Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 340–349

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

Comparative study of word embedding methods in topic segmentation

Marwa Naili*, Anja Habacha Chaibi, Henda Hajjami Ben Ghezala

RIADI laboratory, National School of computer Science (ENSI),
University of Mannouba 2010, Tunisia

Abstract

The vector representations of words are very useful in different natural language processing tasks in order to capture the semantic meaning of words. In this context, the three known methods are: LSA, Word2Vec and GloVe. In this paper, these methods will be investigated in the field of topic segmentation for both languages Arabic and English. Moreover, Word2Vec is studied in depth by using different models and approximation algorithms. As results, we found out that LSA, Word2Vec and GloVe depend on the used language. However, Word2Vec presents the best word vector representation yet it depends on the choice of model.

Peer-review under responsibility of KES International

Keywords: Word embedding, LSA, Word2Vec, GloVe, Topic segmentation.

1. Introduction

One of the interesting trends in natural language processing is the use of word embedding. The aim of this latter is to build a low dimensional vector representation of word from a corpus of text. The main advantage of word embedding is that it allows to offer a more expressive and efficient representation by maintaining the contextual similarity of words and by building a low dimensional vectors. Recently, the two well known methods for producing word embedding models are Word2Vec [1] and Global Vectors GloVe [2]. These two methods have been drawing great attention and it has been reported to be the most efficient ones for learning vector representations of words [1,2]. For this reason, Word2Vec and Glove have been used in different natural language processing tasks such as Word Similarity [3]. However, it is difficult to choose one of these two methods. In fact, Pennington et al. [2] proved that GloVe is more efficient than Word2Vec. Furthermore, they proved that classical methods can be more useful than Word2Vec in particular the Latent Semantic Analysis (LSA). This technique is considered as one of the most influential early models for word embedding. According to Pennington et al. [2], this can be explained by the fact that Word2Vec learns low dimensional vectors from the start and it does not use all the information from the training corpus. Mitra [4] shared the same concern as Pennington et al. [2] with the following question: "What if I told you that everyone who uses Word2vec is throwing half the model away?". Furthermore, Altszyler et al. [5] proved that the performance of Word2Vec to detect semantic words relations decreases when the corpus size is reduced. Yet they proved that LSA is more stable and it is

* Corresponding author. Tel.: +216-50-765-809 ;
E-mail address: maroua.naili@riadi.rnu.tn

1877-0509
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.009

independent from the corpus size. Baroni et al.[6] proved that Word2Vec outperforms traditional distributional methods like pointwise mutual information (PMI). Likewise, Levy et al.[3] proved that Word2Vec (Skip Grams with Negative Sampling SGNS) outperforms GloVe in many tasks such as word similarity. To explain these results, they said that Pennington et al.[2] only used Google's analogies for the evaluation.
Hence, in this paper, we will study LSA, Word2Vec and GloVe to determine which one is the most efficient method for learning vector representation. Moreover, this study will be done in the field of topic segmentation. In fact, best to our knowledge, no one has used Word2vec or GloVe in this field. Furthermore, this paper exploits the bilingual aspect of these methods by focusing on two languages: English and Arabic.

This paper is organized as follows: Section 2 presents related works in the field of topic segmentation; Section 3 describes LSA, Word2Vec and GloVe; Section 4 presents the proposed topic segmenter; In section 5, experimental results and discussion are reported; The conclusion and future work are presented in section 6.

2. Overview on topic segmentation

Topic segmentation is the process of dividing a document into coherent segments, such as each segment deals with a specific topic. For the last years, many topic segmenters have been presented and they can be classified according to two approaches: endogenous and exogenous approaches. For the endogenous approach, the process of topic segmentation is based only on information within the documents to be segmented. Yet, for the exogenous approach, external resources can be used in the process of topic segmentation in order to add external knowledge. Most of the proposed topic segmenters are used for English language. The most known segmenters are TextTiling[7] and C99[8] which are considered as endogenous topic segmenters. These segmenters are based on lexical repetition. Other examples of endogenous topic segmenters are LCseg[9], F06[10] and TopicTiling[11]. LCseg[9] is a lexical cohesion segmenter which uses lexical chain repetitions to detect topic boundaries based on cohesion function. F06[10] and TopicTiling[11] are based on TextTiling[7]. One of the differences is that F06[10] measures the similarity between sentences with Dice metric and not Cosine metric like in TextTiling[7]. Yet TopicTiling[10] calculates the similarity between blocs based on topic vector representations, which are constructed by LDA model, and the cosine measure. On the other hand, several exogenous topic segmenters are proposed based on different external resources such as: LSA[12], co-occurrence network[10] and generative Bayesian model[13].
Yet for the Arabic language, there is a flagrant lack of research in this field. Only few works deal with endogenous topic segmentation such as the work of Habacha et al.[14]. They adapted C99 and TextTiling to Arabic by using Khoja Stemmer. On the other hand, only two works deal with exogenous topic segmentation[15,16]. Brants et al.[15] used the Probabilistic Latent Semantic Analysis PLSA to propose a new Arabic topic segmenter named TopSeg. Touir et al.[16] also proposed a new topic segmenter based on a list of connectors as an external resource.

