Journal of Theoretical and Applied Information Technology
15th November 2019. Vol.97. No 21
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ARABIC TEXT CLUSTERING BASED ON K-MEANS ALGORITHM WITH SEMANTIC WORD EMBEDDING

[1]HASNAA R. H. SOLIMAN, [2]MOHAMED GRIDA, [3]MOHAMED HASSAN

[1] Assistant lecturer, Faculty of Computers and Informatics, Zagazig University, Egypt.

[2] Assistant professor of Industrial Engineering, Faculty of Engineering, Zagazig University, Egypt.

[3] Head of Information System Department, Faculty of Computers and Informatics, Zagazig University.

E-mail: [1] hrsoliman@zu.edu.eg, [2] mogrida@zu.eg, [3] monirhm@yahoo.com.

ABSTRACT

With the massive growth of Arabic content on the web, clustering of the Arabic textual data into a small number of meaningful groups becomes an essential component in various information retrieval applications, such as recommender systems, sentiment analysis, question answering systems, and search engines. Clustering methods, which are traditionally based on bag of words (BOW) model for text representation, do not consider the order relationships between terms and may result in unsatisfactory clusters especially with complex languages as Arabic. This study introduces a model for enhancing the accuracy of Arabic document clusters by integrating the K-means clustering algorithm with embedding approaches, including Word to Vector (Word2Vec) as a representational basis instead of BOW to capture the semantic information between individual terms. The model performance in the clustering news dataset utilized in previous similar studies was investigated. Accordingly, it was concluded that combing embedding techniques with the k-means algorithm improves the various evaluation measures of clustering as purity, F-measure, and accuracy.

Keywords: *Arabic Text Clustering, Document Embeddings, Word Embeddings, Doc2vec, Word2Vec.*

1. INTRODUCTION

Recently, Arabic Natural Language Processing (ANLP) gained the attention of many researchers in the era of information explosion especially with the tremendous amount of Arabic internet users who resulted in the rapid evolution of Arabic content on the Web in an electronic form. Most of these contents are available in a high-dimensional textual form without labels or Meta semantic information. Generating such information is crucial to many information retrieval applications, such as recommender systems, sentiment analysis, question answering systems, and spam detection. Despite that, the literature deeply addressed the field of online text clustering, in which semantically related contents are grouped into meaningful clusters; few studies addressed it for the Arabic language.

The Arabic language is highly inflectional and morphologically rich. It uses different meanings for the same words, the absence of capital letters by which nouns can be recognized, different forms of letters, is written from right to left, and utilizes diacritics for reflecting word meaning. Therefore, it is not adequate to be accurately stemmed and normalized by standard methods [1]. Because stemming and normalization are crucial for effective clustering, traditional clustering techniques. Additionally, the literature indicated that clustering Arabic documents with traditional text clustering algorithms may lead to inaccurate clusters of semantically related documents.

Recently, the literature utilized distributed Word Embeddings (WEs) approach instead of traditional text representations (bag of words (BOW)) for enhancing quality of document clustering by representing English text as a vector in a multidimensional space by capturing semantic and syntactic relations between words from a huge amount of text [2]–[5]. To the best of our knowledge, the limited research that considered the WEs for the Arabic language did not address the clustering problem.

This paper attempted to find answers for important research questions, which include: (i) Does the integration of embedding approaches with clustering algorithms (K-means) enhance the accuracy of Arabic document clusters when compared with standard K-means with TFIDF weighting schema or with LDA? Putting into consideration orthographic variations and complex morphology of the Arabic language. (ii) Which embedding approach performs better Arabic language clustering?

The rest of this paper is organized as follows: the related work is discussed in section 2 followed by an overview of the embedding approaches in section 3. The proposed framework is introduced in section 4, while an experimental study is presented in section 5 before we conclude the paper in section 6.

2. RELATED WORKS

Improving the results of Arabic text clustering is an interesting topic that was addressed by several studies. Table 1 presents a simple comparison among recent researches conducted on document clustering for the Arabic language.

Froud, et al. [6] and Hussein, et al. [7] utilized Keyphrase-Based methods to address the high dimensionality problem of Arabic documents by representing them with their key-phrases extracted either through the Suffix Tree (ST) model or machine learning techniques. Abuaiadah [8] investigated that applying bisect k-means algorithm for clustering normalized Arabic news documents outperformed the Standard K-Means clustering algorithm using five commonly used similarity and distance functions. Daoud and Sallam [9] improved the initial selection of centroids of the K-means using Particle Swarm Optimization (PSO). Malik et al. [10] indicated that the hybrid PSO-k-means algorithms may produce inaccurate clustering results, especially with high dimensional Arabic data sets.

Bsoul and Mohd [11] and Ghanem and Ashour [12] evaluated the impact of different preprocessing techniques (such as stemming) on the performance of the traditional clustering algorithm for the Arabic language. They reported that despite the improvement of such techniques on clustering results, some of these techniques (like root-based stemmers and light stemming) may result in a large amount of noise in documents representations either by grouping non-semantically similar words into the same stem or failing to group semantically similar terms to the same root. Other researchers focused on utilizing topic modeling for enhancing document representation after extracting the main topics and eliminating the noise via LSA [13].

Kelaiaia and Merouani [14] conducted a comparative study for indicating the influence of probabilistic topic models as Latent Dirichlet Allocation (LDA) in enhancing the quality of clustering on the Arabic benchmark document collection and concluded that LDA outperforms K-means in most cases. On the other hand, Alhawarat and Hegazi [15] suggested that integrating LDA with the k-means clustering algorithm should enhance the clustering results over the simple k-means.

