Comparative study of word embeddings models and their usage in Arabic language applications

Preprint · November 2018

DOI: 10.13140/RG.2.2.33437.77282

2 authors:

Dima Suleiman
University of Jordan
27 PUBLICATIONS **95** CITATIONS
SEE PROFILE

Arafat Awajan
Princess Sumaya University for Technology
70 PUBLICATIONS **123** CITATIONS
SEE PROFILE

Comparative study of word embeddings models and their usage in Arabic language applications

Dima Suleiman
Computer Science Department, King Hussein Faculty of Computing Sciences
Princess Sumaya University for Technology
Amman, Jordan
Information Technology Department
The University of Jordan
Amman, Jordan
dimah_1999@yahoo.com

Arafat Awajan
Computer Science Department, King Hussein Faculty of Computing Sciences
Princess Sumaya University for Technology
Amman, Jordan
awajan@psut.edu.jo

***Abstract*—Word embeddings is the representation of the text using vectors such that the words that have similar syntax and semantic will have similar vector representation. Representing words using vectors is very crucial for most of natural language processing applications. In natural language when using neural network for processing, the words vectors will be fed as input to the network. In this paper, a comparative study of several word embeddings models is conducted including Glove and the two approaches of word2vec model called CBOW and Skip-gram. Furthermore, this study surveying most of the state-of-art of using word embeddings in Arabic language applications such as sentiment analysis, semantic similarity, short answer grading, information retrieval, paraphrase identification, plagiarism detection and Textual Entailment.**

Keywords: word embeddings; deep learning; sentiment analysis; word2vec; Glove; semantic similarity, CBOW, Skip-gram.

I. INTRODUCTION

Word embeddings is one of the important hypothesis representations used to represent words, phrases and sentences to be used in several natural language processing (NLP) applications [1]. Word embeddings is used to represent the words by low dimensional vectors representation, such that the syntax and semantic relationship between words can easily be measured. Furthermore, there are several models for generating word embeddings. In order to be useful, these models must be trained using very large corpus to determine the semantic relationship between words since the semantic similarity is crucial for several applications [2]–[5]. The similarity between words can be measured using cosine similarity, Euclidean distance and other techniques. Recently, two word embeddings models were proposed that played a significant role in variety of NLP applications called word2vec model [6] and Glove model [7]. In this paper, a survey study of word embeddings models including word2vec and Glove models were studied. Furthermore, this study covers the word embeddings usage in several Arabic NLP applications such as sentiment analysis, semantic similarity, text summarization, paraphrase detection, etc. This paper is organized as follows: Section 2 presents an overview of two word embeddings models. Section 3 covers word embeddings uses in Arabic NLP. Evaluation and pre-trained word embeddings models is explained in section 4. Finally, section 5 presents the discussions and conclusion.

II. OVERVIEW OF WORD EMBEDDINGS MODELS

Word embeddings is the distributional vector representation of the words introduced to represent their syntax and semantic. Word2vec and Glove word embeddings models were recently used in various natural language processing applications [6], [7]. In order to deal with various natural language problems and applications, the words in the text must be converted into vectors. Therefore, the semantic similarity between two words can be measured using cosine similarity, Euclidean distance and others [6]. One of the word vector representations that was previously used is called "one-hot" representation [8]. In one-hot, the number of dimensions of each vector is equal to the number of the vocabulary, thus if we have 10,000 vocabulary then we have 10,000 dimensions for each vector. Moreover, for each word vector, all the entries values will be set to "0" except one entry its value will be set to "1". In the vector, the index of the entry that its value is set one is equal to the position of the word in the vocabulary. For example, the vector of the fourth word in the vocabulary will contain "0" in all entries except in the fourth position the value will be "1". On the other hand, the "one-hot" representation has two shortcomings: the first one is that there is no syntax and semantic relationships between the words vectors, while the second shortcoming is the sparse space wasted. Thus, in order to solve the previous problems, recent word embeddings were proposed to consider syntax and semantic of the words. In addition to solving the sparse wasted problem of space by generating dense vectors. More details related to Glove and the two approaches of word2vec model called BOW and Skip-gram models will be explained in the following subsections.

