Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification

Abdelghani Dahou, Shengwu Xiong, Junwei Zhou*, Mohamed Houcine Haddoud and Pengfei Duan
Hubei Key Laboratory of Transportation Internet of Things,
School of Computer Science and Technology, Wuhan University of Technology
Wuhan 430070, China
dahou@whut.edu.cn xiongsw@whut.edu.cn junweizhou@msn.com
haddoud.medhoucine@gmail.com duanpf@whut.edu.cn

Abstract

With the development and the advancement of social networks, forums, blogs and online sales, a growing number of Arabs are expressing their opinions on the web. In this paper, a scheme of Arabic sentiment classification, which evaluates and detects the sentiment polarity from Arabic reviews and Arabic social media, is studied. We investigated in several architectures to build a quality neural word embeddings using a 3.4 billion words corpus from a collected 10 billion words web-crawled corpus. Moreover, a convolutional neural network trained on top of pre-trained Arabic word embeddings is used for sentiment classification to evaluate the quality of these word embeddings. The simulation results show that the proposed scheme outperforms the existed methods on 4 out of 5 balanced and unbalanced datasets.

1 Introduction

A growing number of people get used to give their opinions on social network websites, forums, video sharing websites, blogs and e-commerce websites, leading to a most rising research fields caused by the important opinionated web contents. The opinions could be used for many applications such as consumer modeling, sales prediction, opinion survey or user intent understanding. Sentiment analysis which is used to identify, extract and classify subjective information in the opinions, has attracted a lot of attention. Sentiment analysis can be divided into several levels: document level (Yessenalina et al., 2010), sentence level (Farra et al., 2010), word/term level (Engonopoulos et al., 2011) or aspect level (Chifu et al., 2015).

Currently, sentiment analysis is commonly used for English, while sentiment analysis on the Arabic language is still recognized at its early stages (Nabil et al., 2015; ElSahar and El-Beltagy, 2015), since sentiment analysis on Arabic is considered as a more challenging work. Firstly, Arabic has a very complex morphology and structure. Inflectional and derivational nature of Arabic language makes the monophonically analysis on Arabic more difficult (Hammo et al., 2002). Secondly, Arabic Internet users mostly use dialectal Arabic rather than Modern Standard Arabic (MSA), while MSA is the formal written language but dialectal Arabic is used in informal daily communication. Moreover, dialectal Arabic is not included in education systems or standardized (Habash, 2010). The diversity of different writings and the language cultural creates more challenges to learn and build language models for representations. Nowadays, more than 267 million people speak Arabic as the first language, and more than 250 million as the second language covering 58 countries[1]. There are around 168.1 million Arabic Internet users with a user growth rate of 6,592.5% (November 2015 by Internetworldstats[2]) , making research of sentiment analysis on the Arabic language important.

In this paper, we try to solve the problems of word embeddings and sentiment classification for Arabic text. We firstly use a web-crawler to build a 10 billion Arabic words corpus, and we realize an word embeddings model to produce Arabic word representations using this corpus. Finally, a convolutional

* Corresponding author. Tel.: +86 027 87298267.

[1]http://www.ethnologue.com/statistics/size.

[2]http://www.internetworldstats.com/stats7.htm

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2418–2427, Osaka, Japan, December 11-17 2016.

neural network (CNN) trained on top of pre-trained Arabic word embeddings is used for sentiment classification. The simulation results show that the proposed scheme has a better performance than existed approaches for the Arabic sentiment classification on different datasets. For the convenience of the evaluation of our scheme, we distribute freely the source code and the generated word embeddings on the web[3].

The rest of the paper is organized as follows. The related work will be introduced in Section 2. Section 3 refers to the construction of word embeddings model for Arabic, and Section 4 refers to the CNN model for sentiment classification of Arabic text. Experiments and analyses will be given in Section 5. The conclusion is drawn in Section 6.

2 Related Works

In this Section, we review existing works related to the proposed scheme. We start with the works on word embeddings, which represent individual words of a language as vectors onto a lower dimensional vector space. Then we introduce the works related to sentiment classification. These methods and techniques can be used to generate semantic representations of texts and perform classification for various tasks in NLP, which are the interesting and related works to our study.