The literature indicates that neural embedding approaches have a significant impact in enhancing the performance of document clustering when combined with traditional clustering algorithms, e.g. Deep Embedding Clustering (DEC) introduced by Xie at al. [16]. Rahaman and Hosein [3] proposed a clustering method based on Gaussian word embeddings (word2gauss) that outperformed traditional k-means in purity, inverse purity, and entropy. Sato, et al. [4] presented a clustering approach based on the paragraph vector model [17] to represent phrases and documents. Xu, et al. introduced a proposed Self-Taught Convolutional neural network framework that is combined with a K-means algorithm to cluster the learned representations [5].

Besides, the literature of ANLP shows a significate interest in using word embeddings approaches in some ANLP applications other than clustering [18]. Embedding approaches were used for enhancing the performance of some ANLP applications such as Paraphrase Identification [19], Short Answer Grading [20], [21], Arabic Textual Entailment [22], and Arabic sentiment classification [23]–[29]. Accordingly, there is a research gap in the area of using wording embedding approaches for ANLP clustering applications.

3. NEURAL WORD EMBEDDINGS APPROACHES

One of the main challenges in any NLP application is the document representation model or the approach by which the features reflecting semantic information and category relations are extracted from the document content. Vector representation of words is a way by which a word or document is represented as a vector.

Table 1: Summarization of Recently published Researches on Arabic Document Clustering.

Reference	Year	Approach	Dataset	Clustering Evaluation Measure
Froud, et al. [6]	2013	A novel Keyphrases extraction approach with Agglomerative Hierarchical algorithm	12 categories- 278 Arabic documents	Purity, entropy
Hussein, et al. [7]	2016	Keyphrase-based Hierarchical Clustering	12 categories- 345 Arabic documents and	Purity, entropy
Sangaiah, et al. [30]	2018	improved clustering algorithms with dimensionality reduction (k-means, incremental k-means, Threshold + k-means)	Six Arabic datasets of different categories	F-measure, entropy
Abuaiadah [8]	2016	The bisect K-means clustering algorithm	nine categories- five versions of 300 Arabic documents	purity, entropy
Daoud and Sallam [9]	2017	merged k- means with Particle Swarm Optimization (PSO) algorithm	three different Arabic datasets(BBC, CNN ,and OSAC) with different categories	Precision, Recall, F-Measure and Accuracy
Malik, et al. [10]	2018	hybrid clustering approach (K-Mean, PSO-K-Mean and PCA-K-Mean)	Five different Arabic datasets from UCI Machine Learning Repository	Purity, Rand_index
Kelaiaia and Merouani [14]	2016	probabilistic topic models with Latent Dirichlet Allocation (LDA)	Four different Arabic datasets (CCA, Al Watan, BBC, and Osac) with different classes	Rand index, Jaccard index, F-measure and Entropy
Alhawarat and Hegazi [15]	2018	topic modelling (LDA)/k-means combined method	News Arabic dataset composed of five versions. 9 categories- 2700 documents	purity, precision, recall, F_measure, entropy, NMI, NVI, accuracy, Jaccard_index
AbuZeina [31]	2019	PCA dimension reduction method with k-means clustering algorithm	five categories -250 Arabic documents	accuracy

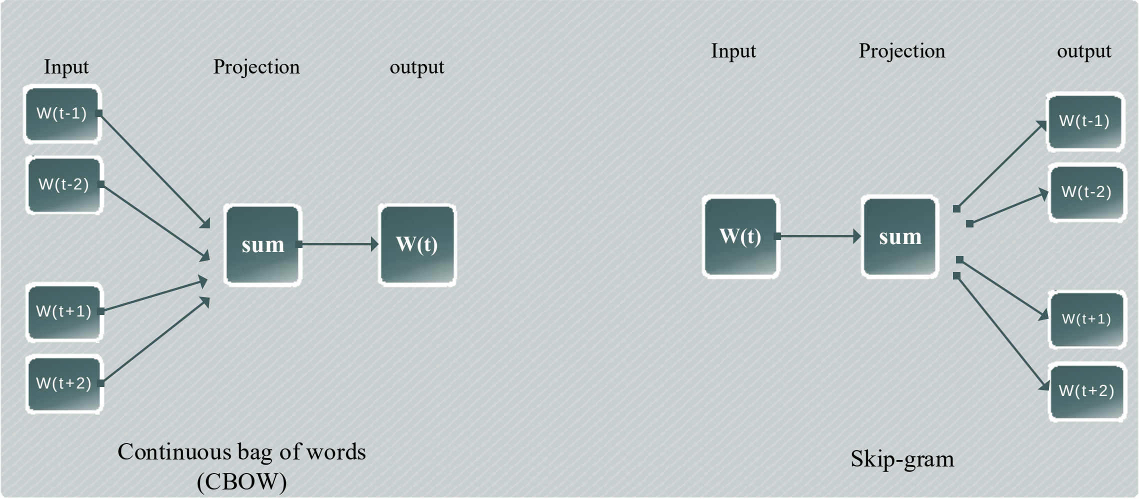

Figure 1: CBOW and Skip-gram models architecture (taken from [32]*).*

Mikolov, et al. [32] developed Word to Vector (Word2vec) as a predictive embedding model developed using a three-layer neural network to convert words into corresponding vectors, which are close to semantically similar vectors in an N-dimensional space. Word2Vec model comes in two flavors: the continuous bag of words (CBOW) and Skip-gram models [33]. CBOW model is similar to the feedforward neural network, in which each current word can be predicted only based on the window of its surrounding context words without considering the order of words. It uses the continuously distributed representation of the context by utilizing a simple neural architecture after removing the nonlinear hidden layer and the projection layer is shared for all words as shown in Figure 1. CBOW aims to maximize the objective function represented by Equation (1). Such an equation receives log probabilities of n context words, which are then summed for computing the probability of each target word.

$$\frac{1}{|V|}\sum_{t=1}^{|V|} log[P(W_t|W_{t-n},\dots,W_{t-1},W_{t+1},\dots W_{t-n})]. \qquad (1)$$

Where |V| is the vocabulary size and n is the number of context words in the sliding window.