2.1 Word2vec model

Word2vec model is a neural network that consists of one input layer, one output layer and one hidden layer. Hidden layer has no activation functions. In addition, the number of neurons in the hidden layer is equal to the dimensions of the vector representing the word in the word embeddings. The two problems of "one-hot" representation were overcome using word2vec model proposed by Mikolov [6]. Word2vec model utilizes large datasets in training in order to represent the

ISSN: 0857-1812

semantic and syntax of the words accurately, such that the similarity between words can be measured effectively. Word2vec consists of two approaches including continuous bag-of-words model (CBOW) and Skip-gram model where both of them achieved improvements in term of accuracy and computational cost. Accordingly, the processes of adding, subtracting or even finding the similarity between words were performed easily using the vectors generated from word2vec model. For example, if vector ("Queen") is subtracted from vector ("King") and the vector ("Man") is added to the result of subtraction then the result will be very close to vector ("Woman") [9]. In addition, algebraic operations such as subtracting two vectors are used to retrieve the similarity value between two words. For example, subtraction of vector ("Man") from vector ("Woman") or vice versa will produce small value representing the difference in gender between them.

Furthermore, the semantic similarity between words is needed in various natural language processing applications including, machine translation, text summarization, sentiment analysis and many others [10]. The two approaches of word2vec model which are CBOW and Skip-gram will be covered briefly in the following subsections. The two approaches use the same hyper parameters such as the window size and the vocabulary size. The window size represents the number of words in the context and denoted by *c* and the vocabulary size denoted by $|v|$.

2.1.1 Continuous Bag-of-Words Approach (CBOW)

Continuous bag-of-words approach uses log-linear classifier to classify the predicted middle word given the surrounding future and history words. The architecture of CBOW can be shown in Fig. 1. CBOW maximizes equation (1) [10] where *w(t)* represents the current word while the context words are represented using the following symbols{ *w(t-c)* , … ,*w(t-2)* , *w(t-1)* , *w(t+1)* , *w(t+2)* , … , *w(t+c)* }.

$$\frac{1}{|V|}\sum_{t=1}^{|V|}\log\left[p\left(wt\middle|\begin{matrix}wt-c,\dots,wt-2,wt-1,\\ wt+1,wt+2,\dots,wt+c\end{matrix}\right)\right] \qquad (1)$$

The size of the sliding window determines the number of the words in the context, such that if the size of the sliding window is five then the number of the context words is four and the value of *c* will be equal to four. Moreover, in order to predict a word, the preceding two words and the following two words, of the middle word to be predicted, must be considered in the context.

2.1.2 Continuous Skip-gram Approach (Skip-gram)

CBOW and Skip-gram approaches of word2vec model are very similar in the structure. However, there is only one difference between the two approaches. While in CBOW the input for neural network are the context words and the output is the middle word, in Skip-gram model the input is the current word (middle word) and the output are the context words. For example, if the window size is five then, in case of CBOW the input will be two history words and two future words while the output will be the middle word. In case of Skip-gram the opposite is true, the input will be the middle word and the output will contain two future words and two history words. Furthermore, if the number of context words or the window size increased, the quality of the model will increase. On the other hand, the computation complexity will increase. The architecture of the Skip-gram model can be shown in Fig. 1. The context words are represented using the following symbols { *w(t-c)* , … ,*w(t-2)* , *w(t-1)* , *w(t+1)* , *w(t+2)* , … , *w(t+c)* }, while *w(t)* represents the current word. As in CBOW the objective of the model is to maximize the log of the probability in equation (2) [10]. In addition, the sliding window is used to predict the next word. Finally, in both models the input for the neural network will be the one-hot representation of the words.

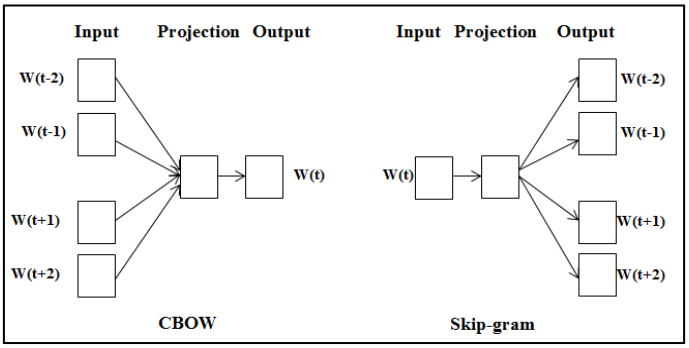

Fig. 1. CBOW and Skip-gram models architecture [6]

$$\frac{1}{|V|}\sum_{t=1}^{|V|}\sum_{j=t-c,j<>t}^{t+c}\log[p(wj|wt)] \qquad (2)$$

