Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4955–4963
Marseille, 11–16 May 2020

Toward Qualitative Evaluation of Embeddings for Arabic Sentiment Analysis

Amira Barhoumi [1,2], Nathalie Camelin[1], Chafik Aloulou[2], Yannick Estève[3], Lamia Belguith[2]
Le Mans University [1], Sfax University[2], Avignon University [3]
Avenue Laennec, 72085 LE MANS CEDEX 9, France [1],
Route de Tunis Km 10 B.P. 242 SFAX 3021, Tunisia [2],
Rue Louis Pasteur, 84000 Avignon, France [3]
{amira.barhoumi.etu, nathalie.camelin}@univ-lemans.fr
{chafik.aloulou, l.belguith}@fsegs.rnu.tn
yannick.esteve@univ-avignon.fr

Abstract

In this paper, we propose several protocols to evaluate specific embeddings for Arabic sentiment analysis (SA) task. In fact, Arabic language is characterized by its agglutination and morphological richness contributing to great sparsity that could affect embedding quality. This work presents a study that compares embeddings based on words and lemmas in SA frame. We propose first to study the evolution of embedding models trained with different types of corpora (polar and non polar) and explore the variation between embeddings by observing the sentiment stability of neighbors in embedding spaces. Then, we evaluate embeddings with a neural architecture based on convolutional neural network (CNN). We make available our pre-trained embeddings to Arabic NLP research community with free to use. We provide also for free resources used to evaluate our embeddings. Experiments are done on the *Large Arabic-Book Reviews* (LABR) corpus in binary (positive/negative) classification frame. Our best result reaches 91.9%, that is higher than the best previous published one (91.5%).

Keywords: Embeddings, Word2vec, Sentiment Analysis, Intrinsic and Extrinsic Evaluation, Convolutional Neural Networks, Arabic language.

1. Introduction

Sentiment analysis (SA) task focuses on analysing opinions and sentiments at different levels (word, sentence and document) (Nasukawa and Yi, 2003). In this work, we are dealing with SA at document level. It refers to identifying the polarity of a given textual statement (Pang et al., 2008). Generally, polarity consists in associating positive/negative classes to statements (sometimes extended to more fine-grained categories). SA applications have spread many languages, with a recent growing interest in Arabic language.

The majority of recent works on Arabic SA lies on word embeddings in order to capture semantic and syntactic similarities. Their quality needs large corpora so that each word in the vocabulary appears multiple times in different contexts. These embeddings deal mainly with words as *space separator units*. However, Arabic is characterized by its agglutination and morphological richness that contribute to sparsity. So, it is important to take into account Arabic word complexity.

The majority of works dealing with continuous representations in Arabic NLP has focused on the word unit. In this work, we target the complex structure of Arabic words in SA task by investigating two ways to build embedding sets: one set by lexical unit (word and lemma).

Word embeddings are widely used for various NLP tasks. Embedding evaluation techniques fall into two major categories: intrinsic and extrinsic methods. The aim of this work is to evaluate our proposed embeddings for Arabic sentiment analysis. Indeed, we propose a rigorous protocol to evaluate our embeddings first in an intrinsic evaluation of embedding models during their training, and then evaluate their performances for SA downstream task. The intrinsic method is specific to SA task. It consists in observing the sentiment stability of neighbors in embedding spaces. The extrinsic method discusses performances obtained with convolutional neural network (CNN). We make available our pre-trained embeddings and we also provide for free resources used to evaluate them.

The rest of the paper is structured as follows. Related works are introduced in section 2 We present our methodology to propose embeddings based on word and lemma in section 3. In section 4, we present our neural architecture used for Arabic SA task. We report, in section 5, the experimental framework and discuss obtained results in section 6 Finally, we conclude, in section 7, and give some outlooks to future works.

2. Related works

2.1. Sentiment analysis task

Sentiment analysis research has benefited from scientific advances in deep learning techniques, and several recent works have been done with this type of learning for Arabic [1]. (Al Sallab et al., 2015) tested different deep networks. (Dahou et al., 2016; Barhoumi et al., 2018; Barhoumi et al., 2019) used a convolutional neural network (CNN) architecture. (Hassan, 2017; Heikal et al., 2018; Al-Smadi et al., 2018) used recurrent neural network (RNN) and its variants.

[1] For an overview of Arabic SA field, (Al-Ayyoub et al., 2018; Al-Ayyoub et al., 2019; Badaro et al., 2019) build a complete survey.