Via neural language models, various deep learning methods have been proposed to learn word vector representations. WORD2VEC (Mikolov et al., 2013) has been proposed for building word representations in vector space, which consists of two models, including continuous bag of word (CBOW) and Skipgram (Skip-Gram). Global vectors for word representations are also used to build word representations (Pennington et al., 2014), where training is based on statistics of word to word co-occurrence from a corpus.

Recently, sentiment classification becomes one of the most motivating research area among natural language processing (NLP) community. Many tools and applications have been applied to Arabic sentiment classification. (Abdul-Mageed et al., 2011) reported efforts for classifying MSA news data at the sentence level for both subjectivity and sentiment using support vector machine (SVM) classifier. (Abdul-Mageed et al., 2014) presented an SVM-based system for subjectivity and sentiment analysis for Arabic social media named SAMAR. (Farra et al., 2010) proposed an Arabic sentence level classification based on syntactic and semantic approaches. (El-Halees, 2011) proposed a combined classification approach for document level sentiment classification using different classifiers, including lexicon based classifier, maximum entropy classifier and k-nearest neighbors (KNN) classifier. More recently, CNN have achieved remarkably strong performance tackling NLP tasks and gotten some interesting results (Kalchbrenner et al., 2014; Kim, 2014). (Kalchbrenner et al., 2014) introduced a dynamic CNN for modeling sentences, and (Kim, 2014) proposed an improved scheme which employs dynamic-updated and static word embeddings simultaneously for sentence classification based on CNN.

3 Arabic Word Embeddings

Machine learning offers significant benefits for representations of a word from text and understanding natural language. The WORD2VEC tool[4] (Mikolov et al., 2013) remains a popular choice benefited from its fast training and good results. CBOW and Skip-Gram in WORD2VEC use a probabilistic prediction approach which captures syntactic and semantic word relationships from very large data sets. In this work, we explore several architectures to build neural word embeddings for MSA and dialectal Arabic. In particular, we perform a comprehensive analysis to train and evaluate word representations using CBOW and Skip-Gram models, with the goal of generating a better quality representations for Arabic sentiment classification. Fig. 1 shows the steps from collecting the corpus to building word representations.

[3] http://pan.baidu.com/s/1eS2mxCe

[4] https://code.google.com/archive/p/word2vec/.

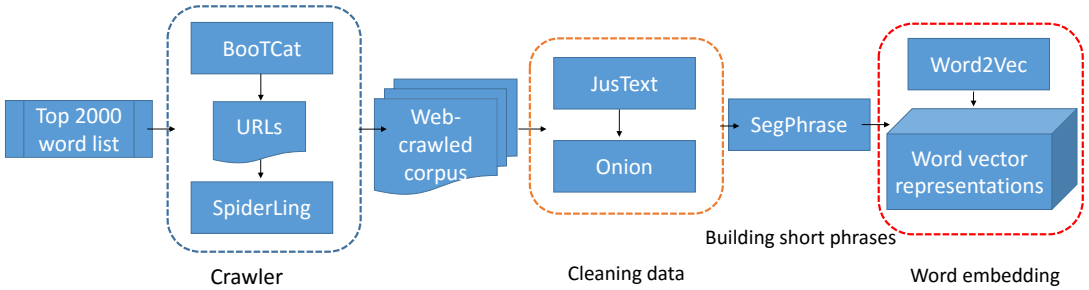

Figure 1: The processes of building Arabic word embeddings

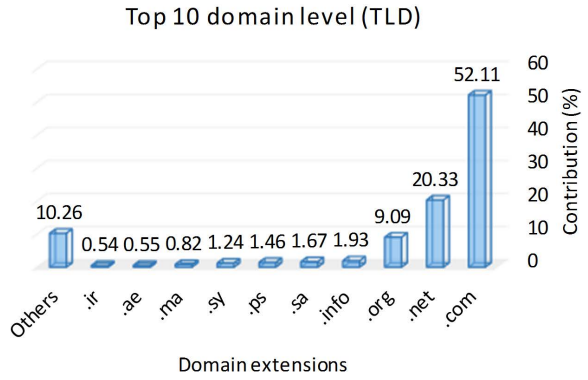

Figure 2: Top 10 Internet top-level domains (TLD).

