsation d'entités peuvent être distinguées selon le degré de supervision qu'elles nécessitent. Les approches non supervisées s'appuient sur des mesures de similarité entre les mentions dans le texte et les entités dans la base de connaissance (Cucerzan, 2007; Han & Zhao, 2009). Ces mesures se fondent en général sur des recouvrements de contextes, mais peuvent combiner plusieurs scores. Elles ont pour avantage d'être simples à implémenter mais elles obtiennent des performances faibles en comparaison d'approches supervisées

(Cassidy *et al.*, 2011) qui s'appuient sur des classifieurs binaires (Lehmann *et al.*, 2010; Varma *et al.*, 2009) ou des modèles d'ordonnancement (Shen *et al.*, 2012; Cao *et al.*, 2007).

Plusieurs travaux récents se sont concentrés sur le calcul de similarité entre des mentions d'entités et des entités. Babelfy (Moro *et al.*, 2014) utilise des algorithmes de marche aléatoire pour projeter des mentions dans des bases de connaissances comme Wordnet [2](Miller, 1995), Babelnet [3](Navigli & Ponzetto, 2012) et YAGO2 [4](Hoffart *et al.*, 2013).

L'apprentissage de représentations de mots a, par ailleurs, récemment permis d'obtenir de très bons résultats pour différentes tâches de traitement automatique de la langue (Mikolov *et al.*, 2013). Les plongements lexicaux (*Word embeddings*) représentent un ensemble de méthodes non supervisées, fondées sur l'observation de régularités dans de très grandes quantités de textes, permettant d'apprendre des vecteurs compacts qui obtiennent de meilleurs résultats que les représentations fondées sur des décomptes (Baroni *et al.*, 2014). Appliquées à une base de connaissance (Bordes *et al.*, 2013), les informations de la base, habituellement représentées sous forme de triplet $(sujet, predicat$ [5]$, objet)$, sont transformées en vecteurs de faible dimension. Cette transformation est obtenue par optimisation d'une fonction qui donne un score élevé lorsque les triplets sont présents dans la base et un score bas sinon. S'inspirant de ce travail, Wang *et al.* (2014) définissent une fonction à trois composants en charge de l'optimisation des représentations des mots apprises à partir de textes, des représentations des connaissances de la base et leur alignement. Cette technique permet de combiner textes et connaissances structurées, ce qui engendre un espace de représentation unique pour les mots, les entités et les relations. La plupart de ces travaux cherchent à compléter une base de connaissance, mais cependant quelques-uns s'intéressent à la tâche de liaison référentielle.

Une extension des travaux de Wang *et al.* (2014) a été développée en parallèle par Fang *et al.* (2016) et Yamada *et al.* (2016). Les auteurs appliquent leurs modèles à plusieurs collections de liaison d'entité. Cependant, comme Wang *et al.* (2014), ces travaux n'utilisent pas directement le contexte d'une mention d'entité pour construire la représentation vectorielle, mais une fonction d'alignement appariant la mention et l'entité. Pappu *et al.* (2017) préfèrent utiliser la représentation des documents de Le & Mikolov (2014) pour représenter conjointement les pages Wikipédia et les mots. Dans les deux cas, l'espace joint des entités et des mots est utilisé pour calculer des similarités entre eux. Zwicklbauer *et al.* (2016) ont intégré des représentations d'entités pour la désambiguïsation d'entités. Néanmoins, ces entités ont été apprises sur des documents qui ne contiennent que des entités et ne peuvent donc pas être utilisées pour aligner des représentations de mots et entités apprises séparément.

3 Représentation jointe des mots et des entités

Nous avons choisi d'apprendre de façon jointe et simultanée les représentations des mots et des entités, permettant ainsi une normalisation plus intégrée. Dans cette section, nous présentons un modèle capable de combiner les vecteurs de mots et d'entités en utilisant uniquement leurs contextes. Ce modèle peut ainsi être considéré comme une extension du modèle original de (Mikolov *et al.*, 2013) ou de sa variation (Le & Mikolov, 2014).

2. https://wordnet.princeton.edu/
3. http://babelnet.org/
4. http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/#c10444
5. un prédicat est aussi nommé relation

3.1 Définitions

Un corpus est une séquence de mots w ou de textes d'ancrage a (*anchor texts*), où les mots appartiennent au vocabulaire V et un texte d'ancrage $a = (w, e)$ est une paire composée d'un mot $w \in (V \cup \emptyset)$ et d'une entité $e \in \xi$. ξ est l'ensemble des entités de la base de connaissances. Nous utilisons des caractères gras $\boldsymbol{e}$, $\boldsymbol{w}$ pour représenter les vecteurs correspondant à ces éléments ($\boldsymbol{e}$, $\boldsymbol{w} \in \mathbb{R}^d$, où $\mathbb{R}^d$ est appelé l'espace joint et d est le nombre de dimensions).