The majority of neural networks takes as input continuous vector representations of words (*word embeddings*). Word2vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2016) are the most common algorithms for learning pre-trained embeddings. Contextualized word embeddings *Elmo* (Peters et al., 2018) recently appear to handle both linguistic contexts and word syntax/semantic. There are some embedding resources that are freely available for Arabic language: (Dahou et al., 2016; Soliman et al., 2017) built word embedding sets obtained by training *skip-gram* and *CBOW* versions of word2vec, (Grave et al., 2018) distribute pre-trained word vectors Arabic, trained on Common Crawl and Wikipedia using fastText. (Barhoumi et al., 2018) presents a rigorous comparison of these embedding resources and shows that their systems suffer from a low coverage of pre-trained embeddings at word level.
To the best of our knowledge, (Salama et al., 2018; Barhoumi et al., 2019) are the only works dealing with Arabic specificity in embedding models. (Salama et al., 2018) studied the effect of incorporating morphological information to word embedding in 2 ways: (i) including POS tags with words before embedding and, (ii) performing lemma abstraction of morphological embeddings obtained in (i). (Barhoumi et al., 2019) proceeded differently and built embeddings for different Arabic lexical unit (word, token, token\clitics, lemma, light stem and stem).

2.2. Embedding evaluation techniques

Embedding evaluation techniques fall into two categories: intrinsic and extrinsic. On one hand, intrinsic evaluation methods (Baroni et al., 2014; Schnabel et al., 2015) consist in quantifying directly various linguistic regularities in embedding space. Syntactic and semantic analogies (Mikolov et al., 2013; Nayak et al., 2016) are the most used intrinsic methods. On the other hand, extrinsic evaluation methods assess the quality of embeddings for other NLP tasks such as part-of-speech tagging, chunking, named-entity recognition, sentiment analysis, *etc*.
The majority of works dealing with Arabic word embedding evaluation use extrinsic technique. Many Arabic embeddings models have been evaluated in applications such as machine translation (Shapiro and Duh, 2018; Lachraf et al., 2019), Sentiment analysis (Dahou et al., 2016; Soliman et al., 2017; Fouad et al., 2019), Information retrieval (El Mahdaouy et al., 2018), *etc*. Many downstream applications show the usefulness of word embeddings. For intrinsic evaluation of Arabic word embeddings, (Elrazzaz et al., 2017) is the only work up to our knowledge. It quantifies syntactic and semantic analogies in embedding spaces.
In this work, we evaluate embeddings with both intrinsic and extrinsic methods within SA frame. We propose a new protocol for intrinsic evaluation showing the sentiment stability of neighbors in embedding spaces that will be detailed in section 6.1.

3. Methodology

In this section, we explain specificity of Arabic language in subsection 3.1, and justify our choice of word and lemma as lexical units in subsection 3.2. Then, we present our intuitions for embedding construction in subsection 3.3.

3.1. Arabic language specificity

Arabic language is characterized by its agglutination and morphological richness. It is also difficult due to diacritization problem. For example, the word جمل /jml/ can be interpreted in 3 ways:

- جَمَل /jamalun/ (camel) with neutral polarity.
- جُمَل /joumalun/ (sentenses) with neutral polarity.
- جَمَّل /jammala/ (beautify) with positive polarity.

Each interpretation is made by different diacritization and reflects well-defined polarity.
Moreover, Arabic word structure is very complex. Indeed, if we consider a word as a sequence of characters delimited by separators (blank or any punctuation mark), this word is composed of inflected form and it may contain zero or several clitics. It can be decomposed into proclitic(s) at its beginning, inflected form and enclitic(s) at its end. For example, the word اسيعجبه /AsyEjbh/ (will he like it?) consists of interrogation ا /A/ and future س /s/ particles, inflected form يعجب /yEjb/ and relative pronoun ه /h/, which are all agglutinated. The complexity of Arabic is that these clitics could be found also in inflected forms. The latter could be found alone, thus constituting words. For example, the particle س is part of the verb سمح /smh/ (allow).

3.2. Choice of lexical units

Based on complex structure of Arabic words, (Salama et al., 2018) showed the utility of lemma embeddings. (Barhoumi et al., 2019) conducted an exhaustive comparison of six possible Arabic lexical units. Both works show that lemmas are the best. That is why in this work, we focus on lemmas and we carry out a thorough qualitative embeddings evaluation. Indeed, lemma represents the lexical unit that keeps relevant semantic information which is important in sentiment analysis task. Lemmatization refers to the process of relating a given textual item to the actual lexical or grammatical morpheme corresponding to dictionary input. In this way, lemmatization allows reducing sparsity and vocabulary size (see table 1).

Word			**Lemma**
	gender	number	
جميلَان	male	dual	جميل (pretty)
جميلون	male	plural	
جميلَات	female	plural	
احبّ	male and female	singular	احبّ (love)
يحبّون	male	plural	
تحبَّان	male and female	dual	

Table 1: Examples showing reducing sparsity between word and lemma levels.

In addition to lemmas, we decide to evaluate also words in order to see closely the gain obtained by lemmas.