On the other hand, the continuous skip-gram model uses a similar architecture by reversing the input and the output of the neural network.

The skip-gram model uses the current word to predict the surrounding context words in the window. Equation (2) calculates the objective function of the skip-gram [34]. Despite the slowness of the skip-gram is compared to CBOW, it does a better job for infrequent words [32].

$$\frac{1}{|V|}\sum_{t=1}^{|V|}\sum_{-n\le j\le n, j\ne 0} log[P(W_{t+j}|W_t)]. \qquad (2)$$

In 2014, Le and Mikolov proposed an unsupervised algorithm, named Doc2vec or paragraph level embedding, which is an adaptation of word2vec used for learning continuous distributed representations for pieces of texts (sentence, paragraphs or documents)[17]. Similar to the Word2Vec, Doc2Vec comes in two flavors: a distributed bag of words (DBOW) and paragraph vector distributed memory (PV-DM) as illustrated in Figure 2.

PV-DM works in the same way as the CBOW described previously except that the target word vectors are not summed but concatenated with the document tokens for predicting a context word given the concatenated document and word vectors. This model acts as a memory of the paragraph topic because of its ability to represents the missing information from the current context via paragraph vector. While PV-DBOW works in the same way as the skip-gram model except it utilized document vector representation as input and ignore the order of the words in the document.

Mahdaouy, et al. [34] have enhanced Arabic text classification using document embedding $(\vec{d})$, which was learned by averaging the word vectors of each document. According to such a model, each word vector $(\vec{w_i})$ is weighted based on its importance in each document by using the frequency of the target word $(X^d_{w_{id}})$ and document length (l_d)as depicted in Equation (3)

$$\vec{d} = \frac{X^d_{w_1}}{l_d}\cdot\vec{w_1} + \frac{X^d_{w_2}}{l_d}\cdot\vec{w_2} + \cdots + \frac{X^d_{w_{l_d}}}{l_d}\cdot\vec{w_{l_d}} \quad (3)$$

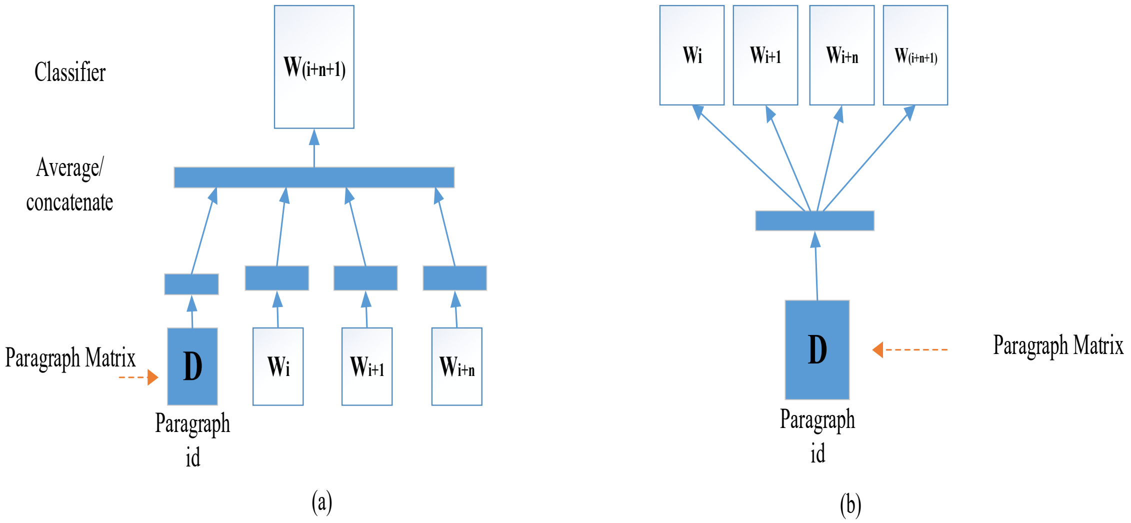

Figure 2: Main approaches in Paragraph embedding model: (a) document vectors distributed memory model (PV-DM); (b) Paragraph Vector distributed Bag of words (PV-DBOW).

4. PROPOSED FRAMEWORK

The proposed framework consists of four main stages: preprocessing, generation of a self-word embedding model (SWE), document vector representation, and finally document clustering with its validation as shown in Figure 3.

Due to the grammatical nature and the morphological richness and of the Arabic Language, preprocessing phase is a vital part in the proposed framework because of its ability to clean documents from data that have no distinctive meaning and insignificant to the analysis, such as such as usernames, hashtags, URLs, digits, all non-Arabic words, stop words, punctuation marks, repeated words and special symbols ($, %, &, |, _, -?). As letters in the Arabic language can be written in multiple forms with Arabic diacritics, the normalization process is performed by removing all Diacritics and mapping different forms of letters in a single form for standardization among the entire dataset. For instance, the different forms of the “aleph” letter (أ, آ, إ)are mapped to one form ”ا”, the form ”ta” letter ”ة” is mapped to ”ه ” , and the form of “ya” letter “ي” is mapped to “ى ”. After cleaning and normalization, each document is transformed into a vector of normalized tokens and a vocabulary set of size n contains all words extracted from the entire dataset is generated.