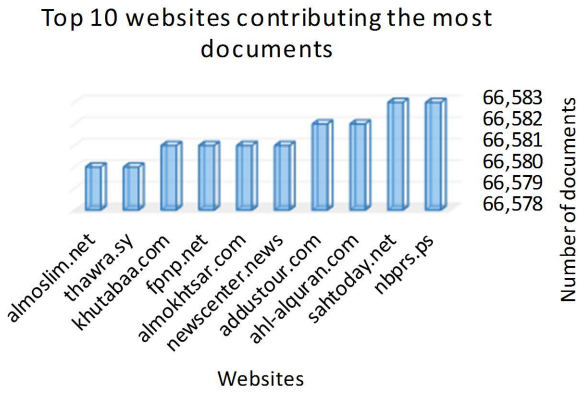

Figure 3: Top 10 websites exploited from the crawler seeding list which contribute the most documents.

3.1 Crawling and Preparing the Corpus

The existing corpora are rare freely accessible for download and are typically not large enough for CNN based Arabic sentiment classification. In order to generate Arabic word representations, we create a huge web-crawled corpus for MSA and dialectal Arabic text.

The acquisition of the corpus is performed by means of SpiderLing[5] (Suchomel et al., 2012), which is an open-source and free crawler for effective creation and annotation of linguistic corpora.

The steps for creating the corpus are described as follows:

1) The seeds for the crawler consist of 10,421 URLs, which are generated using Bing queries, after feeding BootCAT [6] with a word-list of top 2000 words based on their frequency. It should be noticed that the word-list has to be filtered from stop words. Then, the seeds are filtered from duplicated URLs, while the filtered seeds are fed to SpiderLing as a start point for Arabic web pages crawling. The top-10 level web domains used by SpiderLing to produce our corpus are shown in Fig. 2, while Fig. 3 presents the list of top-10 domains that contributed the most of documents.

2) For data normalization, we used jusText[7] (Pomikálek, 2011) which is a heuristic based boilerplate removal tool, which is used to exclude contents such as navigation links, advertisements and headers from downloaded web pages. It only keeps paragraphs containing full sentences and removes contents which are not in the desired language. Also, we used Onion[8] (Pomikálek, 2011) for near-duplicate detection and removing. The deduplication is performed on paragraph level and threshold is set to 0.5.

3.2 Pre-processing and Normalization

An intrinsic limitation of word representations for the Arabic language is that sometimes we need two words to represent one meaning. For example, the word “acceptable” and its antonym “unacceptable”.

[5]http://corpus.tools/browser/spiderling.
[6]http://bootcat.sslmit.unibo.it/.
[7]http://code.google.com/p/justext/.
[8]https://code.google.com/p/onion.

However, the antonym word has no direct (word-to-word) Arabic translation, غير and مقبول can't be easily combined to obtain "غير_مقبول" which is the closest translation to the antonym word. Motivated by this kind of limitation, we thus use SegPhrase[9] (Liu et al., 2015) to learn these phrases.

Based on a data-driven approach presented in (Mikolov et al., 2013) which relies on the count of unigram and bigram to form phrases, we run an experiment using the following equation:

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)}, \quad (1)$$

where δ is a discounting coefficient to prevent the formation of phrases with infrequent words. Bigrams with a score over the chosen threshold are then formed as phrases.

We use an Arabic text consisted of 1.3 billion words and a vocabulary[10] of 3.5 million words to form short phrases based on equation (1). From this experiment, we notice that the average amount of formed phrases occupy 47.42 % of the new vocabulary when we set the threshold to 100. These short phrases occupy 16.18 % from the total number of tokens in the Arabic text. We also notice that raw frequency methods could not reflect the quality of these phrases, since both good and bad phrases can possess high frequency.

To learn these semantical and meaningful phrases and to generate better word embeddings for Arabic, we choose SegPhrase rather than the previous approach. SegPhrase is a framework that aims to extract and mine quality phrases from a large text, combined with a module for phrase segmentation. Thus, text data is transformed from word granularity to phrase granularity. To use SegPhrase for Arabic text, we also need to build two knowledge bases, where the smaller one consists of 94,544 labels and contains high-quality phrases for positive labels. Moreover, the larger one contains 121,127 labels, which is used to filter medium and low-quality phrases for negative labels.

Some examples are سان فرانسيسكو (San Francisco), المملكه العربيه السعوديه (Kingdom of Saudi Arabia) and هواري بومدين (Houari Boumediene) labeled to be positive. In contrast, phrases like تسويق دولي (International marketing), تسلسل زمني (Chronology) and تحت المجهر (Under the microscope) were labeled as negative. A threshold of raw frequency with value 10 is specified for frequent phrase mining, which will generate a candidate set. Another parameter is used to parse our corpus using the generated model which is the ratio of top ranked phrases with value 0.5.