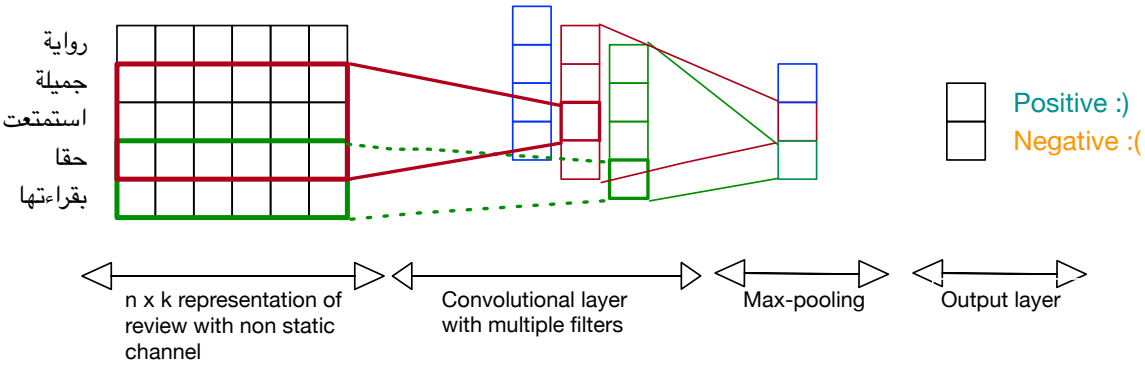

Figure 1: CNN architecture for an example review

3.3. Embedding sets

Embedding models are trained with multiple parameters. (Antoniak and Mimno, 2018) found that nearest neighbors are highly sensitive to small changes in embedding training corpus. (Pierrejean and Tanguy, 2018) explored the impact of changing window size, embedding dimension and training corpora. In this work, we want to know if the type of training corpora affects SA task performance. In other words, is it better to train embeddings with task-specific (polar) corpora in SA framework? or generic corpora are more efficient? And what about corpora size?

For rigorous study, we build embedding models trained with three types of corpora: polar, non polar and mixed at word and lemma levels. This will be detailed in section 5.1. In addition to corpora type, we investigate the impact of epoch number used for model training on the neighborhood of polar units.

4. Description of our sentiment analysis system

Several papers show that CNN architecture gives good performance for sentiment analysis (Kim, 2014; Dahou et al., 2016; Barhoumi et al., 2018). In the same way, we choose to consider a CNN architecture. We develop a CNN architecture similar to the one described in (Dahou et al., 2016) and train it according two modes: with or without adaptation of embeddings. Indeed, we test static and non static CNN learning ways (Kim, 2014) in order to evoke trainable and non trainable aspects of embeddings. Trainable embeddings obtained with non static CNN allow obtaining task-specific embeddings. They are updated while learning task system. Non trainable embeddings are obtained with static CNN, they are not updated during CNN training. We evoke, in this work, trainable/untrainable embeddings in order to respond the following question: for a particular task (SA in this work), is it better to use embeddings models trained with task-specific (polar) corpora or to use task-specific (trainable) embeddings? We want to compare performances of system that takes as input: (i) embeddings that are trained with task-specific corpora and are non updated (untrainable) during system training, or (ii) embeddings that are trained with generic (non task-specific) corpora and are updated (trainable) during system training.

CNN takes as input our embeddings proposed in previous subsection 3.3. Each document *Doc*[2] is represented by a fixed-size matrix of embeddings $M(n, k)$ with n the length of the document and k the dimension of the embedding. CNN applies a *convolution* via filters whose window size is in $\{3, 4, 5\}$, in order to extract new features from the embedding matrix $M(n, k)$. Then, *max_pooling* is applied to the output of the convolution layer in order to only preserve the most relevant features that are concatenated at a fully connected layer with *dropout*[3]. Finally, the CNN applies the *sigmoid* function to the output layer to generate the polarity of the input document. Two polarities are possible: positive or negative. The architecture is illustrated in Figure 1. Many hyper-parameters could be fine-tuned in CNN architecture: size of filters, rate of the dropout, pooling way, *etc.* We detail in the following the choices of two parameters: document length and padding/truncating type.

4.1. Document length

As mentioned above, the CNN input is a fixed-size matrix. In our case, the matrix represents a review: each word occurrence of the review is represented by an embedding. In order to choose the fixed-size n of documents (*i.e.* the number of words to take into account), we use the formula (1) with the hypothesis of Gaussian length's distribution.

$$n = m + 2 \times SD \quad (1)$$

where: *m* the mean of word number in the documents and *SD* the standard deviation.

This empirical rule suppose that 95% of documents lies within two standard deviations, which means that length of 95% of documents is under or equal to n.

4.2. Padding/Truncating

We define here how to represent documents in case the number of words is not n. When the length of any review is greater than n, it is necessary to cut additional words: it is the *truncating*. And when the review is shorter, then it is necessary to fill the representation of the review with zeros: it is the *padding*. But there are three ways to proceed: cut/fill at the beginning of the document (pre), or at the end (post), or equally on both extremities.

