

Plongements lexicaux spécifiques à la langue arabe: application à l'analyse d'opinions

Amira Barhoumi^{1,2} Nathalie Camelin¹ Chafik Aloulou² Yannick Estève¹
Lamia Hadrich Belguith²

(1) LIUM, adresse, 72000 Le Mans, France

(2) MIRACL, adresse, CP Sfax, Tunisie

amira.barhoumi.etu@univ-lemans.fr, nathalie.camelin@univ-lemans.fr,
chafik.aloulou@fsegs.rnu.tn, yannick.esteve@univ-lemans.fr,
l.belguith@fsegs.rnu.tn

RÉSUMÉ

Nous nous intéressons, dans cet article, à la tâche d'analyse d'opinions en arabe. Nous étudions la spécificité de la langue arabe pour la détection de polarité. Nous nous focalisons ici sur les caractéristiques d'agglutination et de richesse morphologique de cette langue. Nous avons particulièrement étudié différentes représentations d'unité lexicale : token, lemme et light stemme. Nous avons construit et testé des espaces continus de ces différentes représentations lexicales. Nous avons mesuré l'apport de tels types de représentations vectorielles dans notre cadre spécifique. Les performances du réseau CNN montrent un gain significatif de 2% par rapport à l'état de l'art.

ABSTRACT

Arabic-specific embeddings : application in Sentiment Analysis

In this article, we are interested in Arabic sentiment analysis task. We study the specificity of the Arabic language for the detection of polarity. We focus on the agglutination and morphological richness of this language. We particularly studied lexical units of different granularities : tokens, lemmas and light stems. We have built and tested continuous spaces of these lexical units. We have measured the contribution of such types of embeddings in our specific framework. The performance of the CNN network has a significant gain of 2%.

MOTS-CLÉS : Analyse d'opinion, représentation vectorielle continue, apprentissage profond, langue arabe.

KEYWORDS: Sentiment analysis, embeddings, deep learning, arabic language.

1 Introduction

Avec Internet et l'explosion des réseaux sociaux, un grand nombre d'internautes expriment leurs points de vue et leurs sentiments sur des entités, des produits, des personnes, *etc.* Dans ce cadre, le domaine d'analyse automatique d'opinions est en plein essor. Il consiste souvent à identifier la subjectivité et la polarité (positive, négative, neutre) d'un énoncé donné (Pang *et al.*, 2008; Yang *et al.*, 2017). Dans ce travail, nous nous intéressons à l'analyse d'opinions à partir de textes rédigés en langue arabe. Les travaux effectués dans ce domaine permettent de distinguer trois approches. La première est symbolique, elle utilise des lexiques (Abdulla *et al.*, 2014a) et des règles linguistiques

(Almas & Ahmad, 2007; Farra *et al.*, 2010). La deuxième consiste en une approche statistique qui s'appuie sur des méthodes d'apprentissage automatique (Abdulla *et al.*, 2014b; Bayouddhi *et al.*, 2015). Quant à la troisième, elle est hybride, elle utilise à la fois des lexiques et des algorithmes d'apprentissage automatique (El-Halees, 2011; Ibrahim *et al.*, 2015; Refaee & Rieser, 2016).

La recherche en analyse d'opinions a tiré profit des avancées scientifiques dans les techniques d'apprentissage profond, et plusieurs travaux ont été récemment réalisés avec ce type d'apprentissage. (Al Sallab *et al.*, 2015) teste différents réseaux profonds. (Dahou *et al.*, 2016; Barhoumi *et al.*, 2018), quant à eux, utilisent une architecture à base de réseau convolutif CNN. La majorité des réseaux neuronaux prennent comme entrée des représentations vectorielles continues (*embeddings*) de mots. L'espace de projection est un espace continu supposé préserver les similarités sémantiques et syntaxiques des mots. Les word embeddings se sont révélés être un atout fondamental pour plusieurs tâches de traitement du langage naturel, y compris l'analyse d'opinions.

