The CSI Journal on
Computer Science and Engineering
Vol. 18, No. 1, 2020
Pages 28-35
Regular Research Paper

A Semantic-based Feature Extraction Method Using Categorical Clustering for Persian Document Classification

Saeedeh Davoudi and Sayeh Mirzaei

School of Engineering Science, College of Engineering,
University of Tehran, Tehran, Iran.

Abstract

Natural Language Processing (NLP) is one of the promising fields of artificial intelligence. Recently, a high volume of text data has been generated through the Internet. This kind of data is a valuable source of information that can be used in various fields such as information retrieval, recommender systems, etc. One practical task of text mining is document classification. In this paper, we mainly focus on Persian document classification. We introduce a new feature extraction approach derived from the combination of K-means clustering and Word2Vec to acquire semantically relevant and discriminant word representations. We call our proposed approach CC-Word2Vec (Categorical Clustering-Word2Vec) and use different classification models to compare the performance of our approach with other techniques like Term Frequency Inverse Document Frequency (TF-IDF), Word2Vec, and Latent Dirichlet Allocation (LDA) methods. Our proposed method resulted in an improvement in the obtained accuracy of all classifiers in comparison with other techniques.

Keywords: Persian document classification, TF-IDF, Word2Vec, CC-Word2Vec, MLP, GB, LDA, K-Means

1. Introduction

Nowadays, due to the advent of technology, a high volume of text data has been generated through web pages, social media, and other sources so that it is very time and energy-consuming to categorize data by humans [1]. Text data is unstructured data that can be analyzed using text mining methods [2]. The text mining framework, as it is mentioned in [3], has three sequential steps: text preprocessing, text representation, and knowledge discovery. According to [3], a classification task is a technique of knowledge discovery, and feature extraction methods are included in the text representation.

Text classification, more specifically, document classification is a supervised task in which the classifier is trained with some pre-categorized documents; Then, it will be expected that the classifier assigns an unseen document to one of the existing categories. There are many classification algorithms for text data including Support Vector Machines (SVM), Naïve Bayes [4], Logistic Regression, K-Nearest Neighbors (KNN) [5], and Neural Networks models [6] [7]. Furthermore, ensemble classifiers, made up of several classifiers, are another new technique for text classification. Gradient Boosting is a popular ensemble classifier for text classification [8].

Feature extraction is a crucial step that should be carried out before the classification task. It can enhance the prediction accuracy of the classification task through finding more discriminant representations or via dimensionality reduction techniques [9]. Although a lot of research has been done on English document classification, the developed methods might not necessarily perform well for Persian document classification [10]. TF-IDF is a commonly used feature extraction method. This method calculates the importance of a word in the whole dataset or corpus [11].

In reference [1], Farhoodi and Yari applied the TF-IDF technique to the Hamshahri dataset to realize which one of SVM or KNN is more efficient in the Persian document classification task. They reported that the KNN algorithm outperforms the SVM classifier. This efficiency can be improved by increasing the number of selected features and using a cosine similarity measure.

In [10], researchers classified Hamshahri news dataset using TF-IDF as the feature extraction method in the presence of entropy instead of stop word lists for removing stop words in the preprocessing step. They applied KNN and Nave Bayes classifiers to examine the accuracy of their method.

In reference [12], TF and TF-IDF methods are used as the feature extraction for classifying the Irna News Website. They operated Gaussian Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, and SVM classifiers to report the most accurate algorithm for Persian text classification. They reported that Multinomial Naïve Bayes classifier is the most accurate algorithm with micro F1-score 0.838530 using the TF-IDF method and removing stop words.

In reference [13], researchers proposed an ensemble classifier consisting of SVM, KNN, and MLP classifiers to categorize two datasets; “Routers” and “Hamshahri”. They also benefited from the TF-IDF method as the feature extraction method. This ensemble classifier resulted in better accuracy and efficiency for both datasets in comparison with SVM, KNN, and MLP, individually.