[2]Each word w_i in *Doc* is represented by x_i: a k dimension vector ($x_i \in \mathbb{R}^k$)

[3]The rate of the dropout is 0.5

Corpora Type	Word			Lemma		
	#voc	#occ	#1	#voc	#occ	#1
Polar	897	47 932	454	392	46 809	245
Non polar	2 875	1 190 762	1231	1 392	1 185 847	726
Mixed	3 196	1 238 695	1441	1 614	1 232 656	885

Table 2: Statistics of corpora per type, where #voc: vocabulary size, #occ: corpus size, and #1: number of units that appear one time in the corpus. All sizes are reported in *kilo (k)*.

To choose the most appropriate padding/truncating protocol, we propose to conduct an analysis of polar words contained in the documents to determine which segment contains the most relevant information for classification. In SA context, this information mainly includes polar words and negation terms that are often used in opinion expression. To determine the polarity of a word, a lexicon of polar words is needed. For negation terms, we define a list of the following Arabic negation terms: { مَا, ليس, لن, لم, لَا, غير}.

Statistics have to be computed on the experiment corpus to measure segment informativeness with regards to the presence of polar words and negation terms. Documents are divided into three equal parts and the percentage of polar words or negation terms contained in each of the three segments is calculated. If the most informative segment is the first one, post-padding/truncating will be applied. However, if the third segment is the most informative, pre-padding/truncating will be applied. The equal-padding/truncating on both extremities is applied in the third case.

5. Experimental framework

In this section, we present the framework setup: training corpora and models used to build embeddings, dataset used to train and evaluate our SA system, polar lexicon used to choose parameters of our SA system (document length and type of padding/truncating) and measure sentiment stability in our embedding spaces.

5.1. Corpora for embedding construction

For embedding construction, we consider three types of corpora: polar, non polar and mixed.

Polar corpus are datasets that are usually used in document level SA. It contains star-rated reviews written by Internet users in order to express their opinions related to a given entity. These reviews are annotated by stars in a scale form 1 to 5 stars. Our polar corpus is a concatenation of BRAD (Elnagar and Einea, 2016) gathering $510k$ book reviews, HARD (Elnagar et al., 2018) composed of $373k$ hotel reviews and the train set of LABR (Nabil et al., 2014) formed by $23k$ book reviews.

Non polar corpus does not contain rated reviews. It is composed of generic textual statements that are not specific to SA. We choose news to construct non polar corpus. The latter is composed of $5\ 222k$ news (El-Khair, 2016).

To go further, and in order to have larger corpus, we concatenate previous polar and non polar corpora and obtain mixed corpus.

A pre-processing is applied to clean and normalize our three corpora. Then, we apply Farasa lemmatiser [4] in order to lemmatise our three corpora. As a result, we obtain two datasets per corpus type: each one corresponds to one lexical unit $\in$ {word, lemma}. The size of each dataset is reported in table 2. As expected, there are less lemmas than words. Indeed, the size of lemma vocabulary represents around one half the size of word vocabulary. Also, the number of lemmas appearing one time is very small compared to the number of words appearing one time in the corpus. So, lemmatization actually helps on reducing sparsity of Arabic.

We trained word2vec on these 6 datasets, obtaining 6 embedding sets. The latter is freely available [5].

5.2. Embedding models

Word2vec model (Mikolov et al., 2013) is a three-layer neural network that is trained to reconstruct linguistic contexts of words. It produces word embeddings, such that words sharing common contexts are closely located in the embedding space. Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or skip-gram (SG).

Based on results shown in (Mikolov et al., 2013; Dahou et al., 2016; Barhoumi et al., 2018), we use skip-gram for embedding construction and we set embedding dimension to 300 as in (Dahou et al., 2016; Soliman et al., 2017; Salama et al., 2018; Grave et al., 2018).

5.3. Corpus for sentiment analysis

In this work, we use *Large Arabic-Book Reviews* corpus LABR (Nabil et al., 2014) to evaluate our SA systems. It contains 63k book reviews: a note (number of stars from 1 to 5) is associated to each review.

In binary classification framework, we regrouped the reviews as proposed in (Nabil et al., 2014): the reviews associated with one or two stars compose the *negative* class and those with four or five stars represent the *positive* class. Thus the neutral reviews are not considered. So, the corpus used, in this work, is composed of 33 234 reviews (84% positive) for the training set and 8 366 for the test set (85% positive). This is the official train / test split. Note that we use 10% of the training set as a validation set.

5.4. Sentiment lexicon

Sentiment lexicon is an important resource for SA task. It consists of a set of couples (ω, s) where ω: a word (or

[4] http://qatsdemo.cloudapp.net/farasa/

[5] https://lium.univ-lemans.fr/en/arsentimentanalysis/

phrase) and s: a sentiment score.

In this work, we collected all available sentiment lexicons up to our knowledge (Badaro et al., 2014; ElSahar and El-Beltagy, 2015; Saif M. Mohammad and Kiritchenko, 2016; Al-Moslmi et al., 2018). This represents a set of 15 lexicons constructed with different methods.