Word2vec (Mikolov *et al.*, 2013) et Glove (Pennington *et al.*, 2014) sont parmi les algorithmes de construction d'embeddings les plus répandus. Plus récemment, les embeddings contextuels *Elmo* (Peters *et al.*, 2018) et BERT (Grave *et al.*, 2018) sont apparus pour gérer à la fois les contextes linguistiques et la syntaxe/sémantique des mots. Pour la langue arabe, quelques ressources d'embeddings sont disponibles. (Dahou *et al.*, 2016) a entraîné le modèle word2vec (Mikolov *et al.*, 2013) de type CBOW sur des pages web. (Soliman *et al.*, 2017) regroupe six modèles d'embeddings (CBOW et Skip-gram) entraînés sur trois types de corpus différents : twitter, wikipédia et des pages web. (Barhoumi *et al.*, 2018) utilise ces embeddings de mots pré-entraînés et obtient de bonnes performances pour l'analyse d'opinions en arabe.

Néanmoins, des améliorations peuvent encore être obtenues si l'on prend en compte les nombreuses spécificités de la langue arabe. Notamment, si l'on considère que la définition d'un mot, au sens graphique, est une séquence de caractères délimitée par deux séparateurs (blanc ou autre marqueur de séparation, tel que la ponctuation), alors un mot en arabe peut avoir une structure très complexe. En effet, ce mot peut être décomposable en proclitique(s), forme fléchie et enclitique(s). Par exemple, le mot **أسيعبه** /AsyEjbh/ (est-ce qu'il va lui plaire), se compose d'une particule d'interrogation **أ** et de futur **س**, de la forme fléchie **يعجب** et du pronom relatif **ه** qui sont tous agglutinés. Dans cette perspective, nous supposons qu'une décomposition en éléments simples du mot complexe pourrait réduire la taille du vocabulaire arabe (comprenant plusieurs centaines de millions de mots) et augmenter les occurrences de chacun des éléments et leurs contextes. Ceci pourrait améliorer la qualité des représentations vectorielles. Dans cet article, notre objectif consiste donc à vérifier et valider cette hypothèse.

Les embeddings pré-entraînés existants représentent un mot arabe sans considération des caractéristiques d'agglutination et de la richesse morphologique de l'arabe. Nous nous focalisons dans ce travail sur la spécificité de la langue arabe et nous construisons des embeddings pour différentes représentations du mot : le token (unité morfo-syntaxique simple, le mot est notamment séparé de la ponctuation), le lemme (forme canonique du token) et le light stemme (suppression des affixes du token). À notre connaissance, c'est le premier travail proposant la construction de tels embeddings. Ces derniers seront prochainement disponibles gratuitement.

Dans cet article, nous nous intéressons à la détection d'opinions par les méthodes d'apprentissage profond pour la langue arabe. Nous effectuons nos expériences sur le corpus *Large-scale Arabic Book Review* (LABR), corpus de critiques de livres en arabe. Nous présentons en section 2 notre méthodologie pour la construction des embeddings spécifiques à l'arabe. Nous proposons ensuite, en section 3, notre système neuronal basé sur un réseau de neurones convolutifs (CNN) en justifiant le

choix de quelques hyperparamètres suite à une analyse du corpus. Nous présentons et analysons, en section 4, les performances. Finalement, nous concluons et exposons les perspectives de ce travail en section 5.

2 Embeddings spécifiques à l'arabe

L'objectif de ce travail consiste à construire des embeddings de mots arabes pré-traités et de les évaluer pour la tâche d'analyse d'opinions¹. Nous détaillons dans cette section le processus de construction de tels embeddings.

2.1 Préparation du corpus d'entraînement des embeddings spécifiques

Nous proposons d'utiliser comme corpus d'entraînement des embeddings spécifiques une fusion de différents corpus arabes existants portant sur la tâche de détection d'opinions ou sur des articles de presse. Nous disposons de quatre de ces corpus. Les trois premiers sont des corpus d'opinion : BRAD (Elnagar & Einea, 2016) regroupant 510k critiques sur livres, HARD (Elnagar *et al.*, 2018) constitué de 373K commentaires sur films et le corpus d'apprentissage de LABR (Nabil *et al.*, 2014) formé de 23K de critiques sur livres. Le dernier corpus *AbuElKhair* (El-Khair, 2016) regroupe 5222k articles de presse. Ce corpus sera noté *Global*.