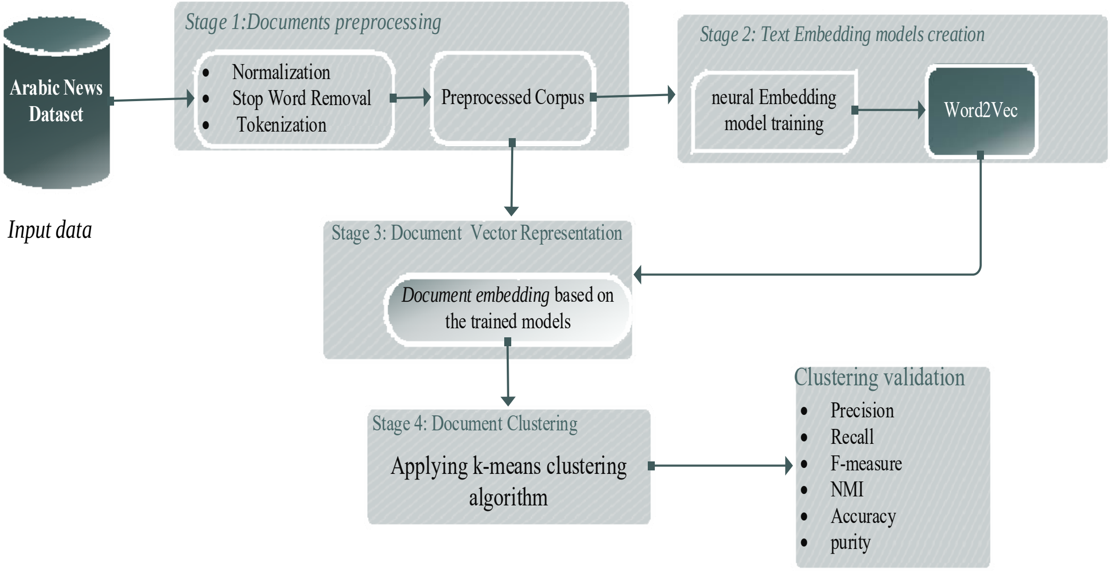
Arabic News Dataset
Input data
Stage 1:Documents preprocessing
Normalization
Stop Word Removal
Tokenization
Preprocessed Corpus
Stage 2: Text Embedding models creation
neural Embedding model training
Word2Vec
Stage 3: Document Vector Representation
Document embedding based on the trained models
Stage 4: Document Clustering
Applying k-means clustering algorithm
Clustering validation
Precision
Recall
F-measure
NMI
Accuracy
purity

Figure 3: Proposed Framework for Arabic Document Clustering.

After the entire dataset is preprocessed, it is utilized to construct the embedding model by extracting the semantic information between the words of the dataset. The embedding model is based on a modified version of the Sum Word Embedding (SWE) introduced by Mikolov, et al. [32]. A python algorithm is implemented to conduct the preprocessing of documents and the building of the embedding model. The model begins by creating a word representation vector for all words in each of the preprocessed documents by using a pre-trained SWE model; after that, the document vector is produced by summing up (or concatenating) all vector representations of its words.

This vector is mean normalized using Euclidean norm of the vector, which is defined by the square root of the dot product of the vector as shown in Equation (4) where n is the number of elements in vector (v):

$$\|v\|_2 = \sqrt{v.v} = \sqrt{v_1^2 + v_2^2 + v_3^2 + \cdots + v_n^2} = \sqrt{\sum_{i=1}^{n} v_i^2}\ . \qquad (4)$$

After transforming the entire dataset into the embedding representation, the k-means clustering algorithm is applied to groups of semantically related documents; the cosine distance is utilized for normalizing the angle between each pair of document embedding vectors (*v and w*) as sown in following Equation (5):

$$\cos\theta = \frac{V.W}{\|V\|_2\ \|W\|_2} = \frac{\sum_{i=1}^{n} V_i W_i}{\sqrt{\sum_{i=1}^{n} V_i^2}\sqrt{\sum_{i=1}^{n} W_i^2}}\ . \qquad (5)$$

5. EXPERIMENTAL STUDY

To verify the proposed model quality, an Arabic news dataset generated by Alhawarat and Hegazi [15] (which consists of 2700 Arabic documents of 9 categories) was utilized. The raw dataset was preprocessed for noise-cleaning and word tokenization before it was used for constructing and training the word embedding model.

Besides, design of experiment (DOE) method was applied as a statistical method for tuning and selecting the best values for three of input parameters that may affect the performance of word embedding model before comparing its performance with the state-of-the-art algorithms, including:

- Embedding vector size (size).
- Distance between each word and its surrounding words (window).
- Minimum allowed word count (min_count).

To indicate the influence of these parameters on the accuracy of the proposed model, the Minitab program was downloaded and Taguchi's experimental design technique was applied for optimizing these three hyper-parameters with five value levels as depicted in Table 2.

Table 2: Word2vec Model Parameters and Levels.

Input Parameters		Levels	values
Symbols	**Factor**		
A	**Size**	5	50, 100, 150, 200, 300
B	**Min_count**	5	1, 2, 3, 5, 10
C	**Window**	5	2, 3, 4, 5, 10

Table 3: Performance of Proposed Model with Different parameter settings.