The first method consists in automatically translating English lexicons. Indeed, translated resources (Saif M. Mohammad and Kiritchenko, 2016) are obtained by translating the following four English lexicons: MPQA (Wilson et al., 2005), S140 (Kiritchenko et al., 2014), NRC (Mohammad and Turney, 2010; Mohammad and Yang, 2011; Mohammad et al., 2013) and Bing liu lexicon (Hu and Liu, 2004).

The second method is based on Pointwise Mutual Information (PMI) between words and two labels (positive and negative). Indeed, the sentiment orientation (SO) of a word (Mohammad and Turney, 2013) represents the difference between PMI scores.

The used lexicons have various size and different structures. In fact, each word is described with different features. These features vary from one lexicon to another. (Badaro et al., 2014) built 4 616 word annotated with their part-of-speech tags, positive and negative scores, offsets in Arabic WordNet, *etc*. (ElSahar and El-Beltagy, 2015) built 5 lexicons with different domains: hotel, restaurant, movie, book and product, each size of which is 217, 733, 86, 873 and 368 respectively. (Saif M. Mohammad and Kiritchenko, 2016) created Arabic sentiment lexicons automatically using two different methods: (1) using PMI-SO on Arabic tweets, and (2) automatically translating English sentiment lexicons into Arabic. Three lexicons are built on Arabic tweets: Arabic Emotion lexicon (43 308 words), hashtag lexicon (22 006 Arabic words and 20 127 dialectal ones). And four translated lexicons which are MPQA (8 221 words), S140 (26 740 words), NRC (32 582 words) and Bing lui lexicon (6 789 words). (Al-Moslmi et al., 2018) built a lexicon of 3 880 positive and negative synsets annotated with their part of speech, polarity, sentiment scores and synonyms.

As a result, our lexicon *ArSentLex* supports several kinds of word-level annotations such as (i) translation of predefined sentiment lexicons and (ii) PMI-SO method.

ArSentLex is defined as a set tuple (ω, pos, ps, ns, p), where: ω is a word, pos its part-of-speech tag, ps its positive score, ns its negative score, and p its polarity (positive or negative). Indeed, each ω is described by four features pos, ps, ns and p which are relevant for SA.

For the concatenation of 15 lexicons, we applied the following steps:

- When polarity feature is not presented in one lexicon, we associate polarity based on positive and negative scores. That means positive polarity is attributed to a word when positive score is higher than negative one and vice versa.

- When a word ω is described by synonyms as in (Al-Moslmi et al., 2018), we add as many lines as there are synonyms. For each synonym, we assign the same features of ω.

		Polarity		
	Unit	**positive**	**negative**	$\cap$
Lex_word	**word**	47 856	42 366	11 307
Lex_lemma	**lemma**	38 340	34 403	9 783

Table 3: Size of our lexicon *ArSentLex* at word and lemma level.

Our lexicon *ArSentLex* is wide compared to others. Table 3 shows number of polar units contained in our lexicon at word and lemma level. *ArSentLex* is freely available [6]. To the best of our knowledge, there is no publicly available large scale Arabic sentiment lexicon similar to our *ArSentLex*.

5.5. Choice of CNN parameters

Document length: Applying the formula (1) - proposed in subsection 4.1- to LABR training dataset, we obtain an average review's length of 64 words and a standard deviation of 118 words. So, we get a threshold n of 300 words, and each document will be represented by 300 words.

For information, more than 96% of reviews in LABR training corpus ($\geq 95\%$ of empirical rule presented above) contains less than 300 words.

Padding/Truncating: The protocol proposed in subsection 4.2 is applied to LABR training corpus.

Statistics are reported in table 4. It shows that the sentiment informativeness of the first segment of documents includes the largest percentage of polar words and negation terms. The other two-thirds are not as informative as the first one. We could therefore suppose that Internet users explicitly express their opinions at the beginning of the review and then justify themselves in a more factual way. The first third of each document therefore seems to contain relevant information to polarity classification.

	1st segment	**2nd segment**	**3rd segment**
% positive words	16,33	0,73	0,82
% negative words	7,29	0,34	0,63
% negation terms	0,74	0,03	0,002

Table 4: Informativity of different segments in LABR training set.

As a result, the post-padding/truncating seems to be the most appropriate to SA of LABR dataset. If the document contains more than 300 words, its end will be cut off. If it is smaller, it will be filled at the end by 0 as necessary.

6. Results and discussion

In this section, we present and discuss results of intrinsic and extrinsic evaluation of our embeddings presented in subsections 5.1 and 5.2.