Nous avons nettoyé le corpus fusionné : nous avons supprimé les urls, les mentions, les hashtags, les nombres, les signes de ponctuation et les mots non arabes. Dans ce travail, nous envisageons une classification binaire de commentaires en positifs et négatifs. Les moyens d'accentuation et d'emphase ne nous semblent donc pas très pertinents pour la classification. Nous avons ainsi supprimé le caractère d'allongement (*kashida* ou *tatouil*) et gardé deux occurrences de caractères identiques consécutifs. Nous avons normalisé les caractères $\{ \tilde{\text{ا}}, \tilde{\text{آ}} \}$ à un simple alif $\{\text{ا}\}$. Nous avons également supprimé les voyelles courtes arabes et les diacritiques (*soukoun* et *chadda*). Ce nettoyage permet de normaliser les différentes possibilités d'écriture de mots en arabe ce qui permet de réduire la taille du vocabulaire et ainsi diminuer le nombre d'hapax.

2.2 Construction des embeddings

L'arabe est une langue caractérisée par son agglutination et sa richesse morphologique. L'application d'outils TAL semble être nécessaire pour réduire la fausse diversité du vocabulaire arabe et construire des espaces d'embeddings : de tokens, de lemmes et de light stemmes. Le token est obtenu par un outil qui permet de séparer les clitiques de l'unité morpho-syntaxique simple. La lemmatisation consiste à réduire tous les mots fléchis à leur forme canonique. Le light stemming consiste à supprimer les préfixes et les suffixes fréquemment utilisés avec les mots sans les réduire à leurs racines. Nous avons appliqué un tokeniseur², un lemmatiseur³ et un light stemmer⁴ à notre corpus *Global*.

1. Les embeddings proposés peuvent être utilisés dans d'autres tâches NLP tel que le résumé automatique, *etc.*

2. <http://qatsdemo.cloudapp.net/farasa/>

3. <http://qatsdemo.cloudapp.net/farasa/>

4. <https://github.com/motazsaad/arabic-light-stemming-py>

Les différents espaces d’embeddings sont construits en utilisant le modèle word2vec (Mikolov *et al.*, 2013). La version Skip-gram de ce dernier est meilleure pour l’analyse d’opinions en anglais (Kim, 2014) et arabe (Barhoumi *et al.*, 2018). Nous avons donc construit 3 espaces d’embeddings (de dimension 300) dédiés aux 3 différentes unités lexicales (tokens, lemmes et light stemmes) avec le type skip-gram. La table 1 reporte la taille de chaque espace. Pour information, le vocabulaire sous forme de mots est de taille 3 millions.

	token	lemme	light stemme
Taille	1 980 255	946 171	1 997 601

TABLE 1 – Taille des espaces d’embeddings proposés pour l’arabe.

3 Système d’analyse d’opinions pour l’arabe

Les réseaux convolutifs CNN ont prouvé leurs performances dans l’analyse d’opinions pour l’anglais (Kim, 2014) et l’arabe (Barhoumi *et al.*, 2018). Nous avons donc choisi cette architecture pour implémenter notre système et évaluer ses performances pour l’arabe⁵. Nous décrivons dans la suite l’architecture du CNN et nous détaillons le choix de quelques hyperparamètres liés à notre analyse du corpus.

3.1 Architecture du CNN

Le CNN prend en entrée une matrice d’*embeddings* de taille fixe et applique une convolution de filtres, dont la taille de la fenêtre est une des valeurs de l’ensemble $\{3, 4, 5\}$, pour extraire de nouveaux attributs à partir de la matrice d’*embeddings*. Puis, un *max_pooling* est appliqué sur la sortie de la couche de convolution dans le but de conserver uniquement les attributs les plus pertinents qui sont concaténés au niveau d’une couche entièrement connectée. Enfin, le CNN applique la fonction *sigmoid* à la couche de sortie pour générer la polarité du document fourni en entrée. Deux polarités sont possibles : positif ou négatif.

3.2 Choix d’hyperparamètres

3.2.1 Longueur du document

Le réseau convolutif CNN prend comme entrée une matrice de taille fixe (cf. section 3.1). Dans notre cas, l’entrée du CNN sera la matrice représentant un commentaire par l’ensemble de ses mots, chacun d’eux représenté par un embedding. Or, chaque commentaire ne contient pas le même nombre de mots. Il convient donc de définir la taille du document. Pour cela, nous utilisons la formule 1 qui nous permet de déterminer le nombre maximum de mots qui représenteront le document.

$$seuil = moyenne + 2 \times \text{ecart type} \quad (1)$$

En appliquant cette formule sur le corpus d’apprentissage de LABR, nous calculons une longueur moyenne des commentaires de 64 mots et un écart type de 117,71 mots. Nous obtenons donc un seuil de 300 mots. Ainsi, chaque document sera représenté par une matrice de 300 mots par 300

5. Ce qui nous distingue de (Barhoumi *et al.*, 2018) est les embeddings d’entrée au réseau CNN

composantes (taille de l’embedding d’un mot). Dans le corpus d’apprentissage de LABR, plus de 96% des commentaires contiennent moins de 300 mots. Nous définissons dans la section suivante comment représenter les documents en fonction de leur taille.