3.2 Extension du texte d'ancrage

Pour créer $\boldsymbol{w}$ et $\boldsymbol{e}$ dans le même espace, nous introduisons le concept de texte d'ancrage étendu (EAT) de manière à combiner l'entité avec ses textes d'ancrage et l'introduire dans le corpus. Pour obtenir les EATs, la mention d'un texte d'ancrage a_i est redéfinie par $a'_{ij} = (w'_i, e_j)$ où $w'_i = w_i$ si w_i est non vide, sinon w'_i est égal à l'ensemble des mots désignant e_j. La figure 1 illustre cette transformation. Nous redéfinissons un corpus comme une séquence de mots et d'EATs a'. Le vocabulaire est donc maintenant défini par $\mathbb{F} = \{V \cup \xi\}$, et l'ensemble des *embeddings* à apprendre contient à la fois les mots, qui incluent les mentions d'entités, et les entités de la base de connaissance.

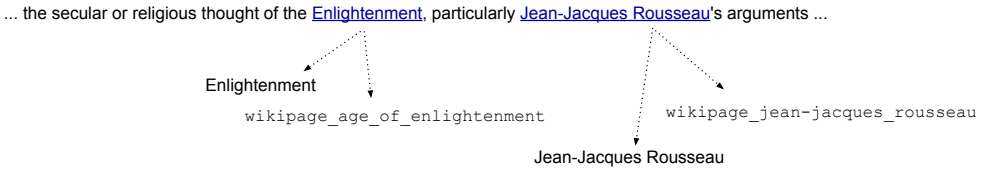

FIGURE 1 – Illustration de la décomposition mention-entité dans le modèle EAT

3.3 Le modèle EAT

L'objectif de notre modèle est de construire une représentation sémantique de chaque élément de $\mathbb{F}$ fondée sur les mots/EATs qui l'entourent dans les phrases. Comme dans Mikolov *et al.* (2013), nous définissons la probabilité entre deux éléments du corpus par l'équation 1 :

$$p(c_o|c_i) = \sum_{f_o \in c_o} \sum_{f_i \in c_i} \frac{exp(\boldsymbol{f_o}^T \boldsymbol{f_i})}{\sum_{j=1}^{|\mathbb{F}|} exp(\boldsymbol{f_j}^T \boldsymbol{f_i})} \quad (1)$$

où f sont les éléments de $\mathbb{F}$ et peuvent correspondre à un mot ou une entité, et c sont les mots ou les EATs du corpus. Il faut noter que si c_o et c_i sont des mots, l'équation 1 devient la fonction softmax entre deux mots et la double somme disparaît. Le processus d'optimisation consiste à maximiser la probabilité moyenne définie par l'équation 1 sur le corpus.

L'implémentation du modèle EAT n'entraîne pas d'importants changements dans l'implémentation actuelle de Mikolov *et al.* (2013) telle que Word2Vec [6]. Comme l'équation 1 est équivalente à celle proposé par Mikolov *et al.* (2013) quand c est un mot, nous avons juste à l'adapter quand c est un

6. Nous avons utilisé Gensim et Hyperwords, disponibles sur `https://radimrehurek.com/gensim/models/word2vec.html` et `https://bitbucket.org/omerlevy/hyperwords` respectivement.

EAT. L'adaptation consiste à étendre c par les combinaisons possibles en gardant le contexte statique, c'est-à-dire que le contexte est le même pour le mot ou l'entité. De manière similaire, si c fait partie du contexte, c'est le contexte qui doit être étendu. La figure 2 montre l'expansion produite quand les vecteurs de mot et d'entité liés à la mention de l'entité "Enlightenment" sont appris pendant la phase d'entraînement. Seules les mentions d'entités sont étendues et de ce fait appartiennent à plusieurs passes d'entraînement, i.e. une passe pour le texte de l'EAT et une passe différente pour son entité.

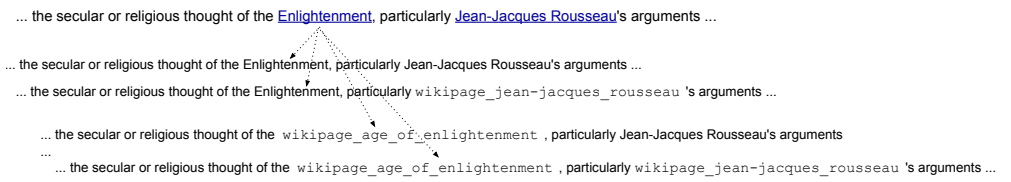

FIGURE 2 – Expansion du contexte pour calculer les différents types d'*embeddings* dans le modèle EAT

.

Le modèle EAT peut être appris en exploitant des corpus existants qui contiennent des textes d'ancrage et les méthodes et *embeddings* fondées sur la méthode originale de Mikolov *et al.* (2013) sont directement utilisables dans notre modèle, dont les modèles skip-gram et CBOW. Enfin, les *embeddings* de mots et d'entités sont appris en fonction de leurs contextes communs, et leur utilisation ne nécessite pas de stratégie d'alignement comme dans (Fang *et al.*, 2016; Yamada *et al.*, 2016), qui se rapprochent de techniques de supervision distante (Mintz *et al.*, 2009).