2.2 Global Vectors for Word Representation(Glove) Model

Glove is word embeddings model that was proposed by Pennington et al. [7]. Glove word embeddings stands for Global vector for word representation since it captures the statistics of the global corpus directly from the model, instead of depending on local context windows like word2vec model. Moreover, Glove uses the statistics efficiently by training the model on the global count of word-to-word co-occurrence. It consists of two steps: the first step is using the training corpus to get the co-occurrence matrix *X*. In matrix *X*, the symbol X_{ij} represents the frequency of the occurrence of words *i* and *j* together. The second step includes constructing the vectors by factorizing *X*.

As a conclusion, each one of the previous word embeddings models has its advantages and disadvantages. In

ISSN: 0857-1812

the case of training small dataset, the Skip-gram model is efficient and the same is true in case of infrequent words, while CBOW is efficient with frequent words [1]. On the other hand, one of the challenges of both CBOW and Skip-gram models was, learning the output vectors which considered as expensive and hard task. Therefore, in order to address the problem of learning the output vectors efficiently, two algorithms were proposed including Negative Sampling and Hierarchical Softmax [11]. Even more, the Negative Sampling only updates sample of the output vectors based on noise distribution. On the other side, the construction of Hierarchical Softmax is based on Huffman tree. Huffman is a binary tree that uses the frequencies of the words to present the words in a tree. After building the tree, the normalization will be used in each step from the root to the target word. Accordingly, for low dimensional vectors and corpus with more frequent words, it is better to use negative sampling while in case of infrequent words, hierarchical softmax is better [1]. As a conclusion, the use of word embeddings is highly dependent on the application, even though word2vec model using negative sampling is very efficient regardless of the application. Finally, based on small dimensional semantic space, word2vec produces best representation of the words vectors compared with Glove.

3 WORD EMBEDDINGS USES IN ARABIC NLP

Word embedding is used in several NLP applications since it considers both the syntax and semantic of the word. Thus, the semantic similarity between words, phrases and sentences can be calculated to improve the performance of the applications.

3.1 Sentiment Analysis Application

Sentiment analysis is NLP application that retrieves and analyzes information from reviews, opinions, and attitudes in order to improve intelligent and decision making process in various domains [12]. Sentiment classification classifies the text into several types including good, bad, positive, negative, etc. These several types of sentiment classification are called polarity [13]. The values of the polarity and sentiment classes vary from one research to another. Moreover, in addition to sentiment classes, there are sentiment expressions that may either be subjective or objective such as happy, sad, angry, etc. [14].

Word embeddings was used in Arabic sentiment analysis and achieved improvements [15]–[17]. Word embeddings facilitates the automatic extraction of the features from Arabic text such as tweets opinion mining, news articles and product reviews. Automatic feature extraction is very crucial since the manual extraction is time consuming especially for Arabic language which is full of morphemes that complicates the manual feature extraction process. However, in order to learn the word embeddings representation of the words, large Arabic corpus from several resources is used. The proposed approach in [17] discriminated between positive and negative in addition to neutral and subjective polarities. In their research, they used CBOW approach of word2vec model. The model was trained using three different datasets including ASTD, ArTwitter and QCRI on both Dialectal Arabic and Standard Arabic. Multiple similarities and analogy queries were used for the model evaluation. In addition, precision, recall, macro-accuracy and F-measure were used as evaluation metric.