Jafari *et al.* [14] also worked on four categories of the new Hamshahri dataset. They used a representative vector to explore its impact on the accuracy of Persian document classification. They realized that a high value of precision and recall can be achieved by removing more extra words and inserting a few words into the representative vector.

In [15], topic models are utilized for Persian text classification and it is concluded that the use of topic models can lead to accuracy improvement with respect to the bag of words-based algorithms such as TF-IDF. Although the TF-IDF method is very common, it suffers from the lack of semantic relations between the words. As a result, another interesting method of feature extraction was presented to solve this problem. It is the Word2Vec model introduced by Tomas Mikolov *et al.* [16].

Researchers in reference [17] introduced a novel combined method benefited from Word2Vec and Latent Dirichlet Allocation (LDA) to extract the features of documents. This model considers both the relation between words and documents and the relationships between topics and documents. They tested their model with the 20Newsgroups dataset [18] and SVM classifier. They concluded that their model outperforms TF-IDF, Word2Vec, and LDA methods.

In the current paper, we use K-means clustering in combination with the Word2Vec embedding scheme. We apply K-means to the word vectors of each category individually and retrain the initial Word2Vec model using the vectors of each cluster, consecutively. This way, semantic relations of the words are more effectively incorporated in the word representations, hence, leading to more discriminant feature vectors for document classification.

2. Proposed Method

In this section, we explain our proposed method and the related concepts.

2.1. TF-IDF

According to [19], TF-IDF is the product of two values; Term Frequency (TF) and Inverse Document Frequency (IDF). TF denotes the frequency of a word in a document. IDF assigns a low weight to a highly frequent term and vice versa. We implemented this method using the Scikit-learn library in python [20]. Words with a frequency of less than 5 in the document will be removed. The TF-IDF formula is as follows:

$$TF = \frac{term\ frequency\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document} \quad (1)$$

$$IDF = log_e^{\frac{total\ number\ of\ documents}{The\ number\ of\ documents\ containing\ the\ term}} \quad (2)$$

$$TF - IDF = TF \times IDF \quad (3)$$

2.2. Word2Vec

Word2Vec, a two-layer neural network, is a word embedding technique. In this technique, the semantic relation between the words is considered. There are two types of this model: CBOW and Skip-gram. The first one predicts a word based on the surrounding words inside the window, and the latter one predicts the surrounding words of a specific word in the sentence. Here, we utilize the CBOW version of Word2Vec as it is faster than Skip-gram.

To implement a Word2Vec model, we used the Gensim library in python [18]. The model parameters are provided in Table I. The number of epochs determines how many times the learning process should iterate. The Word2Vec model does not consider words with a frequency of less than the minimum count threshold. In our model, words with a frequency of less than 5, cannot affect the performance of the model; so, they are removed. The window size parameter determines the number of words neighboring the current word to predict it. Vector dimension denotes the size of the output vector. Hence, in our model, each word is converted to a vector with 300 elements.

TABLE I. Word2Vec Parameters

Parameter	Value
Number of epochs	500
Minimum count	5
Window size	20
Vector dimension	300

2.3. K-Means Algorithm

K-Means is a well-known clustering algorithm. This algorithm aims to cluster data points to K (pre-defined) clusters. It starts with random initial centroids and repeats the following two steps at each iteration until the convergence occurs; 1) Each data point is assigned to the cluster corresponding to the nearest centroid among K centroids, 2) Each centroid is updated as the mean value of the corresponding cluster data points specified in the previous step. We use the Euclidean metric to compute the distance between the word vectors.

2.4. CC-Word2Vec

More accurate semantic relations can be discovered through clustering the corpus and retraining the Word2Vec model by clusters. Hence, we apply this approach to extract word embedding for Persian document classification. However, we do not cluster the whole words of the dataset; instead, we cluster words of each category in the training data by K-means algorithm into K clusters because each category consists of more semantic-related words, and this categorical clustering results in more reasonable clusters. After clustering of the first category, we retrain the Word2Vec model by passing each cluster to the model, then repeat this process for other categories.