The processing steps for cleaning and normalizing the corpus are as follows : 1) Remove punctuation, diacritics, non letters , and non Arabic. 2) Normalization of the different writings of the latter (Alef) إ , أ , آ with ا. 3) Convert the letter (Teh Marbuta) ة to ه.

3.2.1 Training Parameters

By using Word2Vec tool, we build several models based on different architectures and word vector dimensionality. Table 1 shows the training parameters used for various models based on CBOW and Skip-Gram architectures.

Model	dimensionality	Window size	Sample	Negative	Freq. thresh.	Max iterations
CBOW	100,200,300	5	$1 \times e^{-5}$	10	10	3
Skip-Gram	100,200,300	10	$1 \times e^{-5}$	10	10	3

Table 1: Word vector representations training parameters

For the quality evaluation of the generated Arabic word embeddings, we use word analogy questions task. We compare the vectors in (Zahran et al., 2015) with our vectors.

[9]https://github.com/shangjingbo1226/SegPhrase.

[10]We did not set a frequency threshold because we want to use all the words even the rare words.

4 CNN for Sentiment Classification

In this Section, we tackle our downstream semantic task which is Arabic sentiment classification relying on the Arabic word embeddings model from the previous section. It is a binary classification task between positive and negative and covers two domains: reviews and tweets. Here, the standard 10-fold cross validation is applied to the balanced and unbalanced datasets to report accuracy on all different used datasets.

4.1 CNN Architecture

We consider a CNN architecture similar to that described in (Kim, 2014), with one channel that allows the adaptation of pre-trained vectors for each task. The hyper-parameters used for the training of all models[11] will be discussed in later paragraphs.

Due to the absence of a large supervised training set and as described in (Socher et al., 2011; Iyyer et al., 2014), the training word vectors are initialized by a pre-trained model, in our case the CBOW58[12] model. Where each vector has a dimensionality of $k = 300$ and is trained using CBOW architecture. An initialization of word vectors for all words which are out from CBOW58 model's vocabulary has been performed by sampling from a uniform distribution in the range of [-0.25,0.25] following (Kim, 2014) work.

Here, we define S as a sentence of n words, and let m_i be the ith word in the sentence S[13], where each $m_i \in S$ is represented by x_i a k-dimension vector such as $x_i \in \mathbb{R}^k$.

To perform convolution operation via linear filters over S, we convert S to a sentence matrix of shape $(n \times k)$. Where each row corresponds to a vector x_i [14]. Considering our filter $w \in \mathbb{R}^{hk}$ we set 3 different window widths (of h words) from $D \subset \mathbb{N}^*$, where $D = \{3, 4, 5\}$, with a fixed length k for each filter dimensionality. The application of a convolution operation using one filter window size $h \in D$ over the sentence matrix produces new features.

To capture most relevant global features, and deal with variable sentence lengths. We use a *max-over-time* pooling operation (Collobert et al., 2011) to downsample the feature maps.

After concatenating all feature maps in one single vector with a fixed length, we feed this vector through a fully-connected layer with $Dropout$ (Hinton et al., 2012). Here, the $Dropout$ rate is set to 0.5, and a sigmoid function to generate the final classification. A gradient based optimization named Adagrad (Duchi et al., 2011) with Mini-batch size of 32 is used. The output is the probability distribution over labels. Fig.4 provides a schematic illustrating of the model architecture.

5 Experiments and Analyses

In this section we explore the process of analyzing the quality and the impact of word embeddings using two major evaluations. We start by the intrinsic evaluation using word analogy questions task in Section 5.1. Secondly, we do an extrinsic evaluation by performing Arabic sentiment classification task after defining a CNN model trained on top of the generated word embeddings model in Section 5.2.