[6] https://lium.univ-lemans.fr/en/arsentimentanalysis/

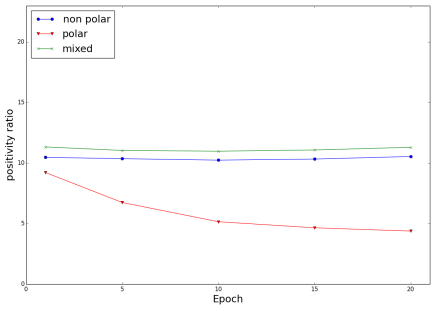

Figure 2: Positivity ratio at word level

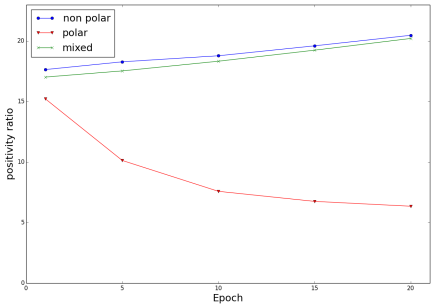

Figure 3: Positivity ratio at lemma level

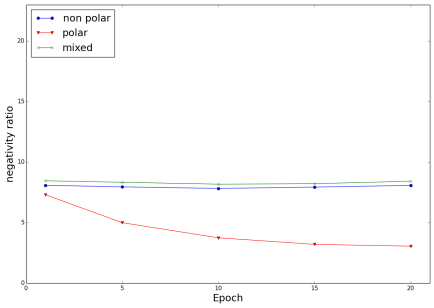

Figure 4: Negativity ratio at word level

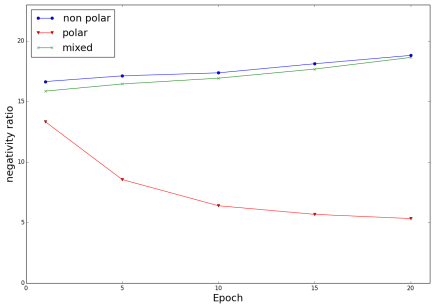

Figure 5: Negativity ratio at lemma level

Figure 6: Sentiment stability of embedding models according to corpora types and lexical units.

6.1. Intrinsic evaluation of embeddings

For intrinsic evaluation of embeddings, we did not use usual methods based on syntactic and semantic analogies witch are generic and non specific to SA task. Considering SA framework, we introduce the notion of *sentiment stability (SS)* of polar units (word and lemma) in embedding spaces. We suppose that polar units in embedding spaces are surrounded by polar units. For each polar unit in *ArSentLex*, we study their nearest neighbors in order to see if they are

in the *ArSentLex* lexicon too (*Lex_word* and *Lex_lemma*).
In order to analyse SS of each model at one specific epoch, we consider at one time: one lexical unit ∈ {word, lemma}, one corpus type ∈ {polar, non polar, mixed} and one epoch number, and we compute how many positive units are in nearest neighbors with positive neighbors (and respectively for negative units). The nearest neighbors of a given unit *unit* are units having the closest cosine similarity score with *unit*.
Whatever the lexical unit, sentiment stability consists in studying stability of *positivity ratio* $(\%^{+}_{Top_n})_{ep_i}$ and negativity ratio $(\%^{-}_{Top_n})_{ep_i}$ according to epoch numbers [7]. The positivity ratio at epoch ep_i is computed for positive units (in positive lexicon $lexicon^{+}$) with formula (2):

$$(\%^{+}_{Top_n})_{ep_i} = 100 \times \frac{\sum_{unit_i \in \{lexicon^{+}\}} \#unit^{lexicon^{+}}_{Top_n}}{n \times \#lexicon^{+}} \quad (2)$$

where: n the number of nearest neighbor unit, $\#unit^{lexicon^{+}}_{Top_n}$ the number of positive units in top_n nearest neighbors of $unit$, $\#lexicon^{+}$ the size of positive lexicon.
We also compute *negativity ratio* according to the same formula (2) by considering only negative units (in negative lexicon $lexicon^{-}$).
Results are reported in Figure 6. Note that, the lower is difference in positivity (or negativity) ratios, the higher is sentiment stability. That means the model is more stable when there is less difference between positivity (or negativity) ratios along epochs. We notice that positivity and negativity ratios $(\%^{+/-}_{Top_n})_{ep_i}$ are close whatever the number of epochs $i \in \{1, 5, 10, 15, 20\}$ for non polar and mixed corpora. We note also that these models are more stable than those obtained with polar corpus.
Histograms in Figure 6 show there is, at maximum, 3 points of ratio difference within models trained with non polar and mixed corpora. Contrariwise with polar corpus, there is, at least, 3 points of ratio difference. We could conclude that non polar and mixed models are more stable than polar models. Nevertheless, we should not ignore the relatively small size of polar corpora (see table 2) which could probably explain the non stability of corresponding model.
For polar models, we note that the best ratios are obtained with models trained in epoch 1. Then, ratios decrease along epochs. It seems, for more epochs, polar units are less close and more scattered in embedding spaces.
A comparison between word ratios and lemma ones represents a relevant note in this work. We notice that positivity and negativity ratios of lemma models are higher than those of word models, whatever the epoch number and for all training corpora types. This could justify our initial hypothesis concerning the need of taking into account Arabic specificity. And it could prove utility of lemma embedding for Arabic language.