Figure 1 illustrates the steps of the proposed approach called CC-Word2Vec (Categorical Clustering Word2Vec). For more clarity, consider we have five categories named A to E. First, we start with category A and cluster unique words in category A using the K-means algorithm. Now, we have K clusters for category A. In the final step, we retrain our previous Word2Vec model K times, each time using one of the K clusters of unique words in category A. This process will repeat for categories B to E again, until the final Wor2Vec model, called CC-Word2Vec, will be resulted. As a result, the initial Word2Vec model will be retrained for times in which K is the number of clusters in the K-means algorithm and N is the number of categories in our corpus.

2.5. Multi-Layer Perceptron

MLP is a non-linear classifier that consists of dense layers of neurons. The first layer is the input layer, and the last layer is the output layer. All layers between these two layers are considered as the hidden layers. The parameters of our MLP model are listed in Table II. As we have a multiclass classification task, Softmax should be utilized as the activation function of the output layer. In order to avoid overfitting issue, we apply the dropout technique. Towards this, we set the dropout parameter equal to 0.3, thus, 30% of neurons at each hidden layer will be removed randomly. All of the hyperparameters are selected empirically through searching among a limited number of values.

2.6. Gradient Boosting

GB classifier is an ensemble model composed of several classifiers. In other words, boosting algorithms combine weak learners to acquire a stronger model. The number of weak learners or regression trees is set to 100 and the maximum depth of each tree is 3 in our model.

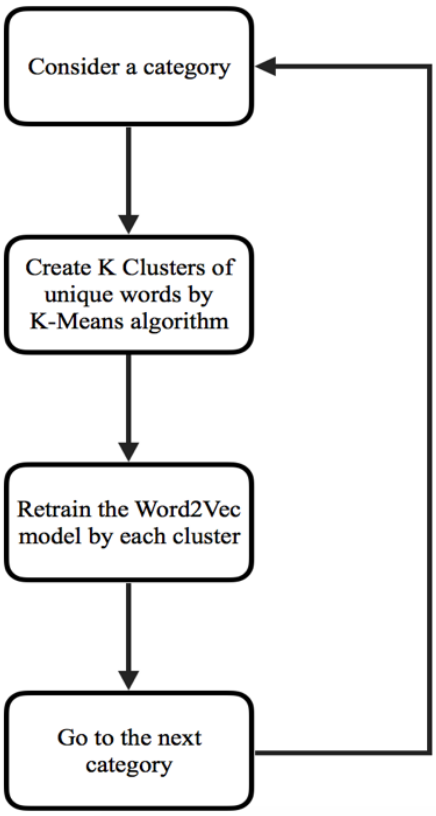

Fig. 1. Workflow of CC-Word2Vec

TABLE II. MLP Model Parameters

Parameter	Value
Activation function	For hidden layers is Relu. For the output layer is Softmax.
Number of hidden layers	4
Dropout	0.3
Loss function	Categorical cross entropy
Optimizer	RMSprop
Number of epochs	500
Number of neurons in each layer (from left to right)	512, 128, 64, 32, 16, 5

2.7. Convolutional Neural Network

In recent years, CNNs have found their way in many fields of study, from image processing to NLP and many other fields. CNNs were introduced for the first time to deal with image data. Afterward, it showed a high performance in other kinds of data like text ones. The architecture of CNNs consists of two main parts: convolution layers and pooling ones. The pooling layer comes after the convolution layer and we can use this sequence unlimitedly. This part of the network is responsible to extract features from the input data and the performance of this job is related to many factors such as the kernel size, the pooling size, padding method, etc.

The architecture of CNN is shown in Figure 2 [21]. In this paper, we will examine the results of a one-dimensional CNN as a classifier. With a small kernel size in early layers, we can extract more ground features. Hence, we select 2 as the kernel size in the first two convolutional layers and 3 for the rest ones. Moreover, max-pooling has a size of 2 between all convolutional layers. To decrease overfitting, we apply the dropout technique after max-pooling layers with a parameter size of 0.3. In the end, we pass the output of CNN to a dense layer with a "softmax" activation method to train the whole network based on the features extracted by the convolution-pooling section. CNN parameters are listed in Table III.