Based on these related works in the field of topic segmentation, we notice that only classical methods are used such as LSA[12], PLSA[15] and LDA[10]. Yet, recently, other modern methods have been proposed and they have showed promising results such as Word2Vec and GloVe. We also can notice that there is a lack of exogenous topic segmenters for the Arabic Language. For these reasons, we will propose exogenous topic segmenters for both English and Arabic languages. As external resources, we will use LSA as a traditional method and Word2Vec and Glove as modern methods. Moreover, we will study in depth which of these methods offers the best word representations that help to detect the semantic meaning of words.

3. Overview on LSA Word2vec and Glove

A word is well described by its context. This idea presents the main principle of LSA, Word2Vec and GloVe. However, the process of each method is different from the other.

- **Latent Semantic Analysis:** LSA[17] is a powerful statistical technique. It is based on two main steps. The first step corresponds to the construction of a term-document matrix M. The size of M is $n * m$ where the rows correspond to m terms, the columns correspond to n document and $M[i, j]$ corresponds to the frequency of the

term i in the document j. The second step is the singular value decomposition where M will be decomposed, according to the equation 1, into three matrices: U, V^T which are two orthogonal matrices and S which is a diagonal matrix. Finally, based on equation 2, only the k largest singular values and their corresponding singular vectors from U and V^T will be used in order to reduce the semantic space which corresponds to M_k.

$$M = U * S * V^T \tag{1}$$

$$M_k = U_k * S_k * V_k^t \tag{2}$$

The LSA method is based on several parameters which are: local and global frequencies setting, local and global weighting functions and the dimension of the semantic space. Naili et al.[12] have conducted an empirical study of LSA parameters in the field of topic segmentation. As result, they proved that these parameters have an important impact on the quality of topic segmentation. Furthermore, the best choices are: local frequency=3, global frequency=1, local weighting function is TF, Global weighting function is IDF and the dimension of the reduced semantic space is equal to 70%. In this paper, we will use these same choices as Naili et al.[12] for LSA.

- **Word2Vectors:** Word2Vec is a shallow word embedding model proposed by Mikolov et al.[1]. The main principle of this method is to learn law dimensional vectors from the begging. In fact, it predicts words based on their context by using one of two distinct neural models: CBOW and Skip-Gram.
Continuous bag of words (CBOW) predicts a current word based on its context. This latter corresponds to the neighboring words in the window. In the process of CBOW, three layers are used. The input layer corresponds to the context. The hidden layer corresponds to the projection of each word from the input layer into the weight matrix which is projected into the third layer which is the output layer. The final step of this model is the comparison between its output and the word itself in order to correct its representation based on the back propagation of the error gradient. Thus, the purpose of CBOW neural network is to maximize the following equation 3:

$$\frac{1}{V}\sum_{t=1}^{V}\log p(m_t|m_{t-\frac{c}{2}}...m_{t+\frac{c}{2}}) \tag{3}$$

where V corresponds to vocabulary size, c corresponds to the window size of each word.
Skip-Gram: it is the opposite of the CBOW models. In fact, the input layer corresponds to the target word and the output layer corresponds to the context. Thus, Skip-Gram seeks the prediction of the context given a word instead of the prediction of a word given its context like CBOW. The final step of Skip-Gram is the comparison between its output and each word of the context in order to correct its representation based on the back propagation of the error gradient. In fact, it seeks the maximization of the following equation 4:

$$\frac{1}{V}\sum_{t=1}^{V}\sum_{j=t-c,j\neq t}^{t+c}\log p(m_j|m_t) \tag{4}$$

where V corresponds to vocabulary size, c corresponds to the window size of each word.
According to Mikolov et al.[1], each one of these models has its own advantage. As example, Skip-Gram is more efficient with small training data. Moreover, infrequent words are well presented. On the other hand, CBOW is faster and works well with frequent words. However, learning the output vectors of CBOW and Skip-Gram represents one of the major limits of these two models. In fact, learning the output vectors can be a hard and an expensive task. In this context, two algorithms can be used to address this problem. The first algorithm is the Negative Sampling algorithm. The main idea of this algorithm is to limit the number of output vectors that need to be updating. Thus, only a sample of these vectors is updated based on a noise distribution. This latter is a probabilistic distribution which is used in the sampling process. The second algorithm is the Hierarchical Softmax. This algorithm is based on Huffman tree. In fact, it is a binary tree that presents all terms based on their frequencies. Then each step from the root to the target is normalized. According to Mikolov et al.[1], each algorithm is better than the other according to the training data. For example, negative sampling is more efficient with low dimensional vectors and it works better with frequent words. Yet, hierarchical softmax is better with infrequent words. As conclusion, using Word2Vec can be a hard and a complicate task considering the different models (CBOW and Skip-Gram) and the used algorithms for training data (negative sampling and

hierarchical softmax). Thus in this paper, we will investigate the performance of these different models and algorithms to determine which ones are more efficient in the domain of topic segmentation.