5.1 Vector Quality Evaluation

The goal of word analogy questions is to correctly identify the relationship between *C* and *D*, given a relationship between words *A* and *B*. The questions will be in the form of $A : B : C : D$, where the pairs of word (*A*, *B*) and (*C*, *D*) are sharing the same relation (e.g. "man:woman", "king:queen"). We hide the identity of the fourth word D and we predict it based on the other three words, using similarity measure functions like cosine similarity, or Euclidean distance. To find a word that is closest to *D* measured by cosine distance, an algebraic computation can be performed on word embeddings. Here, we simply compute vector $vector(D) = vector(C) - vector(A) + vector(B)$

[11] All experiments run with Keras and Theano on an NVIDIA GeForce GTX 780 Ti GPU.

[12] CBOW58 is the best model of Arabic word embeddings built in Section 3 and the results are listed in Section 5.1.

[13] We use the same zero-padding strategy as in (Kim, 2014).

[14] x_i refers to word vector extracted from $CBOW58$ model with $k = 300$.

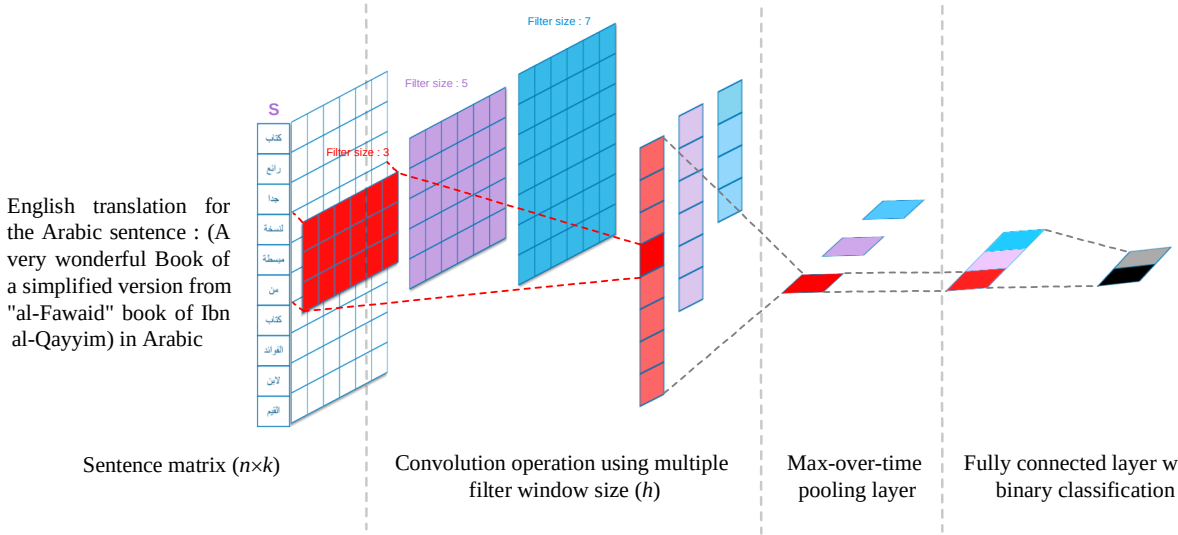

Figure 4: Model architecture for an Arabic example sentence.

To test the quality of the vectors, we used the test cases from (Zahran et al., 2015) with the correction of few Arabic spelling errors and the adding of new analogy questions for opposite words such as opposite 2-gram. The generated new test cases for Arabic contains 24,294 questions. Overall, there are 10,463 semantic and 13,831 syntactic questions. It covers 15 analogy questions, 5 types of semantic questions, and 10 types of syntactic related questions. Some examples are shown in Table 2. The cosine similarity measure we used is defined as:

$$\cos(u, v) = \frac{u \cdot v}{\|u\|\|v\|} \tag{2}$$

Analogy question	Word pair 1				Word pair 2			
Common-capital-countries	Rome	روما	Italy	ايطاليا	Bagdad	بغداد	Iraq	العراق
Oppisite 1-gram	Short	قصير	Tall	طويل	Sad	حزين	Happy	سعيد
Opposite 2-gram	Certain	مؤكد	Uncertain	غير مؤكد	Acceptable	مقبول	Unacceptable	غير مقبول

Table 2: Examples of word analogy questions test set

Equation (3) is used to calculate the cosine similarity of the analogy questions to predict the closest word. Where V is the used vocabulary ignoring the three question words *B*, *A* and *C*. An answer on one of the question analogy is counted correct only if one of the top five predicted words matches the fourth word *D*.

$$\arg\max_{D \in V}(\cos(D, C)\cos(D, A) + \cos(D, B)) \tag{3}$$

Here we use Equation (3) for the evaluation. The results for all different word vector models are shown in Table 3, where "Cov" is an abbreviation for coverage and "Acc" for accuracy, and we represent different dimension as $i-$d where i denotes the dimension size (100-D describe a dimensionality of 100).