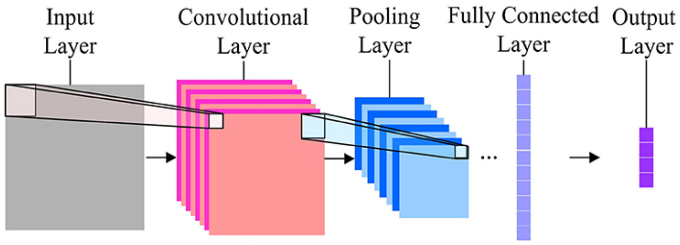

Fig. 2. Architecture of CNNs

TABLE III. CNN Model Parameters

Parameter	Value
Activation function	For convolution layers is Relu. For the output layer is Softmax.
Number of filters in convolution layers (from left to right)	128, 64, 64, 32, 32
Number of neurons in final dense layers (from left to right)	16, 5
Kernel size of convolution layers (from left to right)	2, 2, 3, 3, 3
Dropout	0.3
Loss function	Categorical cross entropy
Optimizer	RMSprop
Number of epochs	500
Pool size	2

TABLE IV. Detext Parameters

Parameter	Value
Feature Extraction Module	Convolution Neural Network (CNN)
Number of Filters	100
Word Embedding Size	300
Training and Test data Batch-size	128
optimizer	Bert-Adam
Number of epochs	500
Learning Rate	0.001

2.8. Latent Dirichlet Allocation

LDA is one of the most popular topic models, described by [22] for the first time in 2003. This method, which is mainly used for the discrete dataset as ours, models each item based on several ground topics. In other words, each document in our dataset is represented by a combination of topic probabilities. This technique can be useful for document classification because each document can be represented by its topics, explicitly. We aim to use the LDA model to compare its results with our method. The number of topics we consider for this paper is 300, the same as the number of features we considered in other methods. It means that each document in our dataset is modeled as a combination of 300 topics. For the LDA model, we removed all stop words as well as words with frequency lower than 5 and trained the model for 500 epochs. This method like a bag of words techniques does not consider the order of words and their contexts. Hence, we expect that its performance would not exceed Word2Vec-base models. The general workflow of LDA is shown in Figure 3 [15]. As this figure illustrates, LDA transfers each document to the distribution of topic probabilities with which the document belongs to that topic.

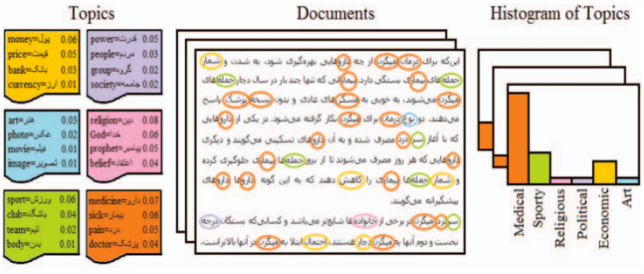

Fig. 3. Workflow of LDA

2.9. Detext Framework

Detext is an open-source deep neural text understanding framework provided by LinkedIn [23]. This framework works based on the End-to-End learning strategy. The input parameters are presented in Table IV. We use this model as comparison material.

For performing classification tasks, the first step is to preprocess the text. At this stage, we remove all numbers, symbols, and extra spaces from the documents and perform lemmatization and normalization using the hazm library [24]. After the text preprocessing, we divided the documents into train and test sets at a ratio of 8 to 2. We extracted features of words by TF-IDF, Word2Vec, and our method, CC-Word2Vec. We eliminated the stop-words for the TF-IDF method; However, we considered them in Word2Vec and CC-Word2Vec models because stop-words can affect the determination of semantic relations between the words which lead to the enhancement of the accuracy of the word embedding model. Ultimately, we passed word vectors as an input to a CNN, MLP, GB classifiers, and detext framework to explore the results of our approach.