- **Global Vectors:** GolVe is one of the most known methods for learning word representations proposed by Pennington et al.[2]. It is based on word occurrences in a textual corpus. In fact, it is based on two main steps. The first one is the construction of a co-occurrence matrix X from a training corpus where:
  X_{ij} is the frequency of the word i co-occurring with the word j
  $X_i j = \sum_k^V X_{ik}$ is the total number of occurrences of the word i in the corpus (V corresponds to the size of the corpus)
  The second step is the factorization of X in order to get vectors. In fact, Pennington et al.[2] showed that, compared to raw probabilities, ratios help to reduce noise by identifying relevant words form irrelevant words. For this reason, they used the following general model (equation 5):

$$F(w_i - w_j, \tilde{w_k}) = \frac{P_{ik}}{P_{jk}} \tag{5}$$

  where w_i, w_j and w_k are three words' vector, $P_{ik} = X_{ik}/X_i$ is the probability of the word k occurring within the context of the word i, w are word vectors and $\tilde{w_k}$ are context word vectors.
  Yet to preserve the linearity and prevent mixing dimensions, Pennington et al.[2] used vector differences and the dot product of the arguments in equation 5 and it becomes:

$$F((w_i - w_j)^T \tilde{w_k}) = \frac{P_{ik}}{P_{jk}} \tag{6}$$

  However, the final model should be unchanged by exchanging $w- > \tilde{w}$ and $X- > X^T$. To resolve this symmetry, equation 6 becomes as follow:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) \tag{7}$$

  Finally, Pennington et al.[2] proposed a least squares objective function by equation 8 where $f(x)$ is a weighting function:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2 \tag{8}$$

4. Proposed topic segmenter

To study the performance of LSA, Word2Vec and GloVe in topic segmentation, we use ToSe-LSA which is proposed by Naili et al.[12]. This segmenter is based on LSA to construct a semantic space which is used as an external resource. Therefore, the main idea is to employ different semantic spaces in order to find which method is more effective for learning vector representation of words to capture their semantic meaning. As shown in Fig. 1, the topic segmenter ToSe is based on five steps. The first one is the pre-processing step which allows the following operations: detection of language, extraction of words, elimination of stop words, stemming the remaining words. The second step is the frequency dictionary construction. This dictionary is composed of vectors. Each vector is associated to a sentence and it is composed of: terms, their corresponding stems, frequencies and vectors. We note that these latter are extracted from the semantic space if it exists. If the term does not belong to the semantic space, it is presented by a null vector. The third step is the similarity matrix construction. In this step, we calculate the similarity between all pairs of all terms that belong to the semantic space. Then, we employ equation (9) to calculate the similarity between sentences. The forth step allows the construction of a rank matrix. We note that the rank presents the number of neighboring elements that belong to the rank mask and have lower values of similarity. The final step is dedicated to detect topic boundaries based on Reynar's algorithm.

$$Sim(S_1, S_2) = \frac{\sum_{t_i \in S_1 \cap SS} \sum_{t_j \in S_2 \cap SS} (Ft_i Ft_j cos(Vt_i, Vt_j))}{\sum_{t_i \in S_1} (Ft_i) \sum_{t_j \in S_2} (Ft_j)} \tag{9}$$

With S_1 and S_2 correspond to sentences 1 and 2; SS corresponds to the Semantic Space; Ft_i and Ft_j correspond to the frequency of terms t_i and t_j; Vt_i and Vt_j correspond to the vectors of t_i and t_j in SS.

The main originality of this segmenter is that it can be used for different languages. In this paper, we are limited to English and Arabic languages. For the English language, the stemming process is conducted by Porter Stemmer. Yet for the Arabic language, the stemming process is based on Light10 Stemmer. Moreover, based on the method that will be used to construct the semantic space, we propose three topic segmenters: ToSe-LSA (the original segmenter), ToSe-Word2Vec and ToSe-GloVe.

