Available online at https://www.asjp.cerist.dz/en/PresentationRevue/134

RIST

Information Processing at the Digital Age Journal

Induction de sens des mots Arabes dans un espace vectoriel des mots.

Djaidri Asma[a], *Aliane Hassina*[b], *Azzoune Hamid*[a]*

[a] *University of Sciences and Technologie Houari Boumediene (USTHB), Algiers, Algeria*
[B] Research Center for scientific and technical information (CERIST), Algiers, Algeria

Résumé

Nous décrivons dans cet article, une nouvelle approche d'induction de sens des mots pour la langue Arabe dans un espace vectoriel des mots. Les modèles de représentation vectorielles suscitent un grand intérêt de la part de la communauté de recherche TALN. Ces modèles sont fondés sur l'hypothèse distributionnelle qui prend en compte le « contexte » d'un mot cible. Ces modèles mappent tous les mots du vocabulaire à un espace vectoriel et fournissent ensuite une description sémantique des mots d'un corpus en tant que vecteurs numériques. Néanmoins, un problème bien connu de ces modèles est qu'ils ne peuvent pas gérer la polysémie. Nous présentons un nouveau modèle simple qui utilise les word embeddings que nous expérimentons pour la tâche non supervisée de l'induction de sens des mots arabes. Les modèles sont développés à l'aide des outils GenSim pour SKIP-Gram et CBOW. Le modèle permet ensuite de créer un indexeur basé sur la similarité cosinus en utilisant l'indexeur Annoy, qui est plus rapide que la fonction de similarité de GenSim. Un ego-network est utilisé pour étudier la structure des relations d'un individu et permet de construire un graphe de mots associés provenant des voisins locaux. Les différents sens des mots sont générés en utilisant du clustering de graphes. Nous avons travaillé avec deux corpus d'information: OSAC et AraCorpus ainsi qu'un modèle de Word Embeddings existant AraVec. Ensuite, nous avons expérimenté les différents modèles pour l'induction du sens des mots et nous avons obtenu des résultats prometteurs.

Keywords: Représentation vectorielle de mots ; Word2Vec; Induction de sens; langue Arabe; TALN.

1. Introduction

Le sens d'un mot est une représentation discrète d'un aspect significatif du mot, ce qui implique que les

**E-mail address:* adjaidri@usthb.dz , haliane@hotmail.fr, azzoune@yahoo.fr

sens d'un mot sont l'ensemble des significations possibles d'un mot que l'on peut trouver dans les dictionnaires, corpus, dictionnaires électroniques…etc. (Jufrasky and Martin, 2016). Le choix de la représentation des sens des mots est un problème fondamental en TALN et dépend du type de l'application. Un inventaire de sens peut être construit de différentes manières: il s'agit généralement d'une liste fixe des sens de chaque mot (Kwong, 2013; Kozlowski and Rybinski, 2017). La construction manuelle de ressources lexicales ou de données annotées manuellement est coûteuse et prend du temps. L'induction de sens des mots (WSI pour word sense induction en anglais) permet de résoudre ce problème en utilisant des algorithmes de clustering qui n'ont pas besoin de données d'apprentissage (Pinto et al., 2017). WSI est un problème ouvert en TALN, lié à la désambiguïsation lexicale des mots (Word Sense Disambiguation -WSD) et qui vise à induire automatiquement des sens de mots d'un corpus. La taille du corpus a un impact important sur le WSI. Cependant, le clustering dans un texte de grande dimension est un problème difficile.

Le plongement des mots littéralement ou la représentation vectorielle d'un mot techniquement (Word Embeddings en *anglais*) est une méthode efficace pour représenter des mots dans une dimension réduite. Un vecteur à une-dimension est utilisé pour représenter les mots (Mikolov et al., 2013). Ces modèles permettent aux mots ayant une signification similaire d'avoir une représentation similaire. Cependant, ces représentations utilisant un seul vecteur sont incapables de capturer les différents sens du mot. Afin de bénéficier de la technique de la représentation vectorielle de mots pour des sens de mots individuels, plusieurs approches ont été proposées(Nguyen et al., 2017; Iacobacci et al ., 2016 ; Pelevina et al., 2016 ; Cocos et al., 2017 ; Bartanov et al., 2016 ; Alexender, 2016). .