Experiment number	*Facors*			*Average of numerical resulting metrics*						
	size	*min_count*	*window*	*precision*	*recall*	*f_measue*	*rand_index*	*Jaccard_Index*	*purity*	*NMI*
1	50	1	2	0.81659091	0.8186939	0.81764104	0.959543809	0.69171669	0.895925926	0.857122
2	50	2	3	0.80237866	0.804327	0.80335165	0.95637682	0.671479461	0.886296296	0.848552
3	50	3	4	0.86518225	0.8662069	0.86569425	0.970224912	0.763553924	0.926814815	0.886237
4	50	5	5	0.84081956	0.8425012	0.84165951	0.964882357	0.726743966	0.910962963	0.872644
5	50	10	10	0.79458512	0.7996561	0.79711252	0.954904231	0.66268362	0.875407407	0.851085
6	100	1	3	0.8224896	0.8244558	0.82347144	0.96083861	0.700742306	0.899259259	0.860824
7	100	2	4	0.86125094	0.8623892	0.86181967	0.969362645	0.759058363	0.923481481	0.885429
8	100	3	5	0.87381014	0.8749867	0.87439805	0.972152265	0.778653819	0.931037037	0.894515
9	100	5	10	0.82785739	0.8308956	0.82937359	0.962125286	0.70893769	0.90037037	0.868263
10	100	10	2	0.81882273	0.8208542	0.81983717	0.960032056	0.695505465	0.896518519	0.859573
11	150	1	4	0.84719802	0.8487452	0.84797089	0.966284742	0.737702579	0.914814815	0.876978
12	150	2	5	0.85609456	0.8572967	0.85669518	0.968225763	0.750351725	0.920666667	0.882047
13	150	3	10	0.9198751	0.921262	0.92056801	0.982387386	0.852893315	0.957703704	0.927703
14	150	5	2	0.80946624	0.8112424	0.81035331	0.95793394	0.68163811	0.891111111	0.852574
15	150	10	3	0.83025089	0.8320748	0.83116185	0.96255041	0.711351668	0.90437037	0.865598
16	200	1	5	0.8199953	0.8220533	0.82102297	0.960295308	0.696509699	0.897851852	0.859283
17	200	2	10	0.92320995	0.9241239	0.92366667	0.983078781	0.858771392	0.95962963	0.929342
18	200	3	2	0.79924134	0.8011758	0.80020738	0.955679552	0.667017358	0.88437037	0.846557
19	200	5	3	0.84332443	0.8446877	0.84400548	0.965408148	0.730713572	0.912666667	0.87335
20	200	10	4	0.87750275	0.8794466	0.87847357	0.973044063	0.784436075	0.93237037	0.899246
21	300	1	10	0.89596318	0.8967204	0.89634162	0.977022272	0.813537454	0.944444444	0.907848
22	300	2	2	0.81141877	0.813258	0.81233735	0.958373499	0.684198907	0.892296296	0.853936
23	300	3	3	0.84346454	0.8447467	0.84410509	0.96543241	0.730620407	0.913333333	0.872814
24	300	5	4	0.86491969	0.8661058	0.86551233	0.970181659	0.764053291	0.926148148	0.887556
25	300	10	5	0.82560676	0.8282824	0.85366005	0.961591097	0.706996614	0.898888889	0.866067

Using the experiment results in Table 3, the response table for the S/N ratio results of the Rand_index was estimated as shown in Table 4. According to such a table, the effect of three input parameters on rand_index can be analyzed using the delta statistics values, which are calculated for each parameter by subtracting the highest and the lowest average value. Besides, based on the delta values, the influence of each parameter is ranked. Results in Table 4 detected that the window parameter has the strongest effect on rand_index with a delta value of 0.1217, followed by min_count with a value of 0.0605, then size with a value of 0.0566.

Table 4: Response Table for S/N ratios (Rand_index).

Level	A	B	C
1	-0.3440	-0.3115	-0.3699
2	-0.3104	-0.2911	-0.3355
3	-0.2875	-0.2723	-0.2662
4	-0.2874	-0.3176	-0.3057
5	-0.2960	-0.3328	-0.2481
Delta	0.0566	0.0605	0.1217
Rank	3	2	1