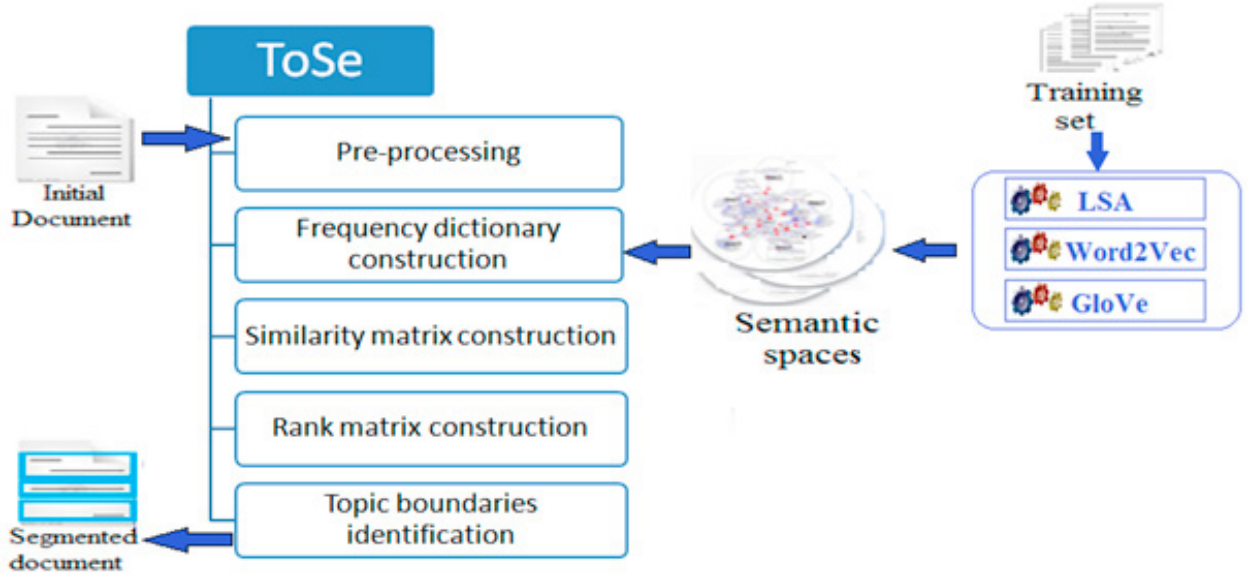

Fig. 1. Topic segmentation process

5. Evaluation and discussion

To determine which method of LSA, Word2Vec and GloVe is better to produce word representations, we conduct an evaluation based on two corpora. The first one is an English corpus which contains scientific articles from ACM digital library. These articles deal with different computer science topics. The second corpus is an Arabic benchmark. This latter is proposed by Al-Sulaiti and it deals with different topics such as: economy, politic, health and science. Based on these corpora, we construct two collections for each language: training collection and a test collection. The training collection is used to construct semantic spaces. The test collection is used in order to compare LSA, Word2Vec and GloVe in the field of topic segmentation and it contains artificial documents. These latter are constructed from a serial concatenation of documents. To report the evaluation results, we use the WindowDiff metric that measures the error rate by using a sliding window. Moreover, to conduct these studies we used : the R package LSA, Word2Vec tool proposed by Mikolov et al.[1] and the R text2vec package for GloVe.

The first part of this evaluation is dedicated to study Word2Vec in the domain of topic segmentation. Then we investigate the performance of LSA, Word2Vec and GloVe. Finally, we compare the proposed segmenters with existent topic segmenters.

5.1. Evaluation of Word2Vec parameters

Table 1 presents the average results of WindowDiff for Word2Vec based on different learning models and algorithms and the two languages English and Arabic. For the English language, there is an important difference between the performances of each combination of models and approximation algorithms. For CBOW, using the negative sampling is slightly better than the hierarchical softmax. Yet for Skip-Gram, hierarchical softmax is much better than the negative sampling algorithm. However, independent from the used algorithm for optimization, the performance of

CBOW is much better than Skip-Gram. For the Arabic language, the performance of each combination is very close. However, independent from the learning model, the negative sampling algorithm is more effective than hierarchical softmax algorithms. On the other hand, Skip-Gram is more efficient than CBOW.

Based on this evaluation, we can say that the choice of the learning model depends on the nature of the training data which confirms the statement that negative sampling works better with low dimensional vectors as claimed by Mikolov et al.[1]. In fact, for the English language, the training data is specific to the topic of computer science. In this case, we are dealing with frequent words. Thus, CBOW is more efficient than Skip-Gram for the English language. Yet for the Arabic language, the training data deals with different topics. In this case, we are dealing with infrequent words. For this reason Skip-Gram is more efficient than CBOW for the Arabic language. Another remark is the high performance of CBOW especially with the negative sampling algorithms for the English language. This can be explained by the fact that negative sampling work better with low dimensional vectors as claimed by Mikolov et al.[1]. In our case, the dimension of vectors is equal to 100 which is a low dimensional vector.
For the rest of this evaluation, we will use the best combination for each language: CBOW with negative sampling for the English language and Skip-Gram with negative sampling for the Arabic language.

Table 1. Average results of WindowDiff for Word2Vec.