4 Système de désambiguïsation d'entités nommées

Le système de liaison d'entités s'appuie sur une architecture standard (Ji *et al.*, 2014) composée de deux étapes principales : pour une mention d'entité repérée dans son contexte textuel, un premier traitement permet de générer les entités candidates possibles pour la désambiguïsation de cette mention ; un second module se charge de choisir l'entité la plus pertinente parmi les entités candidates.

4.1 Génération des entités candidates

La génération des entités candidates repose sur le repérage des mentions d'entité dans le texte. Comme nous nous concentrons dans cette étude sur la désambiguïsation des entités nommées et non sur leur reconnaissance, nous considérons que les positions des mentions d'entité à désambiguïser sont données en entrée du système (nous considérons ici les formes explicites d'entités nommées, en ignorant les mentions en co-référence nominales ou pronominales). Une première étape de typage des mentions d'entité est effectuée, de façon à leur attribuer un des types attendus (Personne, Lieu, Organisation...) [7]. Deux extensions des mentions d'entité sont effectuées, qui peuvent être considérées comme des formes simples de résolution de co-référence : (1) si la mention d'entité est un

7. Pour le typage des entités nommées, nous avons utilisé MITIE https ://github.com/mit-nlp/MITIE .

acronyme, on recherche dans le texte d'autres mentions de même type dont les initiales correspondent à l'acronyme ; (2) on recherche dans le texte d'autres mentions qui incluent (comme sous-chaînes de caractères) la mention visée. Ces formes sont ajoutées comme des variations de la mention d'entité.

Ensuite, les entités candidates sont trouvées en comparant la mention avec les noms des entités de la base de connaissances. Nous utilisons pour ce faire les stratégies suivantes :

— égalité entre la mention et le nom d'une entité de la BC ;
— égalité entre la mention et une variation du nom d'une entité de la BC ;
— inclusion de la mention dans un nom d'une entité de la BC ;
— similarité entre la mention et un nom d'une entité de la BC. Nous avons retenu la distance de Levenshtein, qui est bien adaptée pour pallier des problèmes de coquilles ou de variations simples de noms. Nous avons fixé un seuil maximum de 2 pour considérer qu'une entité est candidate.
— modèle de recherche d'information : toutes les entités de la BC sont indexées par un modèle standard de recherche d'information [8]. On peut alors sélectionner des entités candidates sur la base de leur similarité avec la mention visée, selon un modèle de pondération tf-idf classique.

4.2 Sélection de la meilleure entité candidate

L'objectif de cette étape est de trouver l'entité correcte parmi l'ensemble des entités candidates. Nous posons ce problème comme une classification binaire supervisée et entraînons un classifieur pour sélectionner l'entité résultat. Chaque entité candidate est représentée par un groupe de traits représentatifs de la stratégie utilisée pour la génération de l'entité candidate. Ces traits ont des valeurs binaires pour les stratégies simples de mise en correspondance (1 si une égalité est trouvée, 0 sinon) et une valeur réelle correspondant au score de similarité pour le modèle de recherche d'information et différents scores de similarité contextuelle, globales et locales.

4.2.1 Scores de similarité textuelle globale

Deux scores mesurent la similarité entre le texte contenant la mention d'entité à désambiguïser et une description textuelle des entités de la BC. Pour une mention q et une entité candidate e de la base de connaissance, nous calculons trois vecteurs représentant trois contextes : le document dans lequel q apparaît, noté $d(q)$, la page Wikipédia associée à e, noté $w(e)$ et l'ensemble des entités qui sont en relation avec e dans la base de connaissances, noté $r(e)$. Chaque contexte est représenté par un vecteur, dans le même espace, selon un modèle standard de représentation vectorielle, utilisant une pondération *tf-idf*, $d(q) = (d_1, \ldots, d_n)$ avec $d_i = tf(t_i, d) \times idf(t_i)$, où $tf(t_i, d)$ est une fonction de la fréquence du terme t_i dans le document d et $idf(t_i)$ est une fonction inverse de la fréquence en documents de t_i dans la collection. L'espace vectoriel commun dans lequel ces contextes sont représentés est construit à partir de l'intégralité des pages Wikipédia (les scores *idf* sont donc calculés sur cette collection de documents). Les scores de similarité sont produits par la fonction cosinus entre le vecteur de contexte de la mention et les vecteurs de contexte des entités de la base de connaissances :

$$sim_d(q, e) = cos(d(q), w(e)) = \frac{\sum_i d_i.w_i}{||d(q)||.||w(e)||}$$

8. Nous utilisons le moteur de recherche Lucene.

$$sim_r(q,e) = cos(d(q), r(e)) = \frac{\sum_i d_i.r_i}{||d(q)||.||r(e)||}$$

4.2.2 Scores de similarité contextuelle locale fondés sur les *embeddings* EAT

Trois scores de similarité fondés sur les *embeddings* calculés par le modèle EAT permettent de prendre en compte un contexte plus restreint. Comme ce contexte contient par construction peu de mots, l'utilisation d'*embeddings* permet de pallier le problème d'une représentation lexicale pauvre.