On the other hand, Both CBOW and Skip-gram approaches of word2vec model were used in [15]. Word2vec approaches were trained using Abu El-Khair corpus composes from ten newspapers. The corpus was collected from eight several Arab counties and consists of over three millions words. In addition, the proposed approach used only the XML_UTF-8 formats of the corpus despite the fact the corpus was available in four formats. Furthermore, most of unwanted data, URL, IDs, non-Arabic words, digits, special characters were removed and some letters were normalized such as (آ ، إ ، أ). The experiments were conducted using several values of window size including 10, 30, 50, 100, and 200. In order to test the approaches, two words sentiments expressing (positive and negative) which are "good" "جيد" and "bad" "سيء" were used. This was achieved by selecting similar words related to them in meaning. Furthermore, spelling the letter Hamza "ء" in word "سيء" may create a challenge since many people spelling it wrong. As a result, neither using 10 nor 300 word dimensions are reasonable. Using 10 dimensions for representing each word may make the model considers words to be similar to words "good" and "bad" without having the same meaning. On the opposite side, representing a word using 300 dimensions, may allow the words that have opposite meaning of words "good" and "bad" to appear within the highest top ten similar words. The entire experiments were conducted using subset of health service Arabic tweet datasets which consists of 1398 negative tweets and 628 positive tweets. Even more, three annotators annotated that datasets where three annotators agreed on using 502 positive tweets and 1230 negative tweets for performing experiments. Several machine learning algorithms in addition to using convolutional neural network were used to expand the vocabulary. Finally, the proposed approach increased the accuracy to 0.92 compared with previous works.

Even more, two types of distributed embeddings representation including word embeddings and document embeddings were used in analyzing Arabic sentiment [16]. Two word embeddings were introduced, word2vec and glove models while doc2vec was used for document embeddings. Furthermore, the proposed approach consisted of two steps; the first one was the pre-processing of input text from linguistics perspective while the second step was the prediction of the polarity of the input text. Moreover, two classifiers were trained including logistic regression and multilayer perceptron. On the other hand, the embeddings vectors resulted from learning the paragraph were fed into the classifiers as input. The entire experiments included two tasks: the first one was the binary classification that classified the sentiment into "positive" and "negative" polarity. The second task was to use five-class classifications including "very negative", "negative", "neutral", "positive" and "very

ISSN: 0857-1812

positive“ polarities. Moreover, freely available dataset called LABR was used in experiments, which consists of 63257 book reviews. Finally, the light stemming was used and improved its importance in the classification results.

3.2 Semantic Similarity Application

The semantic and syntactic words representation using multidimensional vectors was proposed in [18] in order to measure the semantic similarity of short Arabic sentences. The CBOW model proposed by Zahran et al. word embeddings model was used to represent the words. Furthermore, 5.8 billion words from several resources were exploited for training including Arabic Wikipedia, Arabase, OSAC, and other datasets. In order to measure the semantic similarity among Arabic sentences, three methods were used. The first method was to sum the vectors of the sentence words, and then saving the results in a new vector where the similarity between the new vectors was calculated using cosine similarity. The second method measured the weights of the words using Inverse Document Frequency IDF in order to distinguish between the documents based on infrequent occurrence of the terms. On the other hand, instead of considering the weights of the words, the last method considered the weights of each part of speech tagging of the words. The entire experiments were conducted in MSR-Video corpus which consists of 750 sentences in addition to using the accuracy as an evaluation metric. Moreover, the correlation between human judgment and the score of semantic similarity was calculated. As a result, using part of speech tagging and IDF outperformed the no-weight methods.

Two machine translation-based word embeddings models were proposed to measure the semantic cross language similarities between Arabic and English sentences [19]. Moreover, the proposed approach composed of three steps including translation, preprocessing and attribution of the semantic score. Google Translate API was used to translate from English sentences to Arabic sentences. In addition, two proposed word embeddings were used to measure the similarity of Arabic sentences. The first word embeddings called Weighting Aligned Words (W-AW) since it uses words alignment and words weighting. On the other hand, the second word embeddings was called Bag-of-Words Alignment (BoW-A) which aligns the words to create the Bag-of-Words for them. As a result, the vector represented each sentence was constructed. After that, the similarity between pair of sentences was measured by comparing their vectors. Furthermore, the proposed method assumed that not all words have the same contribution in the meaning of the sentences in order to improve the results of similarity. Thus, three weighting functions were exploited including IDF, POS and IDF-POS. The experiments were conducted using four datasets from SemEval-2017 STS task consists of 2412 pairs of sentences.

3.3 Short Answer Grading and Information Retreival Applications

Word embeddings was used in short answering grading by utilizing several sentence vector representations and various similarity measures [20]. The system output the value zero to represent wrong answer and five to represent excellent answer. The experiments of the proposed approach were conducted in four datasets including Texas computer science, Extended Texas computer science, Cairo University and SemEval 2013 datasets. Three types of the similarity measures were introduced including string similarity, knowledge-based similarity and corpus based similarity. In addition, word2vec and Glove word embeddings models were utilized. Furthermore, the proposed model consisted of three modules: preprocessing, similarity measures and scaling modules. Preprocessing was crucial and affected the system performance where it included lemmatization, stemming and stop words removal. Even more, the similarity module measured the similarity level between the students' answers and the model answers and output a value between zero and one. Finally, the scaling module took the output value from the similarity module and mapped it to a grade using support vector regression (SVR).