Learning Models	Approximation algorithms	WindowDiff	
		English	Arabic
CBOW	Negative Sampling	5.79%	31.57%
	Hierarchical Softmax	6.93%	32.76%
Skip-Gram	Negative Sampling	11.54%	29.52%
	Hierarchical Softmax	7.67%	30.53%

5.2. Comparison between LSA, Word2Vec and GloVe

Fig. 2 shows the variation of the WindowDiff values of each segmenter (ToSe-LSA, ToSe-Wor2Vec and ToSe-GloVe) for the English language. As first remark, it is clear that ToSe-LSA has the biggest error rates. Yet the performance of ToSe-Word2Vec and ToSe-GloVe is very close. This result is confirmed by Table 2 which presents the average results of each segmenter and the dimension of the semantic space. According to Table 2, ToSe-Word2Vec outperforms the others and yet it has the smallest semantic space. On the other hand, ToSe-GloVe is slightly less efficient than ToSe-Word2Vec and it has the biggest semantic space. Finally, ToSe-LSA has the biggest average rate.

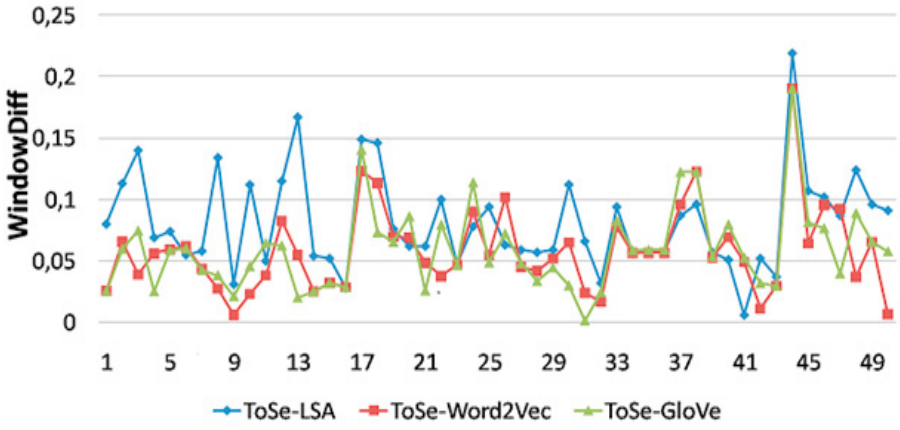

Fig. 2. WindowDiff curves of ToSe-LSA, ToSe-Word2Vec and ToSe-GloVe for the English language

Fig. 3 presents the variation of the WindowDiff values of each segmenter (ToSe-LSA, ToSe-Wor2Vec and ToSe-GloVe) for the Arabic language. We can notice that the variations of these segmenters are very close. Yet ToSe-GloVe has the biggest and smallest values of WindowDiff. These results are confirmed in Table 3 which describes the average

Table 2. Average results of WindowDiff of ToSe-LSA, ToSe-Word2Vec and ToSe-GloVe for the English language.

English topic segmenters	Semantic space dimension's	WindowDiff
ToSe-LSA	3997*100	8.14%
ToSe-Word2Vec	3455*100	5.79%
ToSe-GloVe	5954*100	5.94%

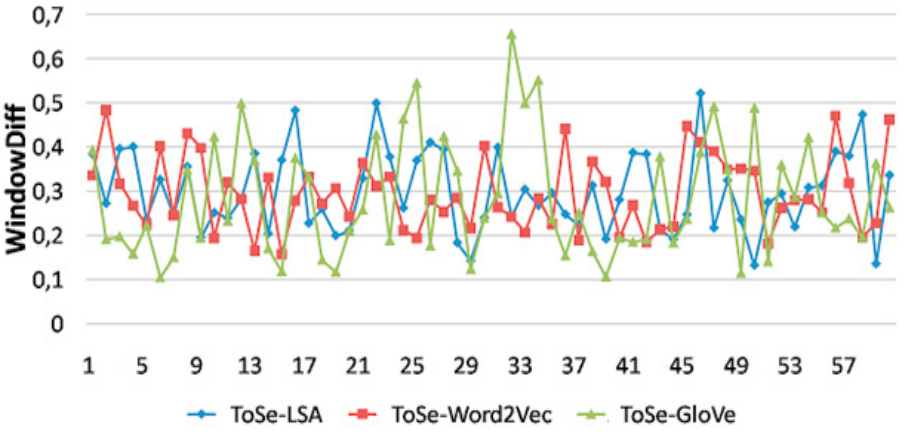

Fig. 3. WindowDiff curves of ToSe-LSA, ToSe-Word2Vec and ToSe-GloVe for the Arabic language.

results of each segmenter according to its semantic space dimension. Based on Table 3, ToSe-LSA has the biggest average error rate with the biggest semantic space. On the other hand, the performance of ToSe-Word2Vec is very close to ToSe-LSA yet the dimension of its semantic space is much smaller. Finally the highest quality of topic segmentation is offered by ToSe-GloVe.