Le paragraphe $p(q)$ où apparaît la mention q est extrait du document $d(q)$. En utilisant les positions des entités, fournies en entrée du système, nous extrayons trois phrases de contexte pour chaque mention : les phrases précédente, courante et suivante. Trois mesures de similarité sont alors calculées sur ce contexte : la valeur moyenne des similarités cosinus entre chaque mot de ce contexte et le vecteur de l'entité candidate cible (EAT_1), la similarité cosinus entre le vecteur moyen des mots et le vecteur de l'entité candidate cible (EAT_2) et, enfin, la moyenne des k meilleures similarités (EAT_3). Les équations pour ces trois similarités sont données ci-dessous :

$$EAT_1(e, p(q)) = \frac{\sum_{w_i \in p(q)} cos(\boldsymbol{e}, \boldsymbol{w}_i)}{||p(q)||} \quad EAT_2(e, p(q)) = cos(\boldsymbol{e}, \frac{\sum_{w_i \in p(q)} \boldsymbol{w}_i}{||p(q)||})$$

$$EAT_3(e, p(q)) = \frac{\sum argmax_{w_i \in p(q) | i=1,..,k} cos(\boldsymbol{e}, \boldsymbol{w}_i)}{k}$$

où $argmax_{w_i} cos(\boldsymbol{e}, \boldsymbol{w}_i)$ est le i-ème mot le plus similaire, en terme de similarité cosinus, à $p(q)$.

4.2.3 Apprentissage supervisé pour la sélection du meilleur candidat

À partir des données d'entraînement, nous générons les entités candidates. Les exemples positifs pour l'apprentissage sont formés par les couples (mention d'entité, entité candidate) qui correspondent à un lien effectif de la référence ; les exemples négatifs sont les couples dont l'entité candidate n'est pas dans la référence. Comme le nombre de candidats générés pour chaque mention peut être très élevé (entre 1 et 460 dans nos expériences), les classes positives et négatives sont très déséquilibrées. Pour résoudre ce problème, nous effectuons un sous-échantillonnage des données en limitant le nombre d'exemples négatifs à 10 fois le nombre d'exemples positifs [9]. Chaque décision du classifieur est associée à une probabilité : l'entité candidate qui a la plus grande probabilité d'acceptation par le classifieur est sélectionnée comme entité désambiguïsée. Dans la tâche standard de désambiguïsation d'entités, il faut aussi déterminer quand une mention d'entité ne fait référence à aucune entité de la base de connaissances (entités *NIL*). Dans notre approche, les entités NIL sont celles pour lesquelles aucun candidat n'est généré ou pour lesquelles tous les candidats générés sont rejetés par le classifieur.

Nous avons testé plusieurs classifieurs pour cette tâche. En raison de la nature particulière du vecteur de traits (peu de dimensions, qui peuvent avoir des valeurs binaires ou réelles), nous avons considéré des modèles tels que Adaboost, Arbres de Décision, Forêts Aléatoires mais aussi des modèles SVM, avec des noyaux linéaires ou RBF. Les meilleurs résultats ont été obtenus, de façon comparable, avec Adaboost et des SVM linéaires. Les résultats présentés par la suite sont ceux obtenus avec Adaboost.

9. Ce nombre résulte du test de plusieurs valeurs sur les données d'entraînement.

5 Évaluation

Pour appliquer le modèle EAT, une collection de documents dans laquelle les entités sont liées au texte est nécessaire. Nous utilisons l'encyclopédie en ligne Wikipédia [10] : chaque page Wikipédia est considérée comme une entité. Les textes des liens ont été légèrement modifiés pour indiquer la partie qui correspond à un mot et la partie qui correspond à une entité. Nous avons également mis en correspondance les entités de Wikipédia avec les entités de la base de connaissances cible (Freebase). Nous présentons dans la section suivante les résultats de l'implémentation de notre modèle, en utilisant *Gensim* et *Hyperwords*.

5.1 Évaluation des *embeddings*

Pour évaluer la qualité des *embeddings* appris, nous les testons sur des données de référence pour une tâche d'analogie standard (Mikolov *et al.*, 2013), qui se définit de la façon suivante : étant donnés deux mots liés par une relation (par exemple, *Paris – France*), le but est de prédire la valeur manquante d'une autre paire de mots liés par la même relation (par exemple, pour la paire *Rome – ? ? ?*, il faut trouver *Italie*). La collection de référence utilisée est constituée de paires de mots liés par des relations syntaxiques et sémantiques. Nous n'utilisons dans cette évaluation que les relations sémantiques. Comme notre objectif est d'évaluer la qualité des vecteurs obtenus pour les mots et les entités, chaque exemple est associé à l'entité correspondante pour 4 des 5 types de relations sémantiques.