On the other hand, in information retrieval, the matching process must consider the terms that have similar semantic as matching even if they do not have the exact syntax matching. In their research [21], they proposed to utilize three neural word embeddings models in already existing information retrieval model. The three word embeddings models were CBOW, Skip-gram and Glove. On the other hand, the information retrieval models including BM25v model, language model and information-based models. In their research [21], the authors integrated the scoring function and the word similarity, where for a given query the set of all similar words and top related words were considered. Furthermore, semantic term matching constraints (SMTCs) was used to examine the proposed system which regularized the original query and their similar ones weights. In addition, the evaluation was performed between the proposed model and the semantic approaches based on Arabic WordNet (AWN) and three word embeddings based on information retrieval models. The experiments were conducted in TREC 2001/2002 datasets which consists of 75 topics. The datasets were preprocessed using several stemmers such as heavy stemming, Farasa stemming, light stemming and normalization. As a result, despite of the fact that all types of stemming improved the performance of information, Farasa stemmer outperformed all of them. Also, the three word embeddings models improved the accuracy but the differences between the models were not significant. On the other hand, the best results were achieved by integrating SPL model with the word embeddings model.

The effect of the model quality on two of NLP applications including grading of short answers and information retrieval was assessed in extrinsic evaluation [22]. Therefore, three word embeddings models were used to represent the words in Modern Standard Arabic (MSA) including Skip-gram, CBOW and Glove. The data were

ISSN: 0857-1812

collected from several Arabic resources like Arabic Wikipedia, Arabic Gigaword Corpus, etc. In addition, the collected data were combined, cleaned and normalized. Examples of preprocessing included removing diacritics, tags and replacing all numerical digits using the word "NUM".

In addition, based on threshold of the frequency of single unit words which was created from n-gram tokens, the short phrases were formed. Moreover, the number of words in the selected corpus was 5.8 billion. The test cases of Mikolov's analogy test [6] were manually translated from English to Arabic in order to test the quality of the generated vectors where the test set consisted of five and nine types of semantic and syntactic questions respectively. Furthermore, the accuracy and the coverage were used for evaluating the models. The accuracy measured how many of the selected test cases were correct while the coverage measured the rate of covered test cases by the models. The evaluation results showed that the performance of the models were efficient in case of unambiguous Arabic translation for the words that are frequent in the corpus. However, in the case of less frequent words and the words that have no direct translation, the performance became less. On the other hand, the rare use of diacritics in Arabic represented a problem of understanding the semantic of the words that have the same form but different meaning. Even more, the cosine similarities between Arabic and English vectors were minimized by building neural network to map the vectors. This model outperformed the standard word-to-word similarity which used mean square error regression neural.

3.4 Paraphrase Identification and plagiarism Detection Applications

Word vector representation and Frequency-Inverse Document Frequency TF-IDF techniques were combined to enhance the paraphrase identification [23]. Moreover, Skip-gram approach of word2vec word embeddings model was used since Skip-gram achieves efficient performance in semantic analysis. On the other hand, TF-IDF was used to identify the words in each sentence that are highly effective in the meaning. Furthermore, there are three phases of the proposed algorithm including pre-processing, feature extraction and paraphrase detection. Preprocessing was used to extract certain information. Feature extraction phase used TF-IDF to weight the features and utilized word embeddings to represent the words. Finally, the last phase was the paraphrase detection between the source and plagiarized documents sentences which was achieved by finding the similarity. Each sentence in the text will be mapped into single vector with multiple entries representing the words of the sentence. The value of each entry resulted from the word representation multiplied by the TF-IDF of the word related to that entry divided by the number of words in that sentence. Finally, each sentence single vector in the suspect text was compared with all sentences vectors in the source document using the cosine similarity. The entire experiments were carried out using OSAC corpus which consists of 22,429 documents covere several topics like Economics, History, Entertainments, etc. Finally, precision and recall were used as evaluation measures where the results were promising.