Table 3. Average results of WindowDiff of ToSe-LSA, ToSe-Word2Vec and ToSe-GloVe for the Arabic language.

Arabic topic segmenters	Semantic space dimension's	WindowDiff
ToSe-LSA	35790*200	29.9%
ToSe-Word2Vec	4581*200	29.52%
ToSe-GloVe	5454*200	28.43%

Based on this evaluation, LSA is the less effective method for learning vector representations for both languages Arabic and English. On the other hand, we can say that Word2Vec offers the best vector representations of words for the English language. With these vectors, the quality of topic segmentation increased. Yet for the Arabic language, GloVe is slightly more effective than Word2Vec. However, we can say that, independent from the used language, Word2Vec outperforms GloVe. In fact, the main advantage of Word2Vec is that with a small semantic space, the semantic meaning of words is successfully detected. On the other hand, the performance of GloVe is similar to Word2Vec but with a much bigger semantic space. Thus, to detect the semantic meaning of words, GloVe needs more information than Word2Vec. This can explain the high performance of GloVe for the Arabic language. In fact, the Arabic language is known by its wide variety of grammatical forms and its complex morphology. For this reason, to detect the semantic meaning of words, more information is needed for this language than the English language. As an example, we present portions of the constructed Arabic semantic spaces in Fig. 4 for each method (LSA, Word2Vec and GloVe). In these portions, we present the same seven words for each semantic space with their corresponding vectors. As shown in fig. 4, each method offers a different vector representation of words. Besides, three stems are driven from the same root: علمَاء (scientists), عَالِم (scientist, world, universe), علم (science, aware, flag, knowledge, teach). However, despite the fact that these stems are driven from the same root, each one can has a meaning. Moreover, each stem can has several meaning. Furthermore, if we calculate the similarity between the two stems عَالِم (scientist) and its plural form علمَاء (scientists) based on the cosine similarity between their corresponding vectors, we have: 0.393 based on LSA, 0.216 based on GloVe and 0.058 based on Word2Vec. In this example, LSA has detected the similarity

Portion of the constructed semantic space based on LSA

علماء **scientists**

0.655393 0.04909635 -0.004053995 -0.07429009 -0.04187484 ... 0.06194106 0.02102263 -0.21404 0.008145259 -0.007852157

عالم **scientist, world, universe**

13.48763 0.002163401 -0.5685304 1.635446 -0.1122248 ... 0.4732695 -0.2361259 0.1011295 -0.04410312 0.4732695

علم **science, aware, flag, knowledge, teach**

0.7386626 8.560872 -0.1792792 -1.141201 -0.5738579 ... 0.05044228 -0.3369546 -0.02022496 0.04251711 0.05044228

بيئ **environment**

0.2824346 -0.08186349 -0.0355529 -0.1933444 0.4347435 ... 0.2267693 s0.1284645 -0.08996381 0.01657486 -0.01346703

زراع **agriculture**

0.09942182 -0.3474069 -0.03241784 -0.8840967 0.7160103 ... -0.2618753 0.3624856 0.1724669 -0.05495267 -0.004328141

طب **medicine**

0.9153517 0.002150209 -0.01601269 -0.05675767 0.1451522 ... 0.08140253 0.2764252 -0.05549601 0.01074258 -0.02274761

مرض **disease, pleasing**

-0.2848984 4.750256 0.5068766 2.181779 1.117963 ... 0.2505128 0.03403633 0.2506133 0.1596935 -0.1860437

Portion of the constructed semantic space based on GloVe

علماء **scientists**

0.6043252 -0.1217516 -0.2597798 0.6687182 -0.03385435 ... -0.2372326 -0.08456135 -0.09681392 0.5370381 0.1741187

عالم **scientist, world, universe**

-0.5040116 -0.3725181 0.3385159 0.264054 -0.2798362 ... -0.05210053 -0.478907 -0.7024179 -0.3554947 0.2943045

علم **science, aware, flag, knowledge, teach**

-0.1718928 0.3269776 0.3826695 0.1263586 -0.2327155 ... 0.1424363 -0.137754 0.0339257 0.4253464 0.3387461

بيئ **environment**

0.03848034 0.002340687 -0.05108983 0.9283525 0.1245753 ... 0.6558648 -1.514748 0.06260147 -0.306283 0.226491

زراع **agriculture**

0.2447208 -0.1651075 -0.3147481 0.06469481 0.3904688 ... 0.6670953 0.5254076 0.2272161 -0.00597282 -0.1662852

طب **medicine**

-0.2985749 0.04293475 0.3484234 -0.2401427 -0.2155349 ... 0.4449667 0.5730591 -0.380407 0.7377837 0.161211