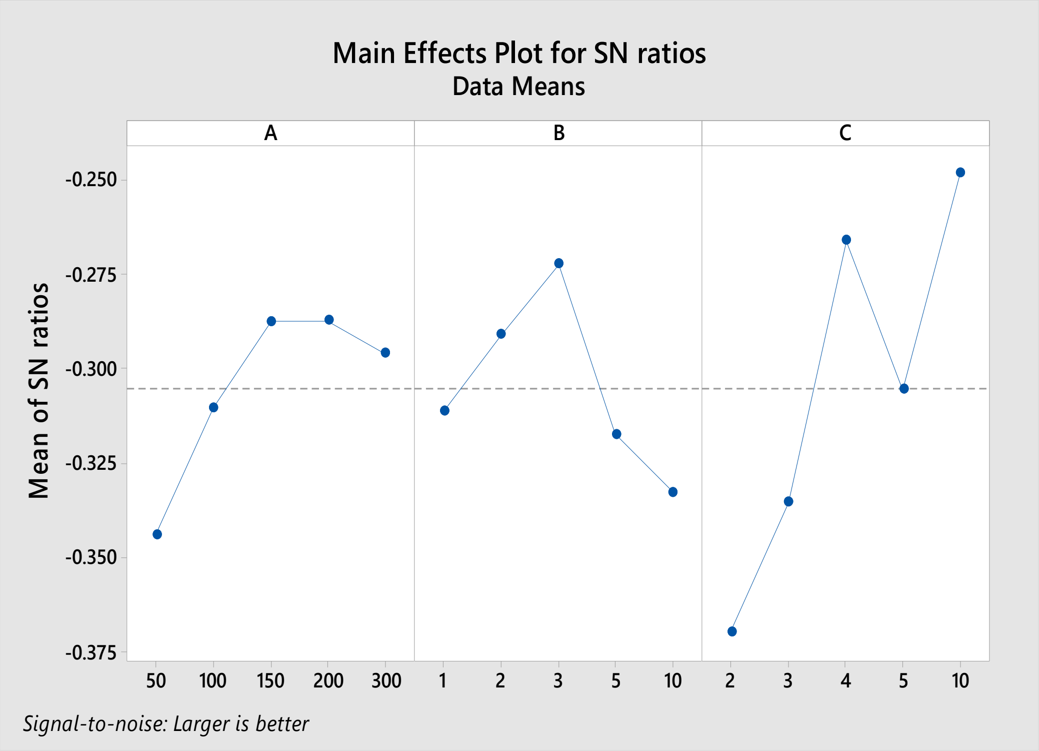

Figure 4: Main Effect Plots for S/N Ratios [Response: Rand_index].

Using these response values, Figure 4 represents the main effects of parameters for the rand_index was generated. According to such figure, it can be observed that rand_index increases with the increase in window value and decreases with the increase in size and min_count parameter values. Additionally, it can be perceived from Figure 4 that the third level of the size (A3), third level of the min_count (B3) and the fifth level of the window parameter (C5) result in the maximum value of the rand_index.

Furthermore, the S/N ratio analysis suggests that level values for A3, B3, and C5 are the optimum levels for the maximum rand_index of the proposed clustering framework. Additionally, it was observed that the medium values are better for Size and Min_count; while the largest value is better for the Window parameter.

In summary, based on the results of DOE in Table 3 and Figure 4, the chosen values of the proposed model parameters are Size = 150, Min_count = 3, and Window = 10.

Table 5: Confusion matrix for Clustering (taken from [15]*).*

Actual classes	**Obtained classes**	
	Same cluster	Different cluster
Similar documents	True Positive (TP)	False Negative (FN)
Different documents	False Positive (FP)	True Negative (TN)

To compare the effectiveness of the proposed model, the clustering results obtained by the proposed model are compared with others obtained by some of the most commonly used models for clustering Arabic texts, including tradition k-means algorithm (with normalized TFIDF weighting schema) and combined LDA/Kmeans[15]. Also, the obtained clustering results compared with the others obtained by the same proposed model however Doc2vec model introduced by [17] was utilized instead of the self Wor2vec model. For each model, the k-means algorithm was run for 20 times, each of them runs with 25 different initial centroids and the mean value of the clustering results was recorded for seven of clustering evaluation metrics.

Five of the commonly used evaluation metrics were calculated using the data illustrated in the confusion matrix presented in Table 5 and they were calculated using the Equations (6-10):

$$Recall = \frac{TP}{TP + FN}. \qquad (6)$$

$$Precision = \frac{TP}{TP + FP}. \qquad (7)$$

$$F\_measure = \frac{2 * Precision * Recall}{Precision + Recall}. \qquad (8)$$

$$Rand\_index\ (or\ Accuracy) = \frac{TP + TN}{TP + FP + TN + FN}. \qquad (9)$$

$$Jaccard\_Index = \frac{TP}{TP + FP + FN}. \qquad (10)$$

In addition to the above five metrics, the Normalized Mutual Information (NMI) and the purity were considered as well [15]. The NMI is a statistical measure for comparing the quality of different clustering results based on the mutual information I (x,y) shared between class (x) and cluster (y) labels in addition to their entropy H() and calculated using the Equation (11):

$$NMI(x;y)=\frac{I(x;y)}{\sqrt{H(x)*H(y)}}. \quad (11)$$

While purity [15] indicates whether the dominant class (w_k) of a cluster (c_j) represents all objects in that cluster or not by estimating the percentage of all objects of dominated classes for each cluster for the number of all objects (N) as shown in Equation (12):

$$purity=\frac{\sum_k \max_j |w_k \cap c_j|}{N} \quad (12)$$

Table 6: Comparison of Different Clustering Models using Different Matching-based Evaluation Measures.

Model	Recall	Precision	F-Measure	Accuracy	Jaccard-Index	Purity	NMI
Proposed model	0.920	0.921	0.921	0.982	0.853	0.958	0.928
Doc2vec	0.867	0.865	0.866	0.970	0.764	0.929	0.848
LDA- K-means	0.864	0.855	0.859	0.969	0.753	0.925	0.852
TFIDF K-means	0.790	0.557	0.653	0.906	0.486	0.783	0.773