La contribution de cet article est une technique qui produit automatiquement un inventaire des sens de mots arabes en utilisant l'induction du sens des mots via les Word Embeddings, où les sens des mots de l'inventaire sont représentés par des groupes de mots. À notre connaissance, il s'agit de la première tentative de création automatique d'un inventaire de sens arabe en utilisant la représentation vectorielle des mots. Les expérimentations montrent que notre approche est prometteuse et démontre une bonne performance de l'induction de sens des mots pour un échantillon de mots arabes ambigus.

2. Les Modèles de la Représentation Vectorielle des Mots

La représentation vectorielle des mots est l'une des dernières solutions proposées pour de nombreuses applications de TALN et qui a eu un grand succès. Elle a été proposée pour la première fois en 2003 par (Bengio et al., 2003) et est devenue populaire avec le modèle Word2Vec en 2013 (Mikolov et al., 2013). Ces modèles « plongent » des mots dans des vecteurs à valeur réelle dans un espace sémantique de dimension inférieure qui peut être appris par des algorithmes d'apprentissage automatique pour prédire des mots et non pour compter des mots. L'avantage principal de ces modèles, outre leur faible dimensionnalité est qu'ils peuvent capturer l'information de similarité des mots: des mots similaires ont des vecteurs similaires. Cependant, ces modèles ne prennent pas en compte les ambiguïtés lexicales, ils représentent tous les sens d'un mot par une représentation vectorielle unique (Nguyen et al, 2017). Afin de pouvoir bénéficier des techniques de la représentation vectorielle des mots pour trouver les sens des mots individuels, nous induisons automatiquement les différents sens des mots arabes et construisons des inventaires pouvant être utilisés ultérieurement pour des applications telles que la désambiguïsation sémantique (WSD).

2.1. Ressources de données

L'objectif principal de ce travail est de construire un embeddings modèle de mots arabes pour la discrimination de sens des mots. À cette fin, nous avons construit deux modèles Word2Vec, Skip-gram et

CBOW pour chacun des deux corpus: le corpus arabe Open Source (OSAC) et le corpus standard arabe moderne nommé: AraCorpus. Nous avons ensuite réalisé WSI avec nos modèles obtenus et un modèle AraVec existant.

2.1.1 Le corpus arabe open source (OSAC)

C'est un corpus construit à partir de plusieurs sites Web. Il est divisé en trois groupes principaux: BBC-Arabe Corpus qui contient 1860786 (1.8M) mots et 106733 mots uniques après suppression des mots d'arrêt, CNN-Arabic Corpus qui contient 2241348 (2.2M) mots et 144460 mots uniques après la suppression des mots d'arrêt. Ensuite, OSAC collecté à partir de plusieurs sites Web présentés dans (Motaz and Wesam, 2010) qui contient environ 18 183 511 (18M) mots et 449 600 mots uniques après suppression de mots vides (Motaz and Wesam, 2010 ; Ibrahim, 2016). Nous n'avons pas utilisé le corpus CNN-Arabic en raison de problèmes de codification dans le corpus.

2.1.2 Le corpus arabe moderne standard (AraCorpus).

C'est une collection d'articles de journaux arabes provenant de dix pays arabes. Il compte 102 134 articles, avec 113 millions de mots (800 Mo) et 296570 mots uniques (Abdelali et al., 2005 ; Ibrahim, 2016).

2.1.3 AraVec

Il s'agit d'un projet open source pré-entrainé de représentation vectorielle de mots, il est gratuit et offre de puissants embeddings modèles**.** AraVec propose six modèles différents construits à partir de trois corpus arabes différents: Twitter, Wikipédia et des pages Web. Le corpus Twitter comprend 1090 millions de mots et 164077 mots uniques. Le corpus de Wikipédia contient 78,9 millions de mots et 140319 mots uniques et le corpus WWW compte 2225,3 millions de mots avec 146273 mots uniques (Soliman et al., 2017).

2.2. le Pré-traitement

Pour créer un modèle Word2Vec, une étape de prétraitement est requise. Nous utilisons l'outil GenSim (Rehurek and Sojka, 2010), qui attend une séquence de phrases en entrée où chaque phrase contient une liste de mots et chaque ligne du fichier est une phrase.

AraCorpus est prêt à être utilisé pour construire un modèle Word2Vec avec GenSim, il suffit de supprimer certains caractères spéciaux mais le corpus OSAC nécessite un prétraitement supplémentaire tel que la normalisation et la suppression: (Soliman et al., 2017)

- des lettres non arabes comme BBC Arabic ou CNN Arabic au début de chaque fichier du corpus,
- des caractères spéciaux attachés aux mots comme "بحسم,
- les chiffres,
- la vocalisation comme : اطلاعاً
- l'allongement des lettres.