3. Experimental Results

In this section, we report the performance of the proposed method in comparison with other approaches.

3.1. Dataset

To investigate our method, we selected 200 documents from 5 frequent categories shown in figure 4 contained in one of the most prestigious Persian text datasets, the Hamshahri news dataset. This dataset and the list of Persian stop words have been provided by the University of Tehran [25].

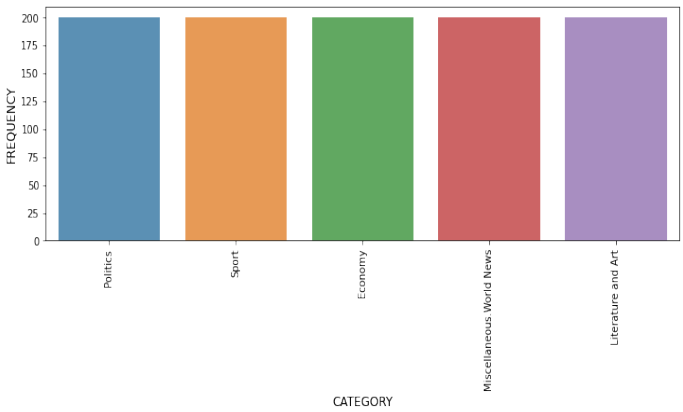

Fig. 4. Categories

3.2. Evaluation

We will use F-measure for the performance evaluation task. F score for each category is calculated as follows.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (6)$$

where TP, True Positive, refers to the number of documents that belonged to a category that are correctly classified. FP, False Positive, refers to the number of documents not belonging to the category but are incorrectly classified to the category. FN, False Negative, refers to the number of documents that are belonged to the category but are incorrectly classified to other ones. Ultimately, Macro F1-score is obtained by taking an average over individual classes' F1- scores.

3.3. K determination for K-Means Algorithm

To find the best *K* for K-means clustering in CC-Word2Vec, we examined different values for it to find the best one. In Figure 5, the F1-score is depicted against the number of clusters *K* for both GB and MLP classifiers. Based on these plots, is selected as the optimum number of clusters since it results in higher F1-score values.

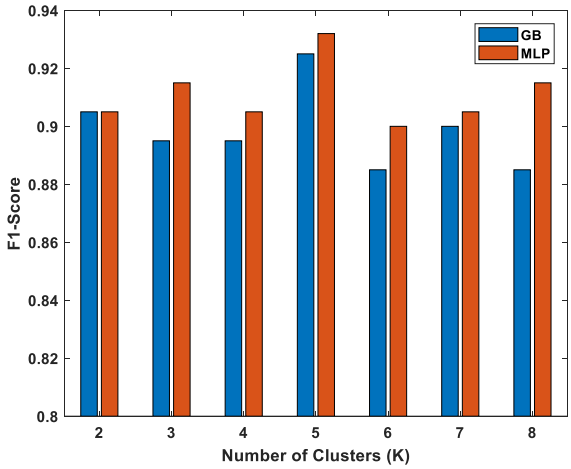

Fig.5. F1-Score against different number of clusters

F1-score values obtained for different classifiers are reported in Table V. We also examine the results of applying K-means clustering to all of the training word vectors instead of different categories and retraining the Word2Vec model for each of the obtained clusters; We call this approach kmeans-Word2Vec. The results of Table V reveal that CC-Word2Vec outperforms LDA, Word2Vec, TF-IDF, and kmeans-Word2Vec methods for three classifiers. Furthermore, our proposed approach exceeds the Detext framework. This performance improvement can be associated with extracting more semantically relevant word vectors using the proposed CC-Word2Vec approach which results in more discriminant features for text classification. We also inspected the Gaussian Mixture Model or GMM as a clustering scheme. GMM is a generalized model of the K-means algorithm in which data points are assumed to belong to a mixture of Gaussian distributions. However, the results for this model were not better than K-Means.