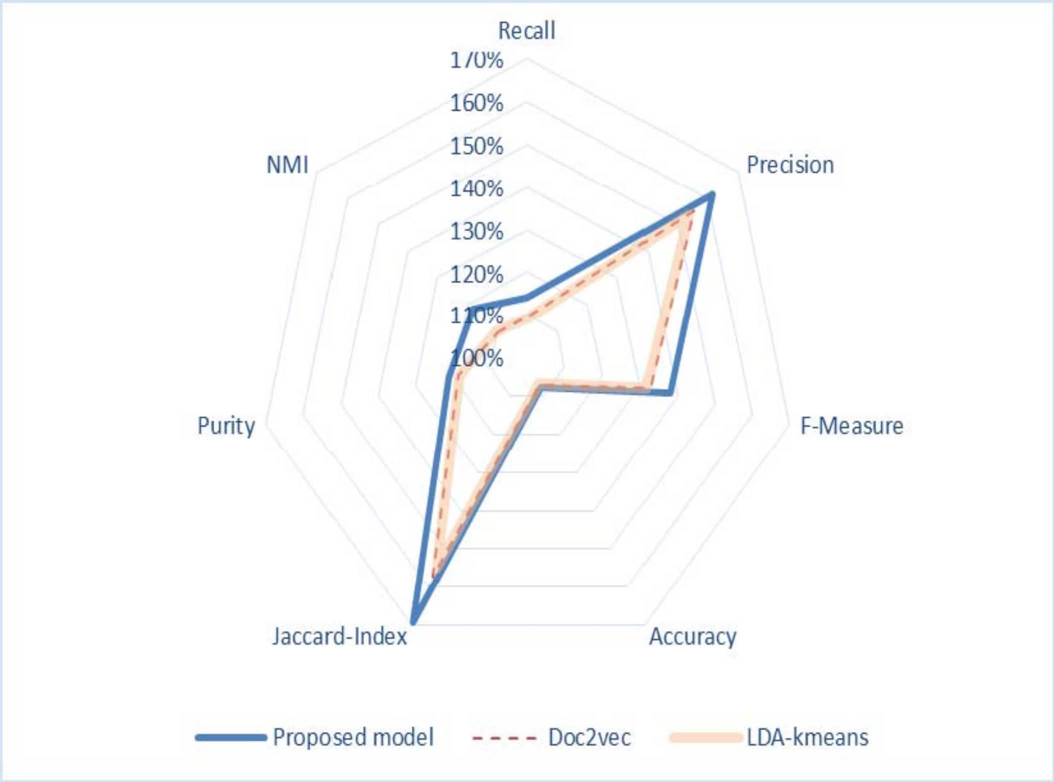

Figure 5: The Percentage of Improvement over the Basic Tfidf K-Means Model.

According to Table 6, it is obvious that combining k- means algorithm with either LDA, document or word embedding techniques results in a better quality of clustering over than the results obtained by traditional k-means algorithm, which based on TFIDF weighting schema for representing the documents as a bag of words. However, it can be observed that the proposed Word2Vec-K-means model outperforms all the other models in the seven calculated measures.

Figure 5 shows the percentage of improvement in each of the seven metrics for three compound models over the traditional Tfidf K-means. Precision and the Jaccard-index were greatly improved by combined models over the traditional Tfidf K-means. On the other hand, less improvement percentage was obtained in the Rand_index due to the high level of accuracy obtained by the basic Tfidf K-means. It is noteworthy that both of the LDA-K-means and the paragraph embedding model (Doc2vec) performed very similarly despite their different architectures.

Although the simple methodology being utilized in this study, it achieved a much better clustering results compared to prior works. Especially, applying word embedding first on the datasets served as both feature-selection and reduction method, which is very important in data mining applications, including clustering**.**

6. CONCLUSIONS AND FUTURE DIRECTIONS

Clustering Arabic documents with traditional text clustering algorithms is a challenging task due to several reasons, as depicted in the introduction section. In this paper, the quality of Arabic text clusters is dramatically improved by integrating the k-means clustering algorithm with Word2Vec embedding model for capturing the semantic information between the text words and utilizing the Euclidean distance for normalizing the length of obtained document vectors. The performance of the proposed model was tested against three of the common literature models using a dataset provided by of one the tested models. The resulting clusters were evaluated using seven clustering metrics and it was concluded that integrating embedding approaches with clustering algorithms (K-means) enhanced the accuracy of Arabic document clusters when compared with standard K-means with TFIDF weighting schema or with LDA with respect to all of the seven performance metrics as confirmed by the conducted results of experiments. Despite the progress achieved by the proposed Word2Vec- K-means model, there may some research worthy opportunities to improve the performance of Arabic clustering in the future through:

- Instead of relying on bag of words (BOW) for obtaining word embedding vectors, it may research worthy to consider creating these vectors by extracting n-grams words from the trained corpus.
- Considering neural embedding model such as FastText to generate word vectors may be an open research area as well.
- The proposed model may be extended by combining weighted tfidf word embedding with LDA for generating document embedding vectors that are feed into the clustering algorithm as an input.
- Hadoop (Map/Reduce) framework can be combined with the proposed model to speed up the preprocessing of large dataset and creating the Word2Vec.

REFRENCES

[1] S. A. Salloum, A. Q. Alhamad, M. Al-emran, and K. Shaalan, “A Survey of Arabic Text Mining,” in *Intelligent Natural Language Processing: Trends and Applications*, 2018, pp. 417–431.

[2] X. Guo, L. Gao, X. Liu, and J. Yin, “Improved Deep Embedded Clustering with Local Structure Preservation,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17) (IJCAI-17)*, 2017, pp. 1753–1759.

[3] I. Rahaman and P. Hosein, “Exploiting Gaussian Word Embeddings for Document Clustering,” in *Future Technologies Conference (FTC)*, 2017, no. November, pp. 1015–1018.

[4] M. Sato *et al.*, “Distributed Document and Phrase Co-embeddings for Descriptive Clustering,” in *Proceedings of the 15th*

Conference of the European Chapter of the Association for Computational Linguistics, 2017, vol. 1, pp. 991–1001.

[5] J. Xu *et al.*, “Self-Taught Convolutional Neural Networks for Short Text Clustering,” *Neural Networks*, vol. 88, pp. 22–31, Apr. 2017.

[6] H. Froud, I. Sahmoudi, and A. Lachkar, “An Efficient Approach to Improve Arabic Documents Clustering based on A New Keyphrases Extraction Algorithm,” in *Comput. Sci*, 2013, pp. 243–256.