2.3. L'Apprentissage d'un Model Word2Vec

Après la préparation du corpus, nous avons construit les modèles CBOW et Skip-gram en utilisant le toolkit GenSim pour OSAC et AraCorpus. Les modèles AraVec (Soliman et al., 2017) ont également été

construits à l'aide de l'outil GenSim, ce qui nous permet de faire une comparaison raisonnable entre les modèles obtenus avec OSAC, AraCorpus et les modèles AraVec.

Le choix des paramètres d'entraînement est une étape importante ici. Nous avons sélectionné un ensemble de paramètres en fonction des évaluations antérieures des expériences présentées dans (Pelevina et al., 2016) et des modèles AraVec (Soliman et al., 2017). Nous avons modélisé des embedding modèles de OSAC et AraCorpus avec une dimension de 300, une taille de fenêtre contextuelle de 5 et une fréquence minimale de 5. La table 1 montre la configuration utilisée pour construire nos modèles pour OSAC et AraCorpus et la configuration utilisée par les créateurs d'AraVec.

Table 1. Configuration de modèles word embeddings

Model Name	#unique word	Min Word Count	Window size	technique	Time
OSAC-CBOW OSAC-SG	140658	5	5	CBOW SG	751.5s 660.5s
AraCorpus-CBOW AraCorpus-SG	296570	5	5	CBOW SG	6340.7s 5395.8s
Twitter-CBOW Twitter-SG	164077	500	3	CBOW SG	1.5 days
WWW-CBOW WWW-SG	146273	500	5	CBOW SG	4 days
Wiki-CBOW Wiki-SG	140319	20	5	CBOW SG	10 Hours

3. Induction du Sens Arabe en Utilisant les Modèles Word2Vec

Nous induisons l'inventaire des sens arabes en regroupant le graphe de similarité des mots de manière similaire à (Pelevina et al., 2016 ; Biemann, 2006 ; Biemann, 2012) où un sens de mot est représenté par un groupe de mots. Par exemple, le mot «ذكر » avec le sens « mentionner ذكر » peut être représenté par le cluster : أورد، حكى، ذكر، اورد، ذم، واورد، اقوالا.

Pour induire des sens, nous construisons simplement un indexeur à l'aide de l'outil « Annoy » pour chaque embedding modèle, ensuite nous l'utilisons comme graphe de similarité. Enfin nous générons un ego-Network pour chaque mot du vocabulaire du modèle sur lequel nous avons effectué un algorithme de clustering pour l'ego-Network (Djaidri et al., 2018).

3.1. Construction d'un Graphe de Similarité de Mots

Le graphe de similarité de mots contient tous les mots du vocabulaire en tant que nœuds liés par des arêtes pondérées par la similarité de cosinus entre eux, le graphe n'est pas orienté. Pour construire le graphe, nous devons récupérer pour chaque mot du vocabulaire les k-voisins les plus proches et les présenter dans un fichier constitué d'une ligne de tuples de mots avec leur poids de similarité. Nous utilisons la bibliothèque Annoy pour les requêtes de similarité car l'implémentation actuelle des k-voisins les plus proches dans Word2Vec via GenSim présente une complexité linéaire par force brute dans le nombre de documents indexés. Cependant, l'outil Annoy peut trouver les voisins les plus proches approximatifs beaucoup plus rapidement. Annoy a la capacité d'utiliser des fichiers statiques en tant qu'index et c'est une fonctionnalité importante qui nous aidera plus tard. La similarité entre deux mots, mot_1 et mot_2 est calculée avec la

similitude cosinusoïdale du vecteur de mot_1 et du vecteur de mot_2, la formule est définie comme suit:

$$cos\_sim_{w2v}\,(word_1, word_2) = \frac{word_1 \bullet word_2}{\| word_1 \| \bullet \| word_2 \|} \quad \text{(Erreur ! Signet non défini.)}$$

Où mot_1 et mot_2 sont les vecteurs réels représentant le mot_1 et le mot_2. Le choix du nombre de voisins les plus proches est motivé par des études antérieures (Pelevina et al., 2016 ; Panchenko, 2013).