In Section 3, we have shown the processing steps to build the corpus and the training parameters for our word embeddings. Collecting a big corpus for these tasks is a major step, but the most important is the evaluation part for the quality of the collected data and the pre-processing operation. The used data to build Arabic word embeddings models consist of 3.4 billion words and a vocabulary of 2.2 million words, In (Zahran et al., 2015), they used 5.8 billion words in their corpus with a vocabulary of 6.3 million words. More iterations improved the accuracy significantly during the training process of our models. Pre-processing the crawled corpus by mining quality phrases to avoid forming incorrect or low-quality phrases could help to enhance the quality of short phrases. As shown in opposite-2gram analogy

Model	(Zahran et al., 2015)				Our models											
Arch	CBOW		Skip-G		CBOW						Skip-Gram					
Dimention size	300-D		300-D		100-D		200-D		300-D		100-D		200-D		300-D	
Accuracy & Coverage	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov	Acc	Cov
Capital-common-countries	**95.89**	100	94.16	100	85.93	100	90.26	100	90.91	100	85.50	100	90.26	100	92.21	100
Capital-world	69.81	100	69.38	100	68.81	98.16	74.51	98.16	**78.15**	98.16	66.20	98.16	74	98.16	75.92	98.16
Currency	14.95	98	13.54	98	16.37	96	**20.45**	96.00	17.86	96	9.65	96	10.25	96	9.48	96
City-in-state	19.04	96.88	22.09	96.88	29.58	96.88	39.29	96.88	**46.19**	96.88	25.52	96.88	38.58	96.88	41.63	96.88
Family	36.19	100	33.33	100	27.14	100	33.81	100	**39.05**	100	24.52	100	31.43	100	37.14	100
Adjective-toadverb	30.91	100	19.70	100	27.83	100	32.02	100	**34.36**	100	12.44	100	18.97	100	21.67	100
Opposite-1gram	**22.73**	100	13.64	100	11.82	100	12.73	100	17.27	100	13.64	100	17.27	100	20	100
Opposite-2gram	12.80	28.57	9.55	28.57	40.12	69.05	**46.17**	69.05	43.15	69.05	21.19	69.05	27	69.05	29.18	69.05
Comparative	**76.59**	100	68.02	100	59.44	100	66.98	100	69.84	100	40.40	100	55.48	100	56.59	100
Superlative	**72.25**	100	62.22	100	55.02	100	64.20	100	68.28	100	36.65	100	49.15	100	51.42	100
Present-tense	58.18	100	55.56	100	50.45	100	56.26	100	**60.45**	100	40.30	100	49.75	100	51.72	100
Nationality-adjective	**84.18**	100	84.12	100	79.66	98.33	81.76	98.33	83.02	98.33	78.71	98.33	83.22	98.33	83.68	98.33
Past-tense	**73.01**	100	70.13	100	57.18	100	67.56	100	71.41	100	47.95	100	59.94	100	62.37	100
Plural	53.90	100	47.84	100	42.64	100	46.97	100	**54.33**	100	35.71	100	36.80	100	37.45	100
Plural-verbs	95.56	100	94.86	100	94.86	100	96.07	100	**96.47**	100	86.39	100	88.51	100	88.41	100
TOTAL	54.40	94.90	50.54	94.90	49.79	97.23	46.97	97.23	**58.05**	97.23	41.65	97.23	48.71	97.23	50.59	97.23

Table 3: Total accuracy of (Zahran et al., 2015) word embeddings model and our Arabic word embeddings models on word analogy questions test set. All numbers in the table are percentage.

questions type of Table (3), the coverage rate is 69.05 % compared to 28.57 % in a trained model based on Equation (1). According to comparisons based on the word embeddings dimensionality, it could be found that using low dimensionality does not help much to improve the quality of these vectors when trained on a large corpus. Therefore, a major drawback in these neural based language models is that training on a higher dimensionality is time consuming.

These analogy evaluation scores for different models and architectures are not a great estimator of real-world tasks performance. The use of these word embeddings in a downstream task such as Arabic sentiment classification in our case will be determined by whether these vectors are of good quality or not. These scores provide us with a general idea of what our data looks like.