Another application used word2vec word embeddings representation was plagiarism detection [24]. Furthermore, CBOW approach of the word2vec model was used and trained using OSAC corpus. In order to measure the similarity between words, cosine similarity was used. Simple changes in the sentence meaning resulted in high similarity value approximated from 99% which provided high possibility of detecting plagiarism. Examples of such changes are changing the position of verbs and nouns, changing the order of words or even changing one word with another word that have the same meaning.

3.5 Textual Entailment Application

Another crucial application used distributed representation using word embeddings was the Arabic textual entailment [25]. In Arabic textual entailment the distributional representations and traditional features were employed. However, the proposed approach didn't depend on the external resources but it depended on extracting the semantic and syntactic relationship based on large corpus. A set of features was used to explore if H (Hypothesis) was entailed by T (Text). One of the features was the length of T and H where both of them may have the same length while in some cases H may be shorter than T. Another feature was the similarity score between H and T, also the similarity between the name entities. Furthermore, another feature was using cosine similarity to measure the similarity between the H and T word embeddings. The last feature was using inverse document frequency score.

Moreover, Skip-gram word embeddings model was exploited. The experiments were carried out using the ArbTE datasets where the proposed approach outperformed the previous work in term of accuracy.

4 EVALUATION AND PRE-TRAINED WORD EMBEDDINGS

Evaluating and testing the Arabic space vectors representation using intrinsic and extrinsic evaluation were accomplished in [22]. In intrinsic, the semantic and syntactic similarity of words was evaluated using semantic and syntactic benchmark dataset which was used to evaluate the models quality. However, extrinsic evaluated the word embeddings models through using it in NLP applications. On the other hand, according to the importance of using word embeddings in several NLP applications, pre-trained Arabic word embeddings models was proposed in [26] and called AraVec. AraVec is an open source project of pre-trained Arabic distributed words embeddings that enables Arabic researchers to use free and powerful model in their researches. The model was built using three different domains of Arabic contents including Tweets, article of Arabic Wikipedia and World Wide Web pages with total number of tokens exceeds 3,300,000,000 tokens. Even more, two different word

ISSN: 0857-1812

embeddings approaches were used including CBOW and Skip-gram with tuning of hyper parameters such as minimum word count and the size of the window. Moreover, the final results of word embeddings were highly affected by the quality and preprocessing of the text. Therefore, several preprocessing steps were used such as non-Arabic content filtering, normalization and content filtering that was X-Rated. In non-Arabic content filtering, there was a problem that the Arabic alphabets may overlap with other languages alphabets like Urdu and Persian. Thus, detecting Arabic alphabet was achieved using language detection Python libraries. On the other hand, diacritics were removed in normalization step and several characters were replaced using common characters. Furthermore, quantitative and qualitative methods were utilized for evaluating the pre-trained models. SemEval-2017 datasets were used to measure the textual similarity in order to get the baseline score which was considered reasonable. Therefore, this was achieved through multiplying each vector of the words of the snippet by the TF-IDF value. After that, the mean of the resulted vectors was calculated. The probability of textual similarity was estimated by calculation the cosine similarity between each two snippets vectors. Finally, tool15 was used to evaluate the models by comparing the results with the results of the other models. On the other side, the qualitative evaluation was used to evaluate the models using sentiment words and known name entity types. Moreover, subset of words was randomly selected from lexicon of sentiments and their vectors were taken from the word embeddings models. Furthermore, k-means was used for clustering the words in order to see if the words with the same polarity will be clustered together. Similarly, k-means clustering was used to cluster named entities related to four categories including Organization, Location, Person and Time/Date.

TABLEI Displays the summarization of word embeddings models used in several Arabic Natural Language processing applications.