We also combine features extracted by the LDA method with features extracted by TF-IDF, Word2Vec, C-Word2Vec, and CC-Word2Vec methods pairwise to examine the effect of feature combination in document classification. For the GB classifier, this combination does not result in better accuracy, and still features extracted by the C-Word2Vec method lead to the highest score. For the MLP classifier, we could enhance the accuracy of the CC-Word2Vec method slightly with a combination of LDA features. For this classifier, the highest obtained score is 0.935. However, for the last classifier, the CNN model, the CC-Word2Vec method results in an accuracy of 0.87 which is the highest score for this classifier. To sum up, we can report the CC-Word2Vec along with LDA features as the best method in our study.

4. Conclusion

In this paper, we introduced a novel feature extraction approach, CC-Word2Vec which combines categorical clustering with the Word2Vec embedding method to acquire more semantically relevant word representations. We examined its performance through comparison with TF-IDF, Word2Vec, C-Word2Vec, and LDA methods and a deep neural framework, named Detext, for Persian document classification. We used the Hamshahri news dataset for evaluation. For this purpose, we applied the acquired word vectors as the input to a CNN, MLP, and GB classifiers. The results manifested that CC-Word2Vec outperforms all approaches for all of the inspected classifiers.

TABLE V. Macro f1-score values obtained for different approaches

Method	Macro f1-score		
	Gradient Boosting	*Multi-Layer Perceptron*	*Convolutional Neural Network*
TF-IDF	0.895	0.874	0.625
Word2Vec	0.90	0.885	0.845
Kmeans-Word2Vec (*K*=5)	0.901	0.914	0.85
CC-Word2Vec (*K*=5)	**0.925**	0.932	**0.87**
LDA	0.76	0.76	0.305
TF-IDF + LDA	0.90	0.915	0.52
Word2Vec + LDA	0.90	0.905	0.86
C-Word2Vec + LDA (*K*=5)	0.90	0.92	0.845
CC-Word2Vec (*K*=5) + LDA	0.90	**0.935**	0.805
Detext framework	0.905		

References

[1] M. Farhoodi and A. Yari, "Applying machine learning algorithms for automatic Persian text classification," *2010 6th International Conference on Advanced Information Management and Service (IMS)*, Seoul, 2010, pp. 318-323.

[2] S. Zobeidi, M. Naderan, and S. E. Alavi, ”Effective text classification using multi-level fuzzy neural network,” 2017 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Qazvin, 2017, pp. 91-96.

[3] Hu, Xia, and Huan Liu. "Text analytics in social media." In *Mining text data*, pp. 385-414. Springer, Boston, MA, 2012.

[4] Ayoub Bagheri, Hamed Farzanehfar, Mohammad Hossein Saraee, Mohammad Reza Ahmadzadeh, The Farsi text classification using Bayesian Algorithm. Second Iranian Conference on Data Mining of Iran, 2008.

[5] Bina, B., M. H. Ahmadi, M. Rahgozar, “Farsi Text Classification Using N-Grams and Knn Algorithm A Comparative Study.” DMIN (2008).

[6] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. ACM Comput. Surv. 34, 1 (March 2002), 1-47.

[7] S. Z. Mishu and S. M. Rafiuddin, ”Performance analysis of supervised machine learning algorithms for text classification,” 2016 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2016, pp. 409-413.

[8] F. Alzamzami, M. Hoda and A. E. Saddik, ”Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation,” in IEEE Access, vol. 8, pp. 101840-101858, 2020.

[9] Resham N. Waykole, Anuradha D. Thakare. A review of feature extraction methods for text classification. International Journal of Advance Engineering and Research Development Volume 5, Issue 04, April -2018

[10] Jahantigh, Morteza, Negin Daneshpour, Mohammad Erfani, and Nargess Orojlou. "Presenting an improved combination for classification of Persian texts." In *2016 Eighth International Conference on Information and Knowledge Technology (IKT)*, pp. 234-240. IEEE, 2016.