3.2.2 Padding/Truncating

Lorsque la taille du document est supérieure à celle fixée, il est nécessaire de couper (et ainsi ignorer) les mots supplémentaires : c’est le *truncating*. Lorsque les documents sont plus courts, il est nécessaire de combler la représentation du message par des zéros : c’est le *padding*. Or il existe trois façons de procéder à ce truncating/padding : soit couper/comblent sur le début du message (pré), soit sur la fin du message (post), soit de manière égale sur les 2 extrémités.

Pour choisir le protocole padding/truncating le plus adéquat dans notre cas, nous avons procédé à une analyse des mots polarisés contenus dans les documents afin de déterminer quel est le segment qui contient l’information la plus pertinente pour la classification. Dans le cadre de l’analyse d’opinions, cette information regroupe principalement les mots polarisés et les termes de négation qui sont souvent utilisés dans l’expression d’opinions. Pour déterminer la polarité d’un mot, un lexique de mots polarisés a été utilisé. Ce dernier est la fusion de 15 lexiques polarisés existants (arabes ou traduit de l’anglais vers l’arabe). Le lexique résultant contient 51968 mots positifs et 45638 mots négatifs. Pour les termes de négation, une liste prédéfinie regroupe 6 différents termes de négation.

Nous avons effectué des statistiques sur les pourcentages de mots polarisés et de termes de négation pour mesurer l’informativité des segments. Nous avons divisé le document en trois parties et calculé le pourcentage de mots polarisés ou de terme de négation contenu dans chacun des trois segments. Ces statistiques sont reportées dans le tableau 2.

		1 ^{ier} segment	2 ^{ieme} segment	3 ^{ieme} segment
TRAIN	% mots positifs	16,33%	0,73%	0,82%
	% mots négatifs	7,29%	0,34%	0,63%
	% termes de négation	0,74%	0,03%	0,002%

TABLE 2 – Informativité des différents segments du corpus Train de LABR

Une première remarque basée sur le tableau 2 porte sur l’informativité du premier tiers du document qui comprend le plus grand pourcentage de mots polarisés. Les deux autres tiers ne sont pas aussi informatifs que le premier. Nous pourrions donc déduire que les internautes expriment explicitement leurs opinions au début du commentaire et ils se justifient par la suite de manière plus factuelle. Le premier tiers de chaque document semble donc contenir de l’information pertinente pour la classification en polarité. Le post-padding/post-truncating semble ainsi être adapté à l’analyse d’opinion du corpus LABR. Si le document comprend plus de 300 mots, la fin de celui-ci sera donc coupé. S’il est plus petit, il sera complété par le vecteur 0 autant que nécessaire.

4 Résultats et discussion

4.1 Corpus LABR

Pour évaluer notre système, nous avons utilisé le corpus LABR (Nabil *et al.*, 2014) qui contient 63k critiques de livres composées d’un commentaire et d’une note associée (nombre d’étoiles). Nous

nous plaçons dans le cadre d’une classification binaire et regroupons les critiques comme proposé dans (Nabil *et al.*, 2014) : les commentaires associés à une ou deux étoiles composent la classe *negative* et ceux à quatre ou cinq étoiles composent la classe *positive*. Ainsi les commentaires neutres ne sont pas considérés et le corpus utilisé se réduit à un ensemble de 33234 commentaires (84% positifs) pour le corpus d’apprentissage et 8366 pour le corpus de test (85% positifs). Notons que 10% de l’ensemble d’apprentissage est utilisé comme corpus de développement.

4.2 Performance du système d’analyse d’opinions

Cette section présente les performances de notre système neuronal. La table 4 rapporte les performances du CNN avec les différents espaces d’embeddings implémentés (section 2.2). Nous remarquons que quel que soit l’espace d’embeddings, l’exactitude est supérieure à 91%. Les performances des trois embeddings implémentés sont équivalentes.