5 Discussions and Conclusion

In this paper, a comparative study was performed about several word embeddings models and their usage in Arabic NLP applications. Word embeddings is exploited to convert words to vectors where the conversion of word to vector is very crucial for NLP applications especially when fed into deep learning algorithms. After surveying the related articles we concluded that, the most word embeddings models used in NLP are word2vec and Glove models. The evaluation of word embeddings models can be conducted using either intrinsic or extrinsic. Intrinsic uses the benchmark datasets in order to evaluate the quality of the model while extrinsic evaluate the model by evaluating its effect in NLP applications. After evaluation, in case of frequent words in datasets, CBOW is more efficient while with infrequent words and small datasets, Skip-gram is more efficient. However, learning the output vector efficiently in both approaches of word2vec was a challenge. Therefore, two approaches were used called Negative Sampling and Hierarchical softmax. As a conclusion, using negative sampling with word2vec model provided reasonable results in term of efficiency. On the other hand, word embeddings models are highly dependent on the applications that utilize them. Finally, word2vec produces best representation of the words vectors compared with Glove according to small dimensional semantic space. In addition, the word embeddings were used in various applications such as sentiment analysis, semantic similarity, plagiarism detection, paraphrase identification, information retrieval, short answer grading and Textual Entailment. In all cases, the training of the word embeddings model was conducted in large Arabic dataset such as OSAC, SemEval-2017, LABR, ArbTE, Abu El-Khair corpus, where most of them are freely available. On the other hand, some of researchers collected their own dataset from Twitter, Wikipedia and World Wide Web pages. Moreover, several data preprocessing were applied on the dataset, where the most common preprocessing are normalization, lemmatization, stemming, removing of stop words, diacritics, tags, URLs and punctuation marks in addition to normalizing the numbers and dates by replacing them with Num and Date tokens.

REFERENCES

[1] M. Naili, A. H. Chaibi, and H. H. Ben Ghezala, "Comparative study of word embedding methods in topic segmentation," *Procedia Computer Science*, vol. 112, pp. 340–349, 2017.

[2] D. Suleiman and A. Awajan, "Bag-of-concept based keyword extraction from Arabic documents," in *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, 2017, pp. 863–869.

[3] A. Awajan, "Keyword Extraction from Arabic Documents using Term Equivalence Classes," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 14, no. 2, pp. 1–18, Apr. 2015.

[4] D. Suleiman, A. Awajan, and W. Al Etaiwi, "The Use of Hidden Markov Model in Natural ARABIC Language Processing: a survey," *Procedia Computer Science*, vol. 113, pp. 240–247, 2017.

[5] A. Awajan, "Semantic similarity based approach for reducing Arabic texts dimensionality," *International Journal of Speech Technology*, vol. 19, no. 2, pp. 191–201, Jun. 2016.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Jan. 2013.

[7] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.

[8] R. Socher, "RECURSIVE DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING AND COMPUTER VISION," p. 204.

[9] T. Mikolov, W. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," p. 6.

[10] A. Mahdaouy, E. Gaussier, and S. Ouatik El Alaoui, Arabic Text Classification Based on Word and Document Embeddings. International Conference on Advanced Intelligent Systems and Informatics, 2016.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," p. 9.

[12] G. Beigi, X. Hu, R. Maciejewski, and H. Liu, "An Overview of Sentiment Analysis in Social Media and Its Applications in Disaster Relief," in *Sentiment Analysis and Ontology Engineering*, vol. 639, W. Pedrycz and S.-M. Chen, Eds. Cham: Springer International Publishing, 2016, pp. 313–340.

[13] V. S. Rajput and S. M. Dubey, "An Overview of Use of Natural Language Processing in Sentiment Analysis based on User Opinions," *International Journal of Advanced Research in Computer Science and Software Engineering*, p. 5, 2016.