مرض **disease, pleasing**

0.2167584 0.4031798 0.6938679 0.549248 -0.09713426 ... -0.2220968 -0.4445484 0.2065444 0.5429626 0.053642

Portion of the constructed semantic space based on Word2Vec

علماء **scientists**

-0.067181 -0.018880 -0.096833 0.084557 -0.040788 ... 0.084929 -0.076090 -0.141542 0.150714 0.043517

عالم **scientist, world, universe**

0.156663 0.154261 0.012002 -0.056400 -0.039712 ... 0.046036 -0.085560 -0.068308 0.101477 -0.036430

علم **science, aware, flag, knowledge, teach**

0.001095 -0.067907 0.127607 -0.065524 0.010159 ... 0.032472 -0.122731 0.014329 0.057513 -0.040641

بيئ **environment**

0.043425 -0.090605 0.040043 0.026608 -0.053709 ... -0.079503 -0.072589 -0.075312 -0.099148 0.010922

زراع **agriculture**

-0.100243 0.049416 -0.037133 0.026383 -0.106549 ... -0.109962 -0.109985 -0.000541 -0.058471 -0.003991

طب **medicine**

-0.028759 -0.016265 0.025000 -0.022054 -0.051526 ... 0.038486 -0.035436 -0.133010 -0.013531 0.005958

مرض **disease, pleasing**

-0.143329 -0.067146 0.085772 -0.014821 0.019111 ... -0.055650 -0.116916 -0.017575 0.072467 -0.120436

Fig. 4. Portions of the constructed semantic spaces.

between these stems. Yet, if we calculate the similarity between طب (medicine) and مرض (disease), we have: -0.0002 based on LSA, 0.26 based on GloVe and -0.095 based on Word2Vec. In this second example, only GloVe have detected the similarity between these two stems which belong to the health topic. Thus, based on the complex nature and structure of the Arabic language and compared to other languages, it is difficult to detect the semantic meaning of words which also depends on the vector representation of words. As a consequence, the quality of topic segmentation depends on the used language. In fact, for English language the WindowDiff is less than 9% in contrast for Arabic language the WindowDiff is more than 28%.

If we compare our results with related works, some statements can be doubted. For example, Pennington et al.[2] argue that GloVe performs better than Word2Vec in every task (word analogies, word similarity and named entity recognition). Even Mitra[4] proved the same results. Yet, we share the same statement of Mikolov et al.[1] that Word2Vec outperforms other models. Yet, the performance of GloVe is comparable to Word2Vec and they can be considered as two powerful methods to learn word vector representations in the domain of topic segmentation. On the other hand, if we compare modern prediction-based embeddings (Word2Vec and GloVe) with traditional count-based methods (LSA), we share the same claim as Baroni et al.[6] that LSA is less efficient than the other models for synonym detection, concept categorization, selection preferences and analogy. Yet, Levy et al.[3] argue that LSA outperforms Word2Vec and GloVe for word similarity. Moreover, Pennington et al.[2] argue that LSA outperforms Word2Vec but not GloVe for word analogy and similarity and named entity recognition tasks. Thus, we can explain these different claims by the choice of the domain application that influences the performance of each method.

5.3. Comparison with related works in topic segmentation

To evaluate the performance of the proposed segmenters (ToSe-LSA, ToSA-Word2Vec and ToSe-GloVe), we conduct a comparison with related works in Tables 4 and 5. In fact, for each language (English and Arabic languages), we conduct an evaluation of endogenous and exogenous segmenters on the same corpus. For the English language (Table 4), we can state that exogenous topic segmenters (BSeg, ToSe-LSA, ToSA-Word2Vec and ToSe-GloVe) are more efficient than endogenous topic segmenters (C99, TextTiling, F06, LCseg and TopicTiling). This claim is explained by the fact that adding external knowledge enhances the quality of topic segmentation. Nevertheless, the question is which external resource that improves better the topic segmentation? In this study, we find out that ToSe-LSA which is based on a traditional count-based method has the biggest error rate. Yet the generative Bayesian model of Bseg[13], Word2Vec and GloVe are the most efficient methods to use in topic segmentation especially Word2Vec. This can be explained by the fact that probabilistic models and neural models can detect easily the semantic meaning of words which is not the case for traditional methods such as LSA. For the Arabic language (Table 5), we only compared our work with the work of Habacha et al.[14]. As shown in Table 5, we noticed that ArabTextTiling and ArabC99, which are based on endogenous approach, have the biggest error rate. Yet the performance of ToSe-LSA and ToSe-Word2Vec is similar. Moreover, ToSe-GloVe is the most efficient segmenter.

Table 4. Comparison with existent English topic segmenters.