5.2 Arabic Sentiment Classification

We apply a binary sentiment classification for different corpora from two different domains: reviews and tweets. We run the experiments on the LABR book reviews dataset (Aly and Atiya, 2013) which consists of over 63,000 reviews downloaded from Goodreads[15] in 2013, Arabic Sentiment Tweets Dataset (ASTD) (Nabil et al., 2015) which consists of over 10,000 Arabic tweets, Arabic Gold-Standard Twitter Sentiment Corpus (Refaee and Rieser, 2014) (GS-dataset) collected in 2014, Twitter data set for Arabic sentiment analysis collected by (Abdulla et al., 2013) that consists of 2000 labeled tweets, and also datasets collected by (ElSahar and El-Beltagy, 2015) that covers five domains:

1) *Hotel Reviews (HTL)*: For the hotels domain 15K Arabic reviews are scrapped from TripAdvisor[16].
2) *Attraction Reviews (ATT)*: Attraction reviews are scrapped from TripAdvisor.
3) *Restaurant Reviews (RES)*: Two sources are scrapped to cover restaurants reviews: Qaym[17] and TripAdvisor.
4) *Movie Reviews (MOV)*: collected from 1k movies in Elcinemas.com website, and consists of around 1.5K movies reviews.
5) *Product Reviews (PROD)*: For the Products domain, a dataset of 15K reviews is scraped from the Souq[18] website. The dataset includes reviews from Egypt, Saudi Arabia, and the United Arab Emirates.

We pre-processed all datasets by extracting positive and negative sentences, and splitting them to two main classes. The first one is the balanced class where the numbers of reviews and tweets are equal,

[15] www.goodreads.com.
[16] www.tripadvisor.com.
[17] www.Qaym.com
[18] www.souq.com

and the number is set to the minimum number of positive or negative sentences. The second one is the unbalanced class, where the numbers of reviews and tweets are unequal, and the numbers are set to the maximum numbers of positive and negative sentences, respectively. Datasets preparation statistics are shown in Table 4.

Datasets	Polarity	LABR	ASTD	Gold-Standard	Twitter Data set	ATT	HTL	MOV	PROD	RES
Balanced	Positive	8012	665	858	972	204	6192	384	807	9092
	Negative	8012	665	858	972	204	6192	384	807	9092
Unbalanced	Positive	42689	665	858	978	5242	27428	969	2882	25858
	Negative	8012	1496	1897	972	204	6192	384	807	9092

Table 4: Dataset preparation statistics

Models	LABR	ASTD	Gold-Standard	Twitter Data set	ATT	HTL	MOV	PROD	RES
CNN-balanced	**86.7**	75.9	73.8	86.3	74.2	**88.6**	**83.2**	**83.3**	77.1
CNN-unbalanced	**89.6**	79.07	75.8	85.01	96.2	**91.7**	**80.7**	**87.3**	78.5
(ElSahar and El-Beltagy, 2015)-Linear SVM	78.3	-	-	-	-	87.6	74.3	75.8	83.6
(Abdulla et al., 2013)-SVM	-	-	-	87.2	-	-	-	-	-
(Refaee and Rieser, 2014b)-SVM-BOW	-	-	87.74	-	-	-	-	-	-

Table 5: Comparison of existing methods with our CNN models on the same datasets. SVM (Abdulla et al., 2013), SVM+BOW SVM and Bag of Word from (Refaee and Rieser, 2014), Linear SVM (ElSahar and El-Beltagy, 2015).

Concerning Section 4 and Section 5.2 which evaluates the ability of a CNN based model to perform Arabic sentiment analysis. Table 5 presents results of our CNN models against other methods listing their best classification experimental results. Each cell has numbers that represent the accuracy of the evaluation performed on each dataset. There is a huge gap between different datasets in their sizes. A bigger dataset has better performance in terms of model accuracy. LABR dataset reaches 89.6 % when it has been trained in unbalanced form. The existence of sarcastic and dialectal Arabic (reviews/tweets) really could have a severe impact on the model accuracy. One problem is that there is not that much consecutive semantic/syntactic content in a single tweet. These datasets are examined in their balanced and unbalanced form. The results show that a CNN architecture with one non-static channel and one convolutional layer outperform the listed techniques on a scale 4 of 5 when using balanced datasets. By using all the data in an unbalanced form for the training and validation, the model is more accurately, showing a significant good performance compared to other models.