Des premières expériences ont été réalisées en utilisant l'outil *Hyperwords* pour implémenter le modèle EAT. Nous avons d'abord utilisé la configuration suggérée par les auteurs (Levy *et al.*, 2015) (configuration *skip-gram*, negative_sampling$= 5$, fenêtre$= 2$, vecteurs=mots+context). Les résultats obtenus sont comparables aux valeurs rapportées par Levy *et al.* (2015) et sont supérieures à celles obtenues par Mikolov *et al.* (2013). Notre modèle qui représente simultanément les mots et les entités obtient des résultats un peu inférieurs à ceux obtenus avec l'implémentation *Hyperwords* originale. La différence va de 1,2 % lorsque la fonction d'addition est utilisée ($61,9$ %) jusqu'à 2,8 % avec l'utilisation de la fonction de multiplication ($67,6$ %) [11]. La différence entre le modèle basique et la version EAT est due aux points additionnels qui sont représentés dans l'espace. Une situation similaire a été observée durant nos expériences : des performances plus faibles sont obtenues lorsque la taille du vocabulaire est augmentée par la baisse du seuil de fréquence pour la prise en compte des mots. Comme mentionné par Levy *et al.* (2015), trouver les bonnes valeurs des paramètres pour une tâche donnée est un élément clé pour obtenir une bonne qualité du traitement. En effet, lorsqu'une valeur de seuil de fréquence élevée est utilisée, les résultats obtenus sont compétitifs, comparés à l'état de l'art sur cette tâche d'analogie, mais beaucoup d'entités sont manquantes. Le nombre élevé d'entités filtrées par le seuil de fréquence a un impact important sur la qualité du modèle. D'un autre côté, lorsque le seuil de fréquence est plus bas, un plus grand nombre d'entités est pris en compte mais les résultats sur la tâche d'analogie sont moins bons (notre implémentation EAT-*Gensim* a les paramètres suivants : seuil de fréquence de 30 pour les mots et 0 pour les entités).

Les résultats avec notre modèle pour chaque type de relation sémantique sont présentés dans la table 1, en utilisant l'implémentation EAT-*hyperwords* et EAT-*gensim*. La colonne *mots* contient les résultats obtenus en utilisant seulement les mots et la colonne *entités* contient les résultats pour

10. Nous nous appuyons ici sur une image de la collection Wikipédia récupérée en juin 2016.
11. Plus de détails sur les fonctions d'addition et de multiplication peuvent être trouvés dans Levy *et al.* (2015).

	EAT-*hyperwords*			EAT-*Gensim*	
Type de relation	**mots**	**entités**	**entité→mot**	**mots**	**entités**
capital-common-countries	95,7 %	63,0 %	87,5 %	75,69 %	77,50 %
capital-world	77,0 %	37,3 %	81,3 %	49,68 %	80,00 %
currency	8,2 %	0,0 %	5,2 %	0,00 %	0,00 %
city-in-state	72,3 %	25,8 %	62,6 %	31,67 %	89,83 %

TABLE 1 – Résultats (*accuracy*) selon le type de relation sémantique, sur les données d'analogie, pour les mots et les entités, en faisant varier le modèle EAT et le seuil de fréquence.

	TAC 2015 entraînement	TAC 2015 test
Nb. docs	168	167
Nb. mentions	12 175	13 587
Nb. mentions NIL	3 215	3 379

TABLE 2 – Description des données d'évaluation pour la désambiguïsation d'entités.

les noms d'entités correspondant. Enfin, nous présentons dans la colonne *entité→mot* les résultats obtenus en combinant les mots et les entités : dans ce cas, nous avons remplacé les entités absentes du vocabulaire par leur mot correspondant. Ces derniers résultats montrent clairement une amélioration par rapport aux entités seules, et approchent les résultats obtenus avec les mots seuls, en les dépassant même pour le type *capital-world*.

Pour le modèle EAT-*Gensim*, les résultats pour les entités sont meilleurs que pour les mots sur les types *capital-world* and *city-in-state*, mais n'apportent pas d'amélioration notable pour les types *capital-common-countries* et *currency* [12]. Même si les résultats globaux pour les mots seuls sont moins bons avec l'implémentation EAT-*Gensim* $(40,31\ \%)$ comparés à ceux obtenus avec l'implémentation EAT-*hyperwords* $(61,9\ \%)$, nous préférons la première parce qu'elle a une meilleure couverture en termes d'entités prises en compte.

Pour notre tâche de désambiguïsation d'entités nommées, nous nous appuyons sur l'implémentation *Gensim* pour avoir le maximum de mots et d'entités représentés dans le même espace joint.