ISSN: 0857-1812

TABLEI. AN OVERVIEW OF WORD EMBEDDINGS MODELS USED IN SEVERAL ARABIC NATURAL LANGUAGE PROCESSING APPLICATIONS

Ref	Year	Application	Word Embeddings	Dataset	Pre-processing
[22]	2015	Word Representations	Skip-gram, CBOW and Glove	Arabic Wikipedia Arabic Gigaword Corpus, etc	Normalization, such as removing diacritics, tags and replacing all numerical digits using the word "NUM"
[15]	2016	Sentiment Analysis	CBOW and Skip-gram	Abu El-Khair corpus (training the word2vec) Arabic tweet dataset which consisted of 1398 negative tweets and 628 positive tweets	Removing of special characters, any none Arabic words or digits, such as 1234 or Hindi digit used in Arabic, such as ١٢٣٤.
[16]	2017	Sentiment Analysis	word2vec , glove and doc2vec	LABR(sentiment) freely available 63257 book reviews	Light stemming
[17]	2016	Sentiment Analysis	CBOW	ASTD, ArTwitter and QCRI	Extracting the tokens from sentecne in order to join them again in certain order
[18]	2017	Semantic Similarity	CBOW	5.8 billion words training including Arabic Wikipedia, Arabase, OSAC, and others	Stop-word, punctuation marks, diacritics and non-letters removal. Normalizing letters and Normalizing numerical digits to the token "Num"
[19]	2018	Semantic Similarity	Weighting Aligned Words, Bag-of-Words Alignment	four datasets from SemEval-2017 STS task consisted of 2412 pairs of sentences	Tokenization, Removing punctuation marks, diacritics, and non-alphanumeric characters, normalizing characters, the stop words were not removed
[20]	2015	Short Answer Grading	word2vec and Glove	Texas computer science, Extended Texas computer science, Cairo University and SemEval 2013	Lemmatization, stemming and stop words removal
[21]	2018	information retrieval	CBOW, Skip-gram and Glove	TREC 2001/2002 data sets which consist of 75 topics	Heavy stemming, Farasa stemming, light stemming and normalization
[24]	2017	plagiarism detection	CBOW	OSAC	Without preprocessing
[23]	2016	Paraphrase Identification	Skip-gram	OSAC	Identify sentences using "." "،" "؛" "." "!" "؟". Identify words using spaces
[25]	2017	Arabic Textual Entailment	Skip-gram	ArbTE	SPLIT, Removing URLs and punctuation were removed and numbers and dates were normalized to Num and Date tokenization, Lemmatization and stemming they used MADAMIRA stop words removal
[26]	2017	Evaluation	CBOW and Skip-gram	Tweets, article of Arabic Wikipedia and World Wide Web pages with total number of tokens more than 3,300,000,000	Content filtering and non-Arabic content filtering, normalization

[14] M. Biltawi, W. Etaiwi, S. Tedmori, A. Hudaib, and A. Awajan, "Sentiment classification techniques for Arabic language: A survey," in *2016 7th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 2016, pp. 339–346.

[15] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving Sentiment Analysis in Arabic Using Word Representation," p. 6.

[16] A. Barhoumi, Y. Estève, C. Aloulou, and L. H. Belguith, "Document embeddings for Arabic Sentiment Analysis," p. 9.

[17] A. A. Altowayan and L. Tao, "Word embeddings for Arabic sentiment analysis," in *2016 IEEE International Conference on Big Data (Big Data)*, Washington DC,USA, 2016, pp. 3820–3825.

[18] E. M. B. Nagoudi and D. Schwab, "Semantic Similarity of Arabic Sentences with Word Embeddings," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, Valencia, Spain, 2017, pp. 18–24.

[19] E. M. B. Nagoudi, J. Ferrero, D. Schwab, and H. Cherroun, "Word Embedding-Based Approaches for Measuring Semantic Similarity of Arabic-English Sentences," in *Arabic Language Processing: From Theory to Practice*, vol. 782, A. Lachkar, K. Bouzoubaa, A. Mazroui, A. Hamdani, and A. Lekhouaja, Eds. Cham: Springer International Publishing, 2018, pp. 19–33.

[20] A. Magooda, M. A. Zahran, M. Rashwan, H. Raafat, and M. B. Fayek, "Vector Based Techniques for Short Answer Grading," p. 6.

[21] A. El Mahdaouy, S. O. El Alaoui, and E. Gaussier, "Improving Arabic information retrieval using word embedding similarities," *International Journal of Speech Technology*, vol. 21, no. 1, pp. 121–136, Mar. 2018.

[22] M. A. Zahran, A. Magooda, A. Y. Mahgoub, H. Raafat, M. Rashwan, and A. Atyia, "Word Representations in Vector Space and their Applications for Arabic," in Computational Linguistics and Intelligent Text Processing, vol. 9041, A. Gelbukh, Ed. Cham: Springer International Publishing, 2015, pp. 430–443.

[23] A. Mahmoud and M. Zrigui, "Semantic Similarity Analysis for Paraphrase Identification in Arabic Texts," p. 8.

[24] D. Suleiman, A. Awajan, and N. Al-Madi, "Deep Learning Based Technique for Plagiarism Detection in Arabic Texts," in 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, 2017, pp. 216–222.

[25] N. Almarwani and M. Diab, "Arabic Textual Entailment with Word Embeddings," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, Valencia, Spain, 2017, pp. 185–190.

[26] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256–265, 2017.

ISSN: 0857-1812