La Construction d'un Ego-Network: Le graphe de l'ensemble du vocabulaire peut nous renseigner sur la population entière et sa sous-population mais il ne nous en dit pas beaucoup sur les opportunités et les contraintes auxquelles font face les individus (Hanneman et al., 2005). Pour induire des sens pour chaque mot du graphe de similarité des mots, nous devons regarder de plus près chaque mot en tant qu'individu et ses voisins. Ceci est possible avec un Ego-Network où un seul ego représente un mot individu, des alters représentent les voisins du mot et les arcs entre ces altères (Hanneman et al., 2005 ; Panchenko et al.,2017 ; Robert and Riddle, 2005 ; Pelevina et al., 2016). Comme on peut le voir sur la figure 1, dans le réseau du mot « ذكر » l'ego est « ذكر », les alters sont « ذكر، اتقاكم، اكرمكم، وانثى، اقوالا، ذم، وأورد، أورد، حكى », qui sont pondérés avec la distance de similarité cosinus. Nous utilisons les fichiers d'index fournis que nous avons mentionnés dans la section 3.1 sous forme de graphe pour créer l'ego-Network à partir de ces index.

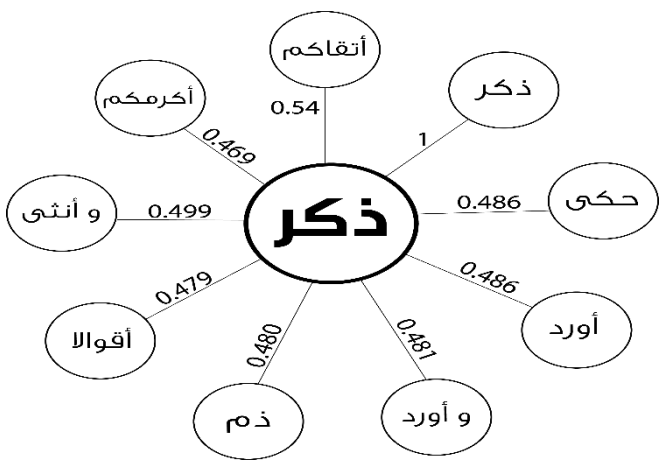

Fig. 1. Ego-Network du mot ذكر avec ses 9 voisins plus proche obtenue du modèle OSAC_CBOW.

3.2. L'Induction du Sens des Mots

Pour discriminer les sens d'un mot donné W, nous clustérisons le graphe des mots connectés en utilisant l'algorithme Chinese Whispers de manière similaire à (Pelevina et al., 2016 ; Biemann, 2006), chaque cluster représentant un sens d'un mot. La table 2 montre une instance des résultats de l'induction pour le mot « ذكر », le mot est induit en deux groupes (c'est-à-dire deux sens) en utilisant le modèle OSAC_CBOW. Le premier groupe {حكي، أورد، وأورد، ذم، اقولا} représente le sens « mentionner/اورد », tandis que le second groupe { اتقاكم، اكرمكم، وانثى} représente le sens « genre/جنس ».

Table 2. Clustering des voisins du mot « ذكر » en deux groupes représentant deux sens différents (genre et mention)

ذكر 1	حكى:0.486	أورد : 0.486	واورد :0.481	ذم :0.480 أقوالا:0.479
ذكر 2	وانثى:0.499	اكرمكم:0.469	اتقاكم:0.454	

La construction du graphe des mots connectés repose sur l'idée de relier deux voisins d'un mot si l'un d'entre eux est l'un des 200 plus proches voisins de l'autre mot. L'algorithme 1 décrit le processus d'induction du sens des mots, où l'entrée $Word2Vec\_modèle$ est l'un des dix embedding modèles entrainés, et $Annoy\_indexer\_de\_w2v$ indexe l'embedding modèle obtenu avec l'Annoy Indexer. Notre algorithme est une variante de l'algorithme WSI décrit dans (Pelevina et al., 2016) où nous utilisons les fichiers de Annoy Indexer comme graphe de similarité de mots, ce qui montre que c'est plus rapide et plus facile.

Algorithme 1. L'induction des sens des mots

```
Entrée: W2v_ modéle , AnnoyIndexeur_de_W2v
Sortie: Le Fichier_ d'inventaire de sens pour les
        mots dans le vocabulaire de w2v_modéle
  Pour chaque mot' dans le vocabulaire :
  G ← Graphe vide pour les mots connectés
  N ← 200 voisins les plus proches de mot'
      récupérer à parir de AnnoyIndexeur_de_w2v
    Pour chaque n∈ N :
     NN ← 200 voisins les plus proches de n
          récupérer à partir de AnnoyIndexeur_de_w2v
    Pour chaque nn ∈ NN :
        Si nn∈ N alors :
            ajouter_lien(nn,n,'poids'=w)

chinese_whispers(G)
```