5.2 Corpus et mesures d'évaluation pour la désambiguïsation d'entités

Pour évaluer notre approche pour la désambiguïsation d'entités, nous utilisons les données de la tâche EDL (*Entity Discovery and Linking*) de la campagne d'évaluation TAC 2015. Nous effectuons cette évaluation sur la tâche diagnostique pour l'anglais, où les mentions d'entités sont données en entrée. La table 2 présente les caractéristiques de ces données : nombre de documents, nombre d'entités à trouver (l'objectif de cette tâche est de faire la liaison de la totalité des entités nommées trouvées dans les documents considérés) et le nombre d'entités qui n'ont pas de correspondance dans la base de connaissances (mentions NIL). La base de connaissances utilisée dans la campagne d'évaluation est construite à partir de Freebase (Ji *et al.*, 2015). L'image considérée contient plus de 43 millions d'entités, mais un premier filtre officiel est appliqué pour supprimer certaines entités

12. Dans les deux cas, les résultats pour le type *family* ne sont pas calculés en raison du nombre d'entités manquantes dans Wikipédia pour cette catégorie.

qui ne sont pas pertinentes pour la campagne (comme les œuvres de fictions, livres ou films ou les entités médicales), ce qui réduit la base à 8 millions d'entités. Parmi celles-ci, seules 3 712 852 (46 %) ont un contenu associé dans Wikipédia et peuvent donc être associées à un contexte textuel et à un *embedding*. Freebase contient, pour chaque entité, plusieurs formes et variations de mentions, mais ces variations sont relativement incomplètes. Pour améliorer la génération de candidats, nous avons procédé à un enrichissement de la base de connaissances avec de nouvelles expressions des entités, extraites automatiquement de Wikipédia : nous avons ajouté les textes des liens à partir des pages de désambiguïsation et des pages de redirections comme variations possibles des entités.

En ce qui concerne l'évaluation, nous utilisons les mesures standard de précision/rappel/f-score sur l'identification correcte de l'entité de la base de connaissance lorsqu'elle existe (*link*), l'identification correcte d'une mention NIL (*nil*) ou le score combiné sur les deux (*all*). Par rapport aux mesures standard de la campagne d'évaluation, nous ne prenons pas en compte le type de l'entité pour son identification, parce que nous voulons évaluer la seule désambiguïsation des entités et pas la performance du typage des entités (qui ne dépend que du système de reconnaissance des entités pour les mentions NIL) et nous ne nous intéressons pas au regroupement des mentions NIL qui se réfèrent à la même entité. Les mesures utilisées sont les mesures officielles *strong_link_match*, *strong_nil_match* et *strong_all_match*. Ces mesures sont définies par les équations suivantes, où e est une mention d'entité, e_r l'entité de la base de connaissances associée à e dans la référence, e_t l'entité associée à e par notre système et $N(x)$ le nombre de mentions d'entité qui vérifient x :

$$P(nil) = \frac{N(e_t = \text{NIL} \wedge e_r = \text{NIL})}{N(e_t = \text{NIL})} \quad R(nil) = \frac{N(e_t = \text{NIL} \wedge e_r = \text{NIL})}{N(e_r = \text{NIL})}$$

$$P(link) = \frac{N(e_t = e_r \wedge e_t \neq \text{NIL})}{N(e_t \neq \text{NIL})} \quad R(link) = \frac{N(e_t = e_r \wedge e_t \neq \text{NIL})}{N(e_r \neq \text{NIL})}$$

$$P(all) = \frac{N(e_t = e_r)}{N(e_t)}$$

Notons que, pour la mesure *all*, la précision, le rappel et le f-score sont égaux, dès lors que le système a fourni une réponse pour toutes les mentions d'entités demandées ($N(e_t) = N(e_r)$).

5.3 Évaluation de la génération des candidats

Le nombre de candidats engendrés a un impact important sur le résultat du processus complet. La table 3 présente des résultats sur la génération des candidats : on note C l'ensemble des candidats, C_{NIL} l'ensemble des mentions d'entités pour lesquelles aucun candidat n'est proposé, C_{AVG} le nombre moyen de candidats par mention d'entités et Rappel(C) le rappel sur les candidats, défini comme le pourcentage des mentions non-NIL pour lesquelles l'entité attendue est présente parmi les entités candidates. On présente également dans ce tableau les scores de désambiguïsation obtenue avec différents ensembles de candidats, en utilisant le modèle de base (sans les *embeddings* EAT).

En appliquant toutes les stratégies pour la génération des candidats, nous obtenons un rappel sur les candidats très élevé, avec 95 % des mentions non-NIL pour lesquelles la bonne entité est parmi les candidats. On remarque que ces stratégies génèrent un grand nombre de candidats par mention. Lorsqu'on applique un filtre par le type des entités et qu'on ne garde que les entités auxquelles on peut attribuer un des types attendus (pour la campagne TAC 2105, ces types sont Personne, Lieu,

Tous	$\|C\|$	$\|C_{NIL}\|$	C_{AVG}	Rappel(C)	P(all)
train	6 843 513	781	562,1	95,60 %	
test	8 339 648	499	613,8	94,19 %	0,646

Typage	$\|C\|$	$\|C_{NIL}\|$	C_{AVG}	Rappel(C)	P(all)
train	3 179 795	952	261,2	92,43 %	
test	3 810 382	626	280,4	90,36 %	0,680

Typage-sim	$\|C\|$	$\|C_{NIL}\|$	C_{AVG}	Rappel(C)	P(all)
train	1 723 470	952	141,6	90,27 %	
test	1 921 577	625	141,4	87,95 %	0,714

TABLE 3 – Résultats de la génération des candidats.