LABR	unité lexicale			(Barhoumi <i>et al.</i> , 2018)
	token	lemme	light stem	www-sg
Dev	91,52%	91,48%	91,36%	89,82%
Test	91,25%	91,50%	91,50%	89,34%

TABLE 3 – Exactitude du CNN sur LABR avec les différents modèles d’embeddings.

Ainsi, les systèmes CNN avec des embeddings d’unités pré-traitées sont plus performants que le CNN appliqué sur des embeddings de mots existants. Nous pensons que ceci revient certainement à une plus grande couverture du vocabulaire et une représentation plus robuste des unités lexicales. Il serait intéressant de valider cette hypothèse dans d’autres tâches de traitement automatique des langues. Nous avons calculé la matrice de confusion du système *CNN_token* (meilleure performance sur le corpus de développement). Le système prédit bien les critiques positives avec 97,68% de précision et 92,46% de rappel. Les commentaires négatifs sont plus difficiles à détecter avec seulement 54,45% de précision et 81,49% de rappel.

Nous avons également comparé notre système avec les résultats des travaux déjà parus sur le corpus LABR. À notre connaissance, les meilleurs résultats obtenus par un système ne se basant pas sur des connaissances *a priori* de type expert sont ceux du système (Barhoumi *et al.*, 2018). Ce dernier a utilisé un CNN et a atteint 89,34% d’exactitude avec des embeddings de mots de type skip-gram pré-entraînés sur des pages web issus de (Soliman *et al.*, 2017).

Dans ce travail, notre système s’appuie sur une architecture CNN similaire à (Barhoumi *et al.*, 2018). Suite à notre analyse du corpus, nous avons réduit la taille du vecteur représentant le document (de 882 à 300) et avons appliqué un processus de post-padding/truncating (au lieu de pré-). Le gain obtenu en ne changeant que la valeur de ces paramètres est de 0.9% en absolu. Le reste du gain est dû à l’utilisation de nos espaces spécifiques d’embeddings. Une des différences repose sur la couverture des espaces d’embeddings sur le corpus LABR. Celle de nos espaces est d’environ 99% alors que celle des embeddings de mots utilisés dans (Barhoumi *et al.*, 2018; Dahou *et al.*, 2016) couvre 81% de LABR. Ceci explique certainement le gain que nous observons.

Nous proposons d’illustrer les différents résultats en fonction de la représentation lexicale via un exemple. La table 4 reporte les différentes représentations lexicales possibles du commentaire أكثر تمنيت ان الموضوعات و الأحاديث فيه أكثر /tmnyt An AlmwDw’t w AAHdyv fyh Akvr/ (j’aurai aimé

plus de thèmes et de conversations) et dont la classe de référence est positive. Notre architecture CNN utilise la fonction sigmoïde en couche de sortie. Elle associe la classe positive (respectivement négative) si le score de sortie est supérieur (respectivement inférieur) à 0.5. Plus le score est proche de 1 (respectivement 0), plus la prédiction de la classe positive (respectivement négative) est certaine.

Unité lexicale	Commentaire	Classe prédite	Score
Token	تمني موضوع احاديث في أكثر	positive	0.71
Lemme	تمنى ان موضوع احاديث في أكثر	positive	0.6
Light stem	تمنيت ان موضوع احاديث فيه اكثر	négative	0.39

TABLE 4 – Exemple de commentaire et sa prédiction selon les différentes unités lexicales .

Pour cet exemple, le CNN prédit correctement la classe positive avec les embeddings de tokens et ceux de lemmes. Nous remarquons que le score obtenu avec les embeddings de tokens (0.71%) est supérieur à celui obtenu avec les embeddings de lemmes (0.6%) et que la classe prédite avec les embeddings de light stems n'est pas correcte (0,39%).

5 Conclusion et perspectives

Nous avons souhaité prendre en compte les spécificités de la langue arabe. Ainsi, nous avons proposé et construit 3 espaces d'embeddings prenant en compte l'agglutination et la richesse morphologique de la langue arabe. Nous avons mesuré l'utilité de tels embeddings et nous avons trouvé des performances proches, de l'ordre de 91%. Le meilleur résultat sur le corpus de développement est obtenu par les tokens tandis que sur le corpus de test, il est atteint avec des embeddings de lemmes ou light stemmes. Nous notons une amélioration d'environ 2% par rapport à la baseline (89,34%). À notre connaissance, c'est le premier travail qui construit et teste les embeddings de différentes représentations lexicales en arabe.