Nous calculons le poids W en utilisant quatre équations :

$$W = sim(n, nn) \quad (2)$$

$$W = (sim(mot', nn) + sim(n, nn))/2 \quad (3)$$

$$W = (sim(mot', nn) + sim(mot', n) + sim(n, nn))/3 \quad (4)$$

$$W = sim(mot', nn) \quad (5)$$

Le choix de ce paramètre a une grande influence sur les résultats du clustering. La table 3 montre notre évaluation de la granularité des inventaires des sens donnés en utilisant les quatre équations décrites précédemment. On note: « S_T_F » pour décrire « sens très fin », « S_F » pour décrire « sens fin », « S_T_G » pour décrire « sens très grossier » et « S_G_G » pour décrire « sens à gros grain ».

Pour le clustering, nous avons utilisé l'algorithme Chinese Whispers (Biemann, 2006) car il ne nécessite aucun paramètre, nous ne faisons donc aucune hypothèse sur le nombre de sens des mots.

Table 3. La granularité des sens obtenues en appliquant les quatre équations pour les dix modèles

	Eq2.	*Eq3.*	*Eq4.*	*Eq5.*
Osac_CBOW	S_T_F	S_T_F	S_T_F	S_G_G
Osac_SG	S_T_F	S_T_F	S_T_F	S_G_G
Aracorpus_C	S_T_F	S_F	S_F	S_G_G
Aracorpus_S	S_T_F	S_F	S_F	S_G_G
Twr_CBOW	S_T_F	S_T_F	S_T_G	S_G_G
Twr_SG	S_T_F	S_T_F	S_T_G	S_G_G
Wiki_CBOW	S_T_F	S_T_F	S_T_G	S_G_G
Wiki_SG	S_T_F	S_T_F	S_T_G	S_G_G
WWW_CBOW	S_T_F	S_F	S_T_G	S_G_G
WWW_SG	S_T_F	S_F	S_T_G	S_G_G

4. Evaluation

L'évaluation de la tâche d'induction de sens des mots dans la langue arabe est très difficile car pour l'arabe, nous ne connaissons aucune méthode d'évaluation et nous ne pouvons pas calculer la précision et le rappel de l'approche proposée car le fichier Gold-standard de la langue arabe n'est pas encore réalisé. Nous ne pouvons pas non plus comparer avec d'autres travaux car, pour cette tache de WSI, il n'y a qu'un seul travail dédié à la langue arabe (Pinto et al., 2007) et dans ce travail, les auteurs ont basé leur évaluation sur leur propre jugement.

Pour notre travail, nous avons décidé d'appliquer une évaluation non-supervisé, en s'inspirant du travail de (Agirre and Soroa, 2007). Après l'analyse des résultats obtenus, nous remarquons que nos résultats s'évaluent sur deux critères: le premier critère est la réussite des mots vectoriels (word embeddings) dans la tache de WSI, le deuxième critère est la réussite du clustering des mots vectoriels. Nous avons remarqué que globalement les mots les plus proches de chaque mot obtenu grâce au mots vectoriels exprime bien les champs lexicaux des mots à induction, on arrive donc à comprendre un ou plusieurs sens de chaque mot.

Pour l'analyse de nos résultats, nous avons construit un inventaire de sens pour les 1000 premiers mots de chaque modèle construit avec les word embeddings, la durée de construction de l'inventaire des sens pour les dix modèles varie de 25 minutes minimum à 40 minutes maximum par modèle. Ensuite nous avons choisi au hasard deux mots qui ont plus d'un seul sens « العربية » et « العالم ».Nous avons comparé les résultats pour six modèles OSAC, AraCorpus et WWW de AraVec pour les modèles CBOW et Skip-Gram.

Les figures Fig 2 et Fig 3 présentent les résultats obtenus. Pour chacun des six modèles, il existe un ou plusieurs clusters de sens, chaque cluster (cercle) représente quatre mots arabes dont le sens est proche ainsi que leur distance de similarité avec le mot « العربية » ou « العالم ». Chaque cluster signifie un ou plusieurs sens probables. Par exemple, pour le mot» العربية »: en utilisant le modèle « OSAC C-BOW », le mot a deux sens différents «la nations » et « la langue ». Dans ce cas, les mots vectoriels et le clustering ont réussi à discriminer les différents sens du mot.