Organisation, Bâtiment et Entité géo-politique), on réduit de plus de la moitié le nombre des entités candidates, ce qui donne une meilleure base pour entraîner le classifieur : même si le rappel est diminué (autour de 90 %), le score de désambiguïsation est amélioré. Une analyse des candidats a également montré que les candidats générés seulement par Lucene et les candidats pour lesquels les scores de similarité contextuelle globale sont tous les deux nuls produisent beaucoup de bruit. Le filtrage de ces candidats avant d'entraîner le classifieur permet à nouveau d'améliorer le résultat de désambiguïsation, avec un score global de 71,4 %, pour un rappel sur les candidats de l'ordre de 88 %. Cette dernière stratégie est celle utilisée dans les résultats présentés dans la section suivante.

5.4 Résultats sur la désambiguïsation d'entités nommées

Nous présentons à la table 4 les résultats obtenus pour la tâche globale de liaison d'entités. Le résultat *baseline* est obtenu en utilisant les traits de la génération des candidats et les scores de similarité contextuelle globale. Les autres résultats sont obtenus en ajoutant chacun des scores calculés avec les *embeddings* du modèle EAT. La dernière colonne montre les résultats en utilisant les scores combinés. Puisque notre modèle de désambiguïsation s'appuie sur des méthodes qui ont des composantes aléatoires (la sélection des exemples négatifs pour le sous-échantillonnage et l'échantillonnage interne du modèle Adaboost), les résultats présentés sont des moyennes sur 10 essais.

Ces résultats montrent une amélioration significative des scores en utilisant les *embeddings* du modèle EAT, par rapport à la baseline. On remarque aussi que, même si les traits combinés donnent les meilleurs résultats, les scores sont proches de ceux obtenus avec l'approche EAT_3 seule, ce qui montre l'importance de ce trait particulier. Si l'on compare les résultats obtenus par notre système avec ceux des participants à la campagne d'évaluation TAC-EDL 2015 (Ji *et al.*, 2015), le meilleur F-score obtenu pour la tâche diagnostique est de 0,737 sur la mesure *strong_typed_all_match* : nous obtenons donc des résultats très proches de l'état de l'art. On note que les résultats ne sont pas directement comparables puisque cette mesure officielle prend en compte le typage des entités, alors que nous ne les calculons pas, mais nous ne disposons pas des scores pour notre tâche.

Une étude des résultats permet de mettre en évidence des apports du modèle EAT pour caractériser le

	baseline	base+EAT_1	base+EAT_2	base+EAT_3	base+$EAT_{1/2/3}$
P(nil)	0,648	0,651	0,652	0,646	**0,657**
R(nil)	0,827	0,826	0,825	0,836	**0,837**
F(nil)	0,727	0,728	0,728	0,729	**0,736**
P(link)	0,745	0,758	0,762	0,766	**0,766**
R(link)	0,676	0,691	0,695	0,692	**0,696**
F(link)	0,709	0,723	0,727	0,727	**0,729**
P(all)	0,714	0,724	0,727	0,728	**0,731**

TABLE 4 – Résultats de la désambiguïsation des entités nommées avec le modèle EAT.

contexte proche des entités. Par exemple, dans les cas ambigus où une personne est évoquée seulement avec son nom de famille, notre modèle est meilleur pour trouver la bonne entité. Par exemple, dans la phrase "*As soon as he landed at the Bangkok airport, Koirala saw Modi's tweets on the quake, Nepal's Minister for Foreign Affairs Mahendra Bahadur Pandey said on Tuesday.*", la mention *Modi* est correctement identifiée comme *Navendra Modi* au lieu de *Modi Naturals*, une compagnie indienne de traitement du pétrole. Des améliorations similaires sont observées pour le type Lieu : par exemple le modèle EAT permet d'identifier correctement *Montrouge* comme la ville de la banlieue parisienne au lieu de l'acteur *Louis (Émile) Hesnard* dont le surnom était *Montrouge*, dans le contexte "*The other loose guy who killed a cop in montrouge seems to have done the same. And there are report of two other armed men running around in Paris. It's kind of a chaos here.*".

6 Conclusion

Dans cet article, nous avons présenté un modèle capable d'apprendre de façon jointe des représentations vectorielles (plongements lexicaux ou *embeddings*) de mots et d'entités nommées dans un même espace vectoriel, le modèle EAT. Ce modèle, fondé sur les textes d'ancrage (*anchor texts*) permet de représenter efficacement les entités dans le même espace que les mots et même souvent de façon très pertinente parce que les entités (des concepts de Wikipédia) sont souvent utilisées dans des contextes qui sont particulièrement représentatifs de leur sens. L'idée de s'appuyer sur les ancrages permet d'utiliser le contexte direct pour la construction des modèles d'entités, en évitant la tâche d'alignement souvent difficile, qui est nécessaire si les *embeddings* sont appris séparément.