Additionally, the accuracy of the proposed model for Gold-Standard dataset is lower than that listed in (Refaee and Rieser, 2014), the reason is the used data set in our work is smaller than the original one, and the full dataset of (Refaee and Rieser, 2014) is not available for free. The CNN model provides a remarkable accuracy improvement over the Linear SVM approach used by (ElSahar and El-Beltagy, 2015), which is the previous best-reported result for their collected datasets. Initializing word vectors using our pre-trained vectors (CBOW58) gives a significant accuracy increase. Random initialization of words out of CBOW58 vocabulary can affect the vectors quality during the training. Although, exploring the effect of hyper-parameters for the CNN model still to be investigated in a detailed way, to observe the impact of these parameters on the network performance tackling different tasks.

6 Conclusions

In this study, we have introduced a crawling scheme for a large multi-domain corpus in order to build an Arabic word embeddings model. We have provided short practical and empirically informed procedures for exploring Arabic word embeddings and convolutional neural network (CNN) for sentiment classification.

The experiment results of building Arabic word embeddings from a web-crawled corpus illustrate that the performance increases along with the quality of the data. The results also show that high dimensionality vectors perform well on a large corpus. The results of CNN for sentiment datasets show that

initializing word vectors using a pre-trained word embeddings gain a remarkable performance. They also indicate that a bigger dataset usually has better performance regarding model accuracy.

Acknowledgments

The work described in this paper was supported in part by the National High-tech R&D Program of China (Grant No. 2015AA015403), by the National Natural Science Foundation of China (Grant No. 61601337), by the Fundamental Research Funds for the Central Universities (Grant Nos. 2015III015-B04, 2015IVA034 and 2016III011), by Nature Science Foundation of Hubei Province (Grant No. 2015CFA059), by Science & Technology Pillar Program of Hubei Province (Grant No. 2014BAA146), by Science and Technology Open Cooperation Program of Hannan Province (Grant No. 152106000048).

References

Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 587–591.

Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Nawaf A Abdulla, Nizar A Ahmed, Mohammad A Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6.

Mohamed A Aly and Amir F Atiya. 2013. Labr: A large scale arabic book reviews dataset. In *ACL (2)*, pages 494–498.

Emil St. Chifu, Tiberiu St. Letia, and Viorica R. Chifu. 2015. Unsupervised aspect level sentiment analysis using self-organizing maps. In *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 468–475.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Alaa El-Halees. 2011. Arabic opinion mining using combined classification approach. pages 1–8.

Hady ElSahar and Samhaa R El-Beltagy. 2015. Building large arabic multi-domain resources for sentiment analysis. In *Computational Linguistics and Intelligent Text Processing*, pages 23–34.

Nikos Engonopoulos, Angeliki Lazaridou, Georgios Paliouras, and Konstantinos Chandrinos. 2011. Els: a word-level method for entity-level sentiment analysis. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, pages 1–9.

Noura Farra, Elie Challita, Rawad Abou Assi, and Hazem Hajj. 2010. Sentence-level and document-level sentiment mining for arabic texts. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1114–1119.

Nizar Y Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Bassam Hammo, Hani Abu-Salem, and Steven Lytinen. 2002. Qarab: A question answering system to support the arabic language. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, pages 1–11.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, pages 1–18.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Association for Computational Linguistics*, pages 1113–1122.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, pages 1–6.

Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mahmoud Nabil, Mohamed Aly, and Amir F Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *EMNLP*, pages 2515–2519.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Jan Pomikálek. 2011. Removing boilerplate and duplicate content from web corpora. *Disertacnı práce, Masarykova univerzita, Fakulta informatiky*.

Eshrag Refaee and Verena Rieser. 2014. An arabic twitter corpus for subjectivity and sentiment analysis. In *LREC*, pages 2268–2273.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*, pages 151–161.

Vít Suchomel, Jan Pomikálek, et al. 2012. Efficient web crawling for large text corpora. In *Proceedings of the seventh Web as Corpus Workshop (WAC7)*, pages 39–43.

Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *EMNLP*, pages 1046–1056.

Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. Word representations in vector space and their applications for arabic. In *Computational Linguistics and Intelligent Text Processing*, pages 430–443.