D'autres pistes d'amélioration restent à explorer pour tenir compte du phénomène d'agglutination et de la richesse morphologique de l'arabe. Nous envisageons notamment de tester les embeddings de n-grammes de caractères à la façon de fasttext, les embeddings de mots à base de caractères ou encore les dernières représentations vectorielles *ELMO*. De plus, en nous appuyant sur les travaux de (Yu *et al.*, 2017) où des *embeddings d'opinions* sont construits pour l'anglais, nous souhaitons également étudier la transposition de ces travaux pour l'arabe en nous appuyant sur les lexiques de mots polarisés que nous avons construits.

Références

- ABDULLA N. A., AHMED N. A., SHEHAB M. A., AL-AYYOUB M., AL-KABI M. N. & AL-RIFAI S. (2014a). Towards improving the lexicon-based approach for arabic sentiment analysis. *International Journal of Information Technology and Web Engineering (IJITWE)*, **9**(3), 55–71.
- ABDULLA N. A., AL-AYYOUB M. & AL-KABI M. N. (2014b). An extended analytical study of arabic sentiments. *International Journal of Big Data Intelligence 1*, **1**(1-2), 103–113.

- AL SALLAB A., HAJJ H., BADARO G., BALLY R., EL HAJJ W. & SHABAN K. B. (2015). Deep learning models for sentiment analysis in arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, p. 9–17.
- ALMAS Y. & AHMAD K. (2007). A note on extracting ‘sentiments’ in financial news in english, arabic & urdu. In *The Second Workshop on Computational Approaches to Arabic Script-based Languages*, p. 1–12.
- BARHOUMI A., CAMELIN N. & ESTÈVE Y. (2018). Des représentations continues de mots pour l’analyse d’opinions en arabe : une étude qualitative. In *Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018*, p. 215.
- BAYOUDHI A., GHORBEL H. & BELGUITH L. H. (2015). Sentiment classification of arabic documents : Experiments with multi-type features and ensemble algorithms. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, p. 196–205.
- DAHOU A., XIONG S., ZHOU J., HADDOUD M. H. & DUAN P. (2016). Word embeddings and convolutional neural network for arabic sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, p. 2418–2427.
- EL-HALEES A. (2011). Arabic opinion mining using combined. *Proceeding the International Arab Conference On Information Technology*.
- EL-KHAIR I. A. (2016). 1.5 billion words arabic corpus. *arXiv preprint arXiv :1611.04033*.
- ELNAGAR A. & EINEA O. (2016). Brad 1.0 : Book reviews in arabic dataset. In *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*, p. 1–8 : IEEE.
- ELNAGAR A., KHALIFA Y. S. & EINEA A. (2018). Hotel arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent Natural Language Processing : Trends and Applications*, p. 35–52. Springer.
- FARRA N., CHALLITA E., ASSI R. A. & HAJJ H. (2010). Sentence-level and document-level sentiment mining for arabic texts. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, p. 1114–1119 : IEEE.
- GRAVE E., BOJANOWSKI P., GUPTA P., JOULIN A. & MIKOLOV T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.
- IBRAHIM H. S., ABDOU S. M. & GHEITH M. (2015). Sentiment analysis for modern standard arabic and colloquial. *International Journal on Natural Language Computing (IJNLC)*, 4(2).
- KIM Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv :1408.5882*.
- 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*, p. 3111–3119.
- NABIL M., ALY M. & ATIYA A. (2014). Labr : A large scale arabic sentiment analysis benchmark. *arXiv preprint arXiv :1411.6718*.
- PANG B., LEE L. *et al.* (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2), 1–135.
- PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

- PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.
- REFAEE E. & RIESER V. (2016). ilab-edinburgh at semeval-2016 task 7 : A hybrid approach for determining sentiment intensity of arabic twitter phrases. *Proceedings of SemEval-2016*, p. 474–480.
- SOLIMAN A. B., EISSA K. & EL-BELTAGY S. R. (2017). Aravec : A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, **117**, 256–265.
- YANG K., CAI Y., HUANG D., LI J., ZHOU Z. & LEI X. (2017). An effective hybrid model for opinion mining and sentiment analysis. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, p. 465–466 : IEEE.
- YU L.-C., WANG J., LAI K. R. & ZHANG X. (2017). Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 534–539.