[11] S. Ghasemi and A. H. Jadidinejad, ”Persian text classification via character-level convolutional neural networks,” 2018 8th Conference of AI and Robotics and 10th RoboCup Iran Open International Symposium (IRANOPEN), Qazvin, 2018, pp. 1-6.

[12] N Rezaeian, G Novikova, Persian Text Classification using naive Bayes algorithms and Support Vector Machine algorithm, Indonesian Journal of Electrical Engineering and Informatics (IJEEI), Vol. 8, No. 1, March 2020, pp. 178-188.

[13] S. E. Rad, and A. R. Behjat, ”Document Classification base on Ensemble Classifiers Support Vector Machine, Multi-layer Perceptron and k-Nearest Neighbors.” J. Biochem. Tech., vol. 2, pp. 174-182, Sep. 2019.

[14] Ashkan, Jafari, Ezadi Hamed, Hossennejad Mihan, and Noohi Taher. "Improvement in automatic classification of Persian documents by means of support vector machine and representative vector." In *International Conference on Innovative Computing Technology*, pp. 282-292. Springer, Berlin, Heidelberg, 2011.

[15] P. Ahmadi, M. Tabandeh and I. Gholampour, ”Persian text classification based on topic models,” 2016 24th Iranian Conference on Electrical Engineering (ICEE), Shiraz, 2016, pp. 86-91.

[16] Mikolov, Tomas, Kai Chen, G. S. Corrado and J. Dean. “Efficient Estimation of Word Representations in Vector Space”. ICLR (2013).

[17] Wang, Zhibo, Long Ma, and Yanqing Zhang. "A hybrid document feature extraction method using latent Dirichlet allocation and word2vec." In *2016 IEEE first international conference on data science in cyberspace (DSC)*, pp. 98-103. IEEE, 2016.

[18] Rehurek, Radim, and Petr Sojka, “Software Framework for Topic Modelling with Large Corpora.” In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks. 2010.

[19] Qaiser, Shahzad, and Ramsha Ali. “Text mining: Use of TF-IDF to Examine the Relevance of Words to Documents.” International Journal of Computer Applications 181 (2018): 25-29.

[20] Pedregosa, Fabian, et al. “Scikit-learn: Machine learning in Python.” the Journal of Machine Learning Research 12 (2011): 2825-2830.

[21] Peng, Min, Chongyang Wang, Tong Chen, Guangyuan Liu, and Xiaolan Fu. "Dual temporal scale convolutional neural network for micro-expression recognition." *Frontiers in psychology* 8 (2017): 1745.

[22] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3, null (3/1/2003), 993-1022.

[23] https://engineering.linkedin.com/blog/2020/open-sourcing-detext.

[24] https://www.sobhe.ir/hazm/

[25] AleAhmad, Abolfazl, et al. "Hamshahri: A standard Persian text collection." Knowledge-Based Systems 22.5 (2009): 382-387.

Saeedeh Davoudi received her B.Sc. degree in the field of Computer - Software Engineering from Kharazmi University, Tehran, Iran, in 2018. Currently, she is a master's student in the field of Algorithms and Computation at the department of Engineering Science, University of Tehran, Iran. Her research interests are Natural Language Processing, Deep Learning, and Machine Learning.
Email: saeedeh.davoudi@ut.ac.ir

Sayeh Mirzaei received her B.S. and M.S. degrees in Electrical Engineering from the University of Tehran, Iran. She received her Ph.D. degree in Electrical Engineering, Telecommunications major from KULeuven, Belgium. She is currently the faculty member of the School of Engineering Science, College of Engineering, University of Tehran. Her research interests include the application of machine learning and Bayesian reasoning techniques in analyzing audio, image, text, and biomedical signals.
Email: s.mirzaei@ut.ac.ir

Paper Handling Data:

Submitted: 05-04-2021
Received in revised form: 11-20-2021
Accepted: 11-25-2021
Corresponding author: Dr. Sayeh Mirzaei
Affiliation of the corresponding author: School of ,Engineering Science, College of Engineering University of Tehran, Tehran, Iran