Approach	English topic segmenter	WindowDiff
Endogenous	C99[8]	18.92%
	TextTiling[7]	28.99%
	F06[10]	79.22%
	LCseg[9]	13.32%
	TopicTiling[11]	30.46%
Exogenous	Bseg[13]	6.76%
	ToSe-LSA	8.14%
	ToSe-Word2Vec	5.79%
	ToSe-GloVe	5.94%

Table 5. Comparison with existent Arabic topic segmenters.

Approach	Arabic topic segmenter	WindowDiff
Endogenous	ArabC99[14]	36.10%
	ArabTextTiling[14]	82.5%
Exogenous	ToSe-LSA	29.9%
	ToSe-Word2Vec	29.52%
	ToSe-GloVe	28.43%

Based on this evaluation, we can conclude that exogenous topic segmenters are much way better than endogenous topic segmenters for both Arabic and English languages. This can be explained by the fact that adding external knowledge enhances the quality of topic segmentation. Furthermore, we notice that prediction-based embedding methods improve topic segmentation.

6. Conclusions

In this paper, we investigated topic segmentation by using word embedding as representational basis. For this reason we used the well known methods: LSA, Wor2Vec and GloVe. The aim of this study is to identify which method is more effective to learn word vector representations that provide the semantic meaning of words for both English and Arabic languages. Yet, compared to other methods, Word2Vec is the most complicate one because of its different models (CBOW and Skip-Gram) and approximation algorithms (negative sampling and hierarchical softmax). For this reason, we studied in depth Word2Vec by evaluating different combination of these models and algorithms in the domain of topic segmentation. As result, we showed that, independent from the used language, negative sampling is the most efficient algorithms. Yet the choice of learning models is more delicate because it depends on the nature of the training data. In fact, we found that CBOW is more efficient with frequent words. Yet Skip-Gram is more efficient with infrequent words. Based on these results, we compared Word2Vec to LSA and GloVe. As results, we showed that Word2Vec and GloVe are more effective than LSA for both languages. Moreover, compared to GloVe, Word2Vec presents the best word vector representations with a small dimensional semantic space. Besides, we showed that the quality of topic segmentation depends on the used language. In fact, for the Arabic language, the quality of topic segmentation decreases compared to the English language because of its high complexity. Finally, compared to existent topic segmenters, we proved that ToSe-Word2Vec and ToSe-GloVe provide a high quality of topic segmentation. To go further, we will investigate the performance of LSA, Word2Vec and GloVe in other fields such as topic analysis.

References

1. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, arXiv; 2013. p. 1301-3781.
2. Pennington J, Socher R, Manning CD. Glove: Global Vectors for Word Representation. In EMNLP. 2014; 14:1532-1543.
3. Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics, 2015; 3:211-225.
4. Mitra B. Vectorland: Brief Notes from Using Text Embeddings for Search, Search Solutions, Microsoft (Bing Sciences), 2015.
5. Altszyler E, Sigman M, and Slezak DF. Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. arXiv preprint arXiv:1610.01520; 2016.
6. Baroni M, Dinu G, Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In ACL; 2014; 1:238-247.
7. Hearst MA. TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics, 1997; 23(1):33-64.
8. Choi FYY. Advances in domain independent linear text, segmentation. Proceeding of NAACL, 2000;p.26-33.
9. Galley M, McKeown K, Fosler-Lussier E, Jing H. Discourse segmentation of multi-party conversation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics; 2003,1:562-569.
10. Ferret O. Improving text segmentation by combining endogenous and exogenous methods. International Conference of Recent Advances in Natural Language Processing (RANLP), Borovets, 2009; p.88-93
11. Reidl M, Beimann C. How text segmentation algorithms gain from topic models. NAACL-HLT Proceeding, Canada, 2012;p.553-557.
12. Naili M, Habacha AC, BenGhezala HH. Parameters driving effectiveness of LSA on topic segmentation, 17th International Conference on Intelligent Text Processing and Computational Linguistics, Springer LNCS Series, Lecture Notes in Computer Science, Konya, 2016a.
13. Eisenstein J, and Barzilay R. Bayesian unsupervised topic segmentation. EMNLP, 2008;p.334-343.
14. Habacha AC, Naili N, Sammoud S. Topic segmentation for textual document written in Arabic language. KES-2014, 18th Annual Conference, Procedia Computer Science, Gdynia, Poland; 2014:35:437-446.
15. Brants T, Chen F, Farahat A. Arabic Document Topic Analysis. LREC'02 Workshop on Arabic Language Resources and Evaluation. Las Palmas, Spain; 2002.
16. Touir AA, Makhtour H, and AlSanea W. Semantic-Based Segmentation of Arabic Texts. Inf. Tech. J., 2008;7(7):1009-1015.
17. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R, Indexing by Latent Semantic Analysis. the American Society for Information Science. 1990;41:391-407.