Nous avons remarqué aussi dans les résultats obtenus qu'il y a des clusters qui peuvent signifier plusieurs sens possibles comme par exemple pour le mot « العالم »; en utilisant le modèle OSAC Skip-Gram, nous avons obtenu deux clusters de sens, le premier cluster signifie « le monde géographique » alors que le deuxième cluster peut signifier deux sens à la fois « le monde » et « le savant ». Dans ce cas, le clustering a réussi à

discriminer entre le sens de « monde géographique » et « le monde et le savant », mais a échoué pour la discrimination entre les sens « le monde » et le « savant ».

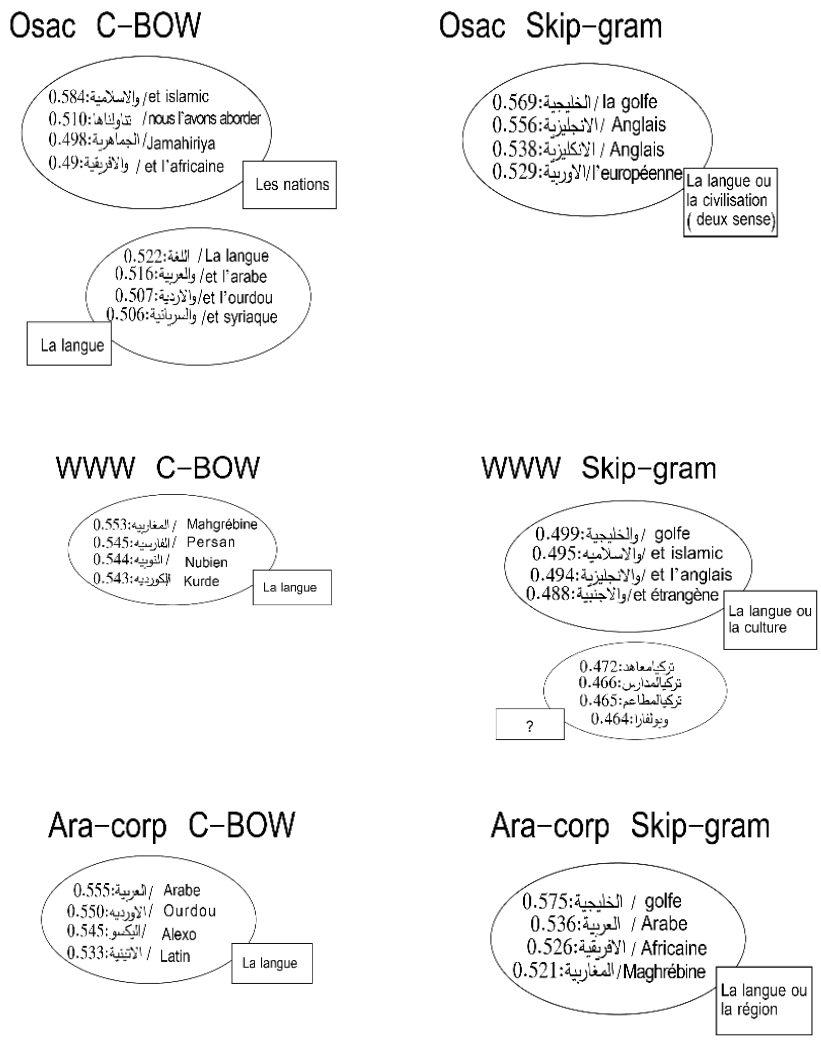

Fig2. Les clusters de sens du mot « العربية » pour les six modèles.