Nous montrons que le modèle EAT peut être intégré sans difficulté dans une architecture standard de désambiguïsation d'entités nommées et permet d'en améliorer les résultats. Trois mesures fondées sur le modèle EAT ont été définies pour évaluer la similarité entre le contexte d'une mention d'entité et les entités candidates de la base de connaissance. L'évaluation de l'apport de ces mesures sur une collection de référence récente, issue de la campagne d'évaluation TAC-EDL 2015 montre que ces mesures permettent d'améliorer les performances de désambiguïsation, en passant de 0,71 à 0,73 de F-score, et permet d'avoir un score comparable aux meilleurs des participants de la campagne.

Remerciements

Ce travail a été partiellement financé par le projet PULSAR-FUI-18 (PUrchasing Low Signals and Adaptive Recommendation).

Références

BARONI M., DINU G. & KRUSZEWSKI G. (2014). Don't count, predict ! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the ACL*, p. 238–247.

BORDES A., USUNIER N., GARCIA-DURAN A., WESTON J. & YAKHNENKO O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, p. 2787–2795.

CAO Z., TAO Q., TIE-YAN L., MING-FENG T. & HANG L. (2007). Learning to Rank : From Pairwise Approach to Listwise Approach. In *24th International Conference on Machine Learning (ICML 2007)*, p. 129–136, Corvalis, Oregon, USA.

CASSIDY T., CHEN Z., ARTILES J., JI H., DENG H., RATINOV L.-A., ZHENG J., HAN J. & ROTH D. (2011). CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In *Text Analysis Conference (TAC 2011)*.

CUCERZAN S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *2007 Joint Conference on EMNLP-CoNLL*, p. 708–716.

FANG W., ZHANG J., WANG D., CHEN Z. & LI M. (2016). Entity disambiguation by knowledge and text jointly embedding. *CoNLL 2016*, p. 260.

HAN X. & ZHAO J. (2009). NLPR_KBP in TAC 2009 KBP Track : A Two-Stage Method to Entity Linking. In *Text Analysis Conference (TAC 2009)*.

HOFFART J., SUCHANEK F. M., BERBERICH K. & WEIKUM G. (2013). Yago2 : A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, **194**, 28–61.

JI H., NOTHMAN J. & HACHEY B. (2014). Overview of TAC-KBP2014 Entity Discovery and Linking Tasks. In *Text Analysis Conference (TAC 2014)*.

JI H., NOTHMAN J., HACHEY B. & FLORIAN R. (2015). Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In *Text Analysis Conference (TAC 2015)*.

LE Q. V. & MIKOLOV T. (2014). Distributed representations of sentences and documents. *Proceedings of The 31st ICML*, p. 1188–1196.

LEHMANN J., MONAHAN S., NEZDA L., JUNG A. & SHI Y. (2010). LCC Approaches to Knowledge Base Population at TAC 2010. In *Text Analysis Conference*.

LEVY O., GOLDBERG Y. & DAGAN I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, **3**, 211–225.

LING X., SINGH S. & WELD D. (2015). Design Challenges for Entity Linking. *Transactions of the Association for Computational Linguistics (TACL)*, **3**, 315–328.

MIHALCEA R. & CSOMAI A. (2007). Wikify ! Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, p. 233–242, Lisbon, Portugal : ACM.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, p. 3111–3119.

MILLER G. A. (1995). Wordnet : a lexical database for english. *Communications of the ACM*, **38**(11), 39–41.

MINTZ M., BILLS S., SNOW R. & JURAFSKY D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, ACL '09, p. 1003–1011.

MORO A., RAGANATO A. & NAVIGLI R. (2014). Entity linking meets word sense disambiguation : a unified approach. *Transactions of the Association for Computational Linguistics (TACL)*, **2**, 231–244.

NAVIGLI R. & PONZETTO S. P. (2012). Babelnet : The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, **193**, 217–250.

PAPPU A., BLANCO R., MEHDAD Y., STENT A. & THADANI K. (2017). Lightweight multilingual entity extraction and linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM 17 : ACM.

SHEN W., JIANYONG W., PING L. & MIN W. (2012). LINDEN : Linking Named Entities with Knowledge Base via Semantic Knowledge. In *21st International Conference on World Wide Web (WWW'12)*, p. 449–458, Lyon, France.

SHEN W., WANG J. & HAN J. (2015). Entity linking with a knowledge base : Issues, techniques, and solutions. *Transactions on Knowledge and Data Engineering*.

VARMA V., BHARATH V., KOVELAMUDI S., BYSANI P., GSK S., N K. K., REDDY K., KUMAR K. & MAGANTI N. (2009). IIT Hyderabad at TAC 2009. In *Text Analysis Conference (TAC 2009)*.

WANG Z., ZHANG J., FENG J. & CHEN Z. (2014). Knowledge graph and text jointly embedding. In *The 2014 Conference on Empirical Methods on Natural Language Processing*.

YAMADA I., SHINDO H., TAKEDA H. & TAKEFUJI Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. *Proceedings of the 20th SIGNLL CoNLL*, p. 250–259.

ZWICKLBAUER S., SEIFERT C. & GRANITZER M. (2016). Doser - A knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In *13th International Conference, ESWC 2016*, p. 182–198.