[7] M. Hussein, A. Alsammak, and T. Elshishtawy, “Keyphrase-based Hierarchical Clustering for Arabic Documents,” in *Proceedings of the 10th International Conference on Informatics and Systems*, 2016, pp. 61–67.

[8] D. Abuaiadah, “Using Bisect K-means Clustering Technique in the Analysis of Arabic Documents,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 3, pp. 1–13, 2016.

[9] A. S. Daoud, A. Sallam, and M. E. Wheed, “Improving Arabic Document Clustering using K-means Algorithm and Particle Swarm Optimization,” in *IEEE Intelligent systems conference*, 2017, pp. 879–885.

[10] H. Malik, N. Laghari, D. M. Sangrasi, and Z. A. Dayo, “Comparative Analysis of Hybrid Clustering Algorithm on Different Dataset,” in *8th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2018, pp. 25–30.

[11] Q. W. Bsoul and M. Mohd, “Effect of ISRI Stemming on Similarity Measure for Arabic Document Clustering,” in *Asia Information Retrieval Symposium*, 2011, pp. 584–593.

[12] O. A. Ghanem and W. M. Ashour, “Stemming Effectiveness in Clustering of Arabic Documents,” *International Journal of Computer Applications*, vol. 49, no. 5, pp. 1–6, 2012.

[13] H. Froud, A. Lachkar, and S. A. Ouatik, “Arabic Text Summarization based on Latent Semantic Analysis to Enhance Arabic Documents Clustering,” in *arXiv preprint arXiv:1302.1612*, 2013.

[14] A. Kelaiaia and H. F. Merouani, “Clustering with Probabilistic Topic Models on Arabic Texts: A Comparative Study of LDA and K-means,” *The International Arab Journal of Information Technology*, vol. 13, no. 2, pp. 332–338, 2016.

[15] M. Alhawarat and M. Hegazi, “Revisiting K-Means and Topic Modeling, a Comparison Study to Cluster Arabic Documents,” *IEEE Access*, vol. 6, pp. 42740–42749, 2018.

[16] P. Xie and E. P. Xing, “Integrating Document Clustering and Topic Modeling,” in *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence*, 2013, pp. 694–703.

[17] Q. Le and T. Mikolov, “Distributed Representations of Sentences and Documents,” in *International conference on machine learning*, 2014, vol. 32, pp. 1188–1196.

[18] D. Suleiman and A. Awajan, “Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications,” in *International Arab Conference on Information Technology (ACIT)*, 2018, pp. 1–7.

[19] A. Mahmoud and M. Zrigui, “Semantic Similarity Analysis for Paraphrase Identification in Arabic Texts,” in *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, 2017, pp. 274–281.

[20] W. H. Gomaa and A. A. Fahmy, “Ans2vec : A Scoring System for Short Answers,” in *International Conference on Advanced Machine Learning Technologies and Applications. Springer, Cham*, 2019, vol. 1, pp. 586–595.

[21] A. Magooda, M. A. Zahran, M. Rashwan, H. Raafat, and M. B. Fayek, “Vector based Techniques for Short Answer Grading,” in *The Twenty-Ninth International Flairs Conference*, 2016, pp. 238–243.

[22] N. Almarwani and M. Diab, “Arabic Textual Entailment with Word Embeddings,” in *In Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 185–190.

[23] S. Al-azani and E.-S. M. El-Alfy, “Using Word Embedding and Ensemble Learning for Highly Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text,” in *Procedia Computer Science*, 2017, vol. 109, pp. 359–366.

[24] S. Al-azani and E.-S. M. El-Alfy, “Combining Emojis with Arabic Textual Features for Sentiment Classification,” in *International Conference on Information and Communication Systems (ICICS)*, 2018, pp.

139–144.

[25] M. Al-Smadi, M. Al-Ayyoub, Y. Jararweh, and O. Qawasmeh, “Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels’ reviews using morphological, syntactic and semantic features,” *Information Processing & Management*, Jan. 2018.

[26] A. M. Alayba, V. Palade, M. England, and R. Iqbal, “A Combined CNN and LSTM Model for Arabic Sentiment Analysis,” in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 2018, pp. 179–191.

[27] A. A. Altowayan and A. Elnagar, “Improving Arabic Sentiment Analysis with Sentiment-Specific Embeddings,” in *IEEE International Conference on Big Data (Big Data)*, 2017, pp. 4314–4320.

[28] A. A. Altowayan and L. Tao, “Word Embeddings for Arabic Sentiment Analysis,” in *IEEE International Conference on Big Data (Big Data)*, 2016, pp. 3820–3825.

[29] A. El-kilany, A. Azzam, and S. R. El-beltagy, “Using Deep Neural Networks for Extracting Sentiment Targets in Arabic Tweets,” in *Intelligent Natural Language Processing: Trends and Applications.*, 2018, pp. 3–15.

[30] A. K. Sangaiah, A. E. Fakhry, M. Abdel-Basset, and I. El-henawy, “Arabic text clustering using improved clustering algorithms with dimensionality reduction,” *Cluster Computing*, vol. 4, pp. 1–15, 2018.

[31] D. AbuZeina, “Exploring bigram character features for Arabic text clustering,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 4, pp. 3165–3179, 2019.

[32] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv preprint arXiv:1301.3781*, 2013.

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

[34] A. El Mahdaouy, E. Gaussier, and S. O. El Alaoui, “Arabic Text Classification Based on Word and Document Embeddings,” *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016, Advances in Intelligent Systems and Computing 533*, vol. 2, pp. 32–41, 2016.