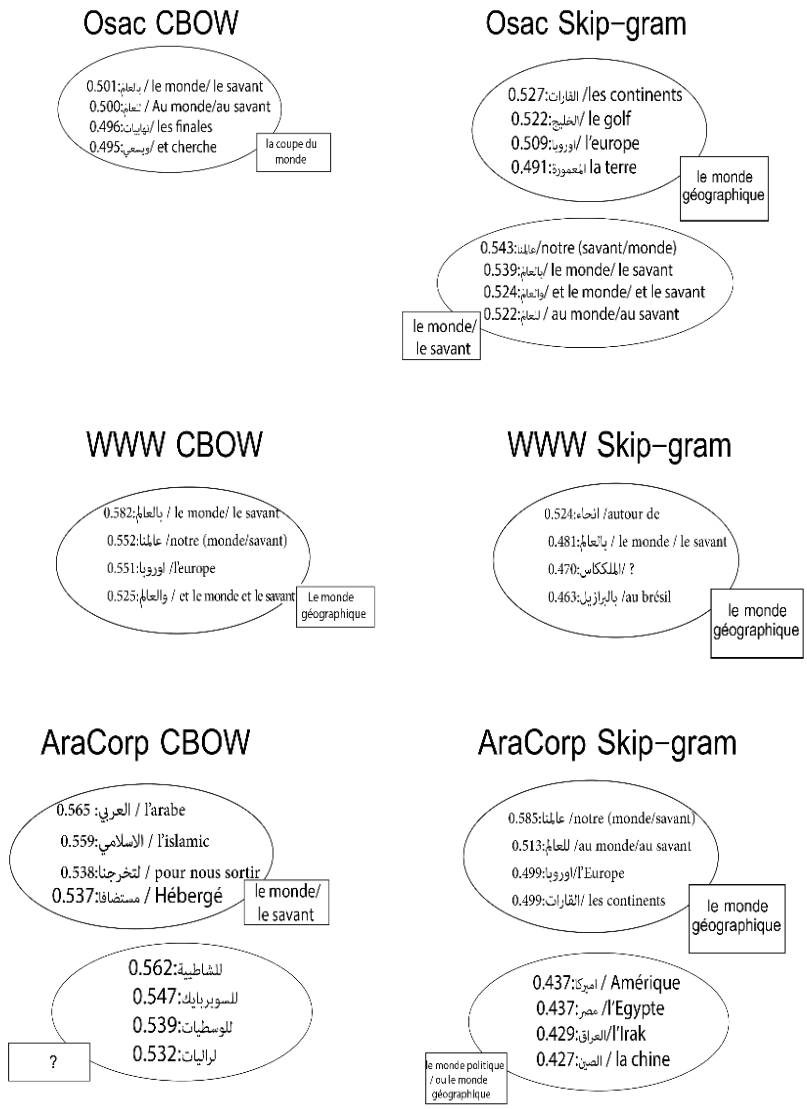

Fig 3. Les clusters de sens du mot « العالم » pour les six modèles.

Afin d'étendre notre évaluation, nous avons sélectionné aléatoirement 9 mots de l'inventaire de sens pour le corpus OSAC obtenu avec les modèles SKIP-gram et CBOW. Pour chaque mot, nous avons clustérisé manuellement les 200 mots voisins les plus proches obtenus avant l'application de l'algorithme de Chinese Whispers. Nous avons supprimé les mots qui sont interprétés correctement et proches pour tous les différents sens du mot étudié. Nous calculons la précision pour chaque mot, dans notre cas, la précision est la somme des précisions de chaque cluster, divisé par le nombre de clusters. La précision d'un cluster mesure le nombre de mots correctement groupés divisé par le nombre total de mots du cluster. Les résultats sont présentés dans

la Table 4.

Table 4. La moyenne des précisions obtenue pour les 9 mots sélectionner pour les deux inventaires de sens OSAC SKIP et OSAC-CBOW

Les mots	Précision (OSAC_SKIP)	Précision (OSAC-CBOW)
العالم	0.64	0.61
العربية	0.57	0.63
عام	0.62	0.65
الأسواق	0.61	0.63
رضوان	0.45	0.48
الاعلام	0.65	0.70
عرض	0.55	0.56
قدر	0.61	0.60
كتب	0.62	0.63
Précision Total	**0.59**	**0.61**

5. Conclusion

Nous avons présenté dans cet article une nouvelle approche d'induction des sens des mots arabes en utilisant des modèles word embeddings qui représentent les mots dans un espace vectoriel. Tout d'abord, nous avons construit des modèles embedding pour la langue arabe en utilisant les corpus arabes disponibles (OSAC et AraCorpus), ensuite nous avons utilisé ces modèles pour induire des sens pour n'importe quel mot du vocabulaire en clustérisant le graphe des mots connectés à l'aide de l'algorithme Chinese Whispers. La construction du graphe de mots connectés pour un mot donné est basée sur l'idée de relier deux voisins d'un mot W si l'un d'entre eux est l'un des K plus proches voisins pour l'autre mot. Nous obtenons les k-plus proches voisins en utilisant l'indexeur de l'outil Annoy qui peut trouver approximativement les voisins les plus proches plus rapidement que la fonction de similarité du GenSim.

Nos résultats sont prometteurs, nous pouvons observer que le choix des corpus et le prétraitement sont deux étapes importantes. À ce stade, nous ne pouvons pas dire lequel des modèles CBOW ou Skip-gram est meilleur pour induire des sens des mots arabes. Cependant, l'utilisation conjointe des deux modèles peut donner de meilleurs résultats.

References

Abdelali, A., Cowie, J., Soliman, H., 2005. "Building a modern standard Arabic corpus." Workshop on computational modeling of lexical acquisition, the split meeting. Croatia.