6.2. Extrinsic Evaluation of embeddings

We train and evaluate our CNN-based SA system (in static and non static modes) with all our embeddings built considering all corpora types {polar, non polar, mixed}, units {word, lemma} and epoch numbers {1, 5, 10, 15, 20}.
Table 5 reports performances of CNN-based system trained with our proposed embeddings. We only mention epoch number ep_i of the embedding model giving the best classification accuracy (Acc). Accuracy computes the ratio of true predicted (positive and negative) labels on LABR test dataset.

Embeddings		**CNN**			
		static		**non static**	
Corpora	Unit	Acc	ep_i	Acc	ep_i
polar	word	91.5	1	91.2	20
	lemma	**91.9**	1	**91.6**	1
non polar	word	90.9	10	90.9	1
	lemma	91.0	10	91.3	15
mixed	word	91.2	15	91.0	1
	lemma	91.3	1	91.1	15

Table 5: Accuracy (Acc) of CNN systems trained with polar, non polar and mixed Arabic-specific embeddings.

Our best system is obtained with static CNN and polar lemma embeddings. It gives 91.9% of accuracy. It outperforms the existing systems tested on LABR dataset. (Barhoumi et al., 2019) obtained the best previous results on LABR dataset by a system giving an accuracy of 91.5%. Which means that our system has a gain of 0.4 in absolute. It is important to mention that our CNN architecture is similar to the one of (Dahou et al., 2016). The only difference with our CNN architecture is the 2 parameters: document length and padding/truncating type described in section 4. To measure the impact of parameter adjustment on CNN performance, we tested the system of (Dahou et al., 2016) with our new parameters, and obtained an accuracy of 90%. So, we conclude the profit of parameter adjustment that brings 0.4% on gain.
Based on its confusion matrix, our best system predicts positive reviews with 93.06% precision and 97.76% recall. Negative reviews are more difficult to detect with 82.01% precision and only 58.46% recall. Our system therefore shows a weakness in prediction of negative class.
It is also important to note that performances obtained with lemma embeddings are slightly higher than those obtained with word embeddings. This is true for static and non static CNN, and whatever the corpora type used for learning embeddings. It seems that lemma embeddings are better for SA task. The size of lemma embedding sets represents around one half the size of word embedding sets (see table 2). That means we could obtain competitive CNN performances with only one half embedding set size.
Moreover, we note that for static or non static CNN, the best results are obtained with polar embeddings. We could conclude that, for SA task, it is pretty good to use embedding models trained with task-specific corpora (even small) than generic corpora (more amount of data).
Furthermore, in order to compare static and non static CNN according to corpora types, we notice that for polar embeddings, the best results are obtained with static CNN.

[7] In this work, we set top nearest neighbors by $n = 10$ (the most used value in literature).

However, for non polar embeddings, the best results are obtained with non static CNN. We could conclude that for static CNN, it is better to use embeddings trained with task-specific corpora. However, when embeddings are trained with non task-specific corpora, it is better to use non static CNN. It seems that for SA task, it is better to use embeddings trained with task specific corpora than task-specific trainable embeddings.

With polar models, best results are obtained with static CNN. In static CNN frame, we note that embedding models trained in epoch 1 (ep_1) give the best performances. These results can be justified by the quality of polar models mentioned in section 6.1. In fact, if we join static CNN performances with positivity and negativity ratios of embedding models (detailed in section 6.1), we could conclude that in static CNN, the higher are ratios, the better are CNN performances.

7. Conclusion

In this paper, we presented a qualitative evaluation of embeddings for Arabic SA task. We investigated the use of embedding sets trained with different types of corpora (polar, non polar and mixed) at different lexical units (word and lemma). Our embedding resources are available to Arabic NLP research community with free to use.

In this work, we evaluated embeddings with both intrinsic and extrinsic methods within SA frame. We proposed a new protocol for intrinsic evaluation specific to SA. It studies the sentiment stability of polar units in embedding spaces based on a huge sentiment lexicon *ArSentLex* that is also available. Our protocol allows to analyse the sentiment stability in embedding space.

For extrinsic evaluation, we discussed performances of our static and non static CNN-based systems in order to evaluate our embeddings for SA task. Our best system is obtained with static CNN and lemma embeddings trained with polar corpora. Its accuracy reaches 91.9% which is higher than all previous works on LABR dataset. We had a gain of 0.4 in absolute.

In future work, we want to enlarge our lexicon and enhance its quality by taking into account embeddings. We will develop a neural architecture to train a sentiment-aware word embedding by integrating the sentiment supervision at both document and word levels, thus enhancing the quality of word embedding as well as the sentiment lexicon.

8. Bibliographical References

Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., and Al-Kabi, M. N. (2018). A comprehensive survey of arabic sentiment analysis. *Information Processing & Management*.

Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., and Al-Kabi, M. N. (2019). A comprehensive survey of arabic sentiment analysis. *Information Processing & Management*, 56(2):320–342.

Al-Moslmi, T., Albared, M., Al-Shabi, A., Omar, N., and Abdullah, S. (2018). Arabic senti-lexicon: Constructing publicly available language resources for arabic sentiment analysis. *Journal of Information Science*, 44(3):345–362.

Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., and Shaban, K. B. (2015). Deep learning models for sentiment analysis in arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 9–17.

Al-Smadi, M., Talafha, B., Al-Ayyoub, M., and Jararweh, Y. (2018). Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews. *International Journal of Machine Learning and Cybernetics*, pages 1–13.

Antoniak, M. and Mimno, D. (2018). Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (2014). A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*, pages 165–173.

Badaro, G., Baly, R., Hajj, H., El-Hajj, W., Shaban, K. B., Habash, N., Al-Sallab, A., and Hamdi, A. (2019). A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):27.

Barhoumi, A., Camelin, N., and Estève, Y. (2018). Des représentations continues de mots pour l'analyse d'opinions en arabe: une étude qualitative. In *25e conférence sur le Traitement Automatique des Langues Naturelles (TALN 2018)*, Rennes, France, May.

Barhoumi, A., Camelin, N., Aloulou, C., Estève, Y., and Hadrich Belguith, L. (2019). An empirical evaluation of arabic-specific embeddings for sentiment analysis. In Kamel Smaïli, editor, *Arabic Language Processing: From Theory to Practice*, pages 34–48, Cham. Springer International Publishing.

Baroni, M., Dinu, G., and 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 Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Dahou, A., Xiong, S., Zhou, J., Haddoud, M. H., and 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*, pages 2418–2427.

El-Khair, I. A. (2016). 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

El Mahdaouy, A., El Alaoui, S. O., and Gaussier, E. (2018). Improving arabic information retrieval using word embedding similarities. *Int. J. Speech Technol.*, 21(1):121–136, March.

Elnagar, A. and Einea, O. (2016). Brad 1.0: Book reviews

in arabic dataset. In *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*, pages 1–8. IEEE.

Elnagar, A., Khalifa, Y. S., and Einea, A. (2018). Hotel arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent Natural Language Processing: Trends and Applications*, pages 35–52. Springer.

Elrazzaz, M., Elbassuoni, S., Shaban, K., and Helwe, C. (2017). Methodical evaluation of Arabic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–458, Vancouver, Canada, July. Association for Computational Linguistics.

ElSahar, H. and El-Beltagy, S. R. (2015). Building large arabic multi-domain resources for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 23–34. Springer.

Fouad, M. M., Mahany, A., Aljohani, N., Abbasi, R. A., and Hassan, S.-U. (2019). Arwordvec: efficient word embedding models for arabic tweets. *Soft Computing*, Jun.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Hassan, A. (2017). Sentiment analysis with recurrent neural network and unsupervised neural language model.

Heikal, M., Torki, M., and El-Makky, N. (2018). Sentiment analysis of arabic tweets using deep learning. *Procedia Computer Science*, 142:114–122.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Lachraf, R., Billah Nagoudi, E. M., Ayachi, Y., Abdelali, A., and Schwab, D. (2019). ArbEngVec : Arabic-English Cross-Lingual Word Embedding Model. In *The Fourth Arabic Natural Language Processing Workshop*, Florence, Italy, July.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mohammad, S. M. and Turney, P. D. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics.

Mohammad, S. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *CoRR*, abs/1308.6297.

Mohammad, S. M. and Yang, T. W. (2011). Tracking sentiment in mail: How genders differ on emotional axes. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*, pages 70–79. Association for Computational Linguistics.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Nabil, M., Aly, M., and Atiya, A. (2014). Labr: A large scale arabic sentiment analysis benchmark. *arXiv preprint arXiv:1411.6718*.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.

Nayak, N., Angeli, G., and Manning, C. D. (2016). Evaluating word embeddings using a representative suite of practical tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 19–23.

Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Pierrejean, B. and Tanguy, L. (2018). Towards qualitative word embeddings evaluation: Measuring neighbors variation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 32–39, New Orleans, Louisiana, USA, June. Association for Computational Linguistics.

Saif M. Mohammad, M. S. and Kiritchenko, S. (2016). Sentiment lexicons for arabic social media. In *Proceedings of 10th edition of the the Language Resources and Evaluation Conference (LREC)*, Portorož, Slovenia.

Salama, R. A., Youssef, A., and Fahmy, A. (2018). Morphological word embedding for arabic. *Procedia computer science*, 142:83–93.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307.

Shapiro, P. and Duh, K. (2018). Morphological word embeddings for Arabic neural machine translation in low-resource settings. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 1–11, New Orleans, June. Association for Computational Linguistics.

Soliman, A. B., Eissa, K., and El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.