Agirre, E., Soroa, A., 2007. "SemEval -2007 Task 02: Evalauting Word Sense Induction and Discrimination Systems." 2007.

Panchenko, A., 2016. "Best of both worlds: Makings word sense Embeddings Interpretable." 10th edition of the Language REsources and Evaluation Conference.

Panchenko, A., 2013. *Similarity Measures for semantic relation extraction.* Belgium: Ph.D thesis , University catholique de louvain..

Bartunov, S., Kondrashkin, D., Osokin, A., Vetrov, D., 2016. "Breaking sticks and ambiguities with adaptive skip-gram." Artificial Intelligence and Statistics, 2016.

Bengio , Y., Ducharme, R., Vincent, P., Janvin, C.,2003. "A neural probabilistic Language model." The

Journal of Machine Learning Research, 2003.

Biemann, C., 2012. “Turk Bootstrap Word Sense Inventory 2.0: A large-scale Resources for lexical Substitution.” in proceedings of the 8th international conference on language resources abd evaluation. Istanbul,Turkey, 2012. 4038-4042.

Biemann, C., 2005. “Chinese Whispers: An Efficient Graph Clustering Algorithm and Its application to Natural Language processing Problems.” In proceedings of the 1st workshop of Graph Based methods for Natural Language Processings . NY, USA, 2006.

Cocos, A., Apidianaki, M., Callision-Bruch, C., 2017. “Word sense Filtering Improves Embeddings-Based Lexical Substituation.” in the proceedings of the 1st workshop on sense,concept and entity representations and thier applications. Valencia,Span, 2017.

Djaidri, A., Aliane, H., Azzoune, H, 2018 . “A new Arabic Word Embeddings Model for Word Sense Induction.” 19th International Conference on Computational Linguistics and intelligent Text Processing , CICling 2018.

Hanneman, A. R., Riddle, M., 2005. Introduction to social network methods. *Riverside, CA: universty of California*, 2005.

Iacobacci, I., Pilehvar M.T., Navigli, R., 2016. “Embeddings for Word sense Disambiguation: an evaluation study.” 2016.

Ibrahim, A., 2016 “Abu EL-kheir Corpus: A modern Standard Arabic Corpus.” *international Journal of Recent Trends in Engineering & Research IJRTER*, 2016.

Jurafsky, D., Martin, J.H, 2016. *Speech and Language Processing.* 2016.

Kozlowski, M., Rybinski,H.,2017. “Word Sense Induction with Closed Frequent Termsets.” Computational Intelligence, 2017.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. “Efficient estimation of word representations in vector space.” CoRR. 2013.

Motaz, S., Wesam, A., 2010. “OSAC: Open Source Arabic Corpus.” *6th ArchEng International Symposiums,EEEECS' 10 the 6th international Symposium on Electrical and Electronics Engineering and computer Science.* Cyprus, 2010.

Kwong, O.Y., 2013. “Word Senses and Problem Difinition.” *in: New Prespectives on Computational and Cognitive Strategies for Word Sense Disambiguation , springer Briefs in Eletrical and comptuer Engineering.* NY, USA, 2013.

Panchenko, A., Ruppert, E., Faralli, S., Ponzetti, S. P., Biemann, C., 2017. “Unsupervised Does not Mean Uninterpretable: the Case for Word Sense Induction and Disambiguation.” EACL. 2017.

Pelevina, M., Arefiev, N., Biemann, C., Panchenko, A., 2016. “Making Senes of Word Embeddings.” Proceedings of the 1st Workshop on Representation Learning for NLP. 2016.

Pinto, D., Rosso, P., Jiménez-Salazar, H., 2007. “UPV-SI: Word Sense Inducion using self-term expansion.” in the proceeding of the 4th international workshop on semnatics evaluations. Prague,Czech Republic, 2007. 430-433.

Nguyen, Q., Nguyen, Q., D., Modi, A., Thater, S., Pinkal, M., 2017. “A Mixture Model for Learning Multi-Sense Word Embeddings .” Proceedings of the 6th Joint Conference on Lexical and Computational Semantics. Vancouver,Canada, 2017. 121-127.

Rehurek, R., Sojka, P., 2010. “Software framework for topic madeliing with large corpora.” in proceedings of the LREC 2010 workshop on new challenges for NLP frameworks. 2010.

Soliman, A. B., Eissa, K., El-Beltagy, R., 2017. “AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP.” 3rd International Conference on Arabic Computational Linguistics ACLing 2017. Dubai, 2017