An Experimental Study on Performance of Text Representation Models for Sentiment Analysis

Sajjad Jahanbakhsh Gudakahriz
Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
Sa.jahanbakhsh84@gmail.com
Amir Masoud Eftekhari Moghadam*
Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
eftekhari@qiau.ac.ir
Fariborz Mahmoudi
Advanced Analytics Dept., General Motors Company, Warren, Mi, USA
fzmahmoudi@gmail.com

Received: 05/Feb/2020 Revised: 20/Jun/2020 Accepted: 15/Jul/2020

Abstract
Sentiment analysis in social networks has been an active research field since 2000 and it is highly useful in the decision-making process of various domains and applications. In sentiment analysis, the goal is to analyze the opinion texts posted in social networks and other web-based resources to extract the necessary information from them. The data collected from various social networks and web sites do not possess a structured format, and this unstructured format is the main challenge for facing such data. It is necessary to represent the texts in the form of a text representation model to be able to analyze the content to overcome this challenge. Afterward, the required analysis can be done. The research on text modeling started a few decades ago, and so far, various models have been proposed for performing this modeling process. The main purpose of this paper is to evaluate the efficiency and effectiveness of a number of commons and famous text representation models for sentiment analysis. This evaluation is carried out by using these models for sentiment classification by ensemble methods. An ensemble classifier is used for sentiment classification and after preprocessing, the texts is represented by selected models. The selected models for this study are TF-IDF, LSA, Word2Vec, and Doc2Vec and the used evaluation measures are Accuracy, Precision, Recall, and F-Measure. The results of the study show that in general, the Doc2Vec model provides better performance compared to other models in sentiment analysis and at best, accuracy is 0.72.

Keywords: Text Representation Models; Sentiment Analysis; Sentiment Classification; Ensemble Classifiers**.**

1- Introduction

With the advent of web 2.0 and the growing growth of social networks, the amount of free data produced by the users has reached an unpredictable amount [1]. With the high volumes of data, there is an increasing interest in the scientific community for creating systems that are capable of extracting information from the data. Sentiment analysis which is also known as opinion mining is a solution to the problem. Sentiment analysis refers to a series of methods, techniques, and tools for extracting opinions and emotional information from various types of texts for use in decision support and decision making systems [2-5].

Sentiment analysis employs a variety of techniques like natural language processing, machine learning, text analysis, statistics, and linguistics for analyzing and extracting information from texts [6]. In essence, a good sentiment analysis system might eliminate the need for polling, changes the methodology of traditional commercial research [7] and is very useful in realizing various goals such as identifying public attitude towards political movements, market intelligence, level of customer satisfaction, predicting the movie sales, and a lot more [8]. The wide range of internet-based technologies has increased the number of consumers who use the published opinions of others for making their purchasing decisions [9]. Therefore, various fields include product marketing, commercial intelligence, elections, military campaigns, political campaigns, medicine, software engineering, tourism, and cyberbullying use sentiment analysis [3, 4, 10-13].

Various steps are required for analyzing the input texts since these texts are obtained from different sources with different formats [14]. The process of sentiment analysis usually consists of three predefined steps which include data acquisition, preprocessing the text, and core process [15]. A general schematic of the steps is shown in Fig. 1.

The first step in sentiment analysis is data acquisition which involves obtaining and acquiring the text that can be used for extracting sentiments. The second step in sentiment analysis is the preprocessing of the texts. The messages and posts published on social networks are

* Corresponding Author

usually imperfect, noisy, poorly structured sentences, irregular expressions, typos, and non-dictionary expressions. Therefore, it is necessary to perform the preprocessing operation [16]. The third step in sentiment analysis is the core process, which is the extraction of sentiments about the selected entities from the text. In fact, at this step, the main task of sentiment analysis is carried out which involves various levels and dimensions.

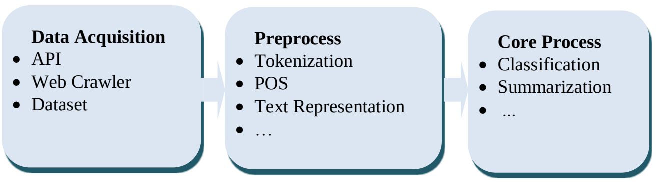

Fig. 1 General Steps of Sentiment Analysis

A fundamental problem of text mining tasks such as sentiment analysis is how to represent the text documents to make them mathematically computable. The raw data in the form of a series of symbols and words aren't proper for the machine learning algorithm; rather, it is required to transform the raw data into the form of a vector or any other format which can easily be processed [17]. Therefore, text representation is one of the important tasks in the preprocessing step of sentiment analysis and the efficiency of the tasks is significantly dependent on the selected model [18]. There are lots of models for text representation, that the most common of these are: TF-IDF, LSA, Word2Vec and Doc2Vec. Each of the models has different application and efficiency. The main purpose of the present study is to compare the efficiencies of these four common and well-known text representation models in sentiment analysis by existing ensemble classifier systems.

To this end, we employed an ensemble classifier system of sentiment classification to evaluate and compare the effectiveness of the four models. In this way, first, the texts are converted to the format of each of the reviewing models. Then it is given to the used ensemble classification system to classify these texts. Finally, the efficiency of the classification is evaluated.

The rest of this paper is organized as follows. Section 2 reviews in details the compared natural language representation models in this study. Section 3 presents the used ensemble classifier system in this study. The details of the experiments used for comparing the models and the results of these experiments are presented in section 4. Finally, Section 5 provides a summary and conclusions of the study.

2- Overview of Text Representation Models

In text mining and sentiment analysis, it is necessary to transform the selected text into a representation that is ready to be understood and analyzed by computer algorithms. The mapping from textual data to real-valued vectors is called feature extraction. The main challenge is finding the most valuable features by the models. The important matter is that before modeling the text, the main step is to break down the input string into words and then each word must be expanded and explained. The result will be a set of expressions corresponding to the input string. There are various models for doing this, in this section; we will discuss four models including TF-IDF[1], LSA[2], Word2Vec, and Doc2Vec. These models are commonly recognized models which are highly popular.

2-1- The TF-IDF Model

The TF-IDF model is one of the most common models of weighting words. In this model, each word is given a weight based on its frequency and the frequency of the word in other corpus texts [19]. In fact, the main goal of the weighting system is to show the importance of the word in the text. The weight of a word increases with its frequency in the text; however, it is controlled by the number of texts where the word appears. In other words, if a word appears in the majority of texts, it is most likely a common word, so it doesn't have a high value for the text evaluation. This method can be a good measure for determining the weight and importance of a word in a text. In fact, it shows how much a word can be unique and significant.

The value of TF-IDF is obtained by multiplying the term frequency in the document, $TF(t,d)$, by the inverse document frequency, $IDF(t,D)$, using Formula 1.

$$TFIDF(t,d,D) = TF(t,d).IDF(t,D) \qquad (1)$$

The value of $TF(t,d)$ is the number of times that the term t frequencies in document d and is normalized using formula 2.

$$TF(t,d) = \frac{F(t,d)}{Max\ F(t_i,d)} \qquad (2)$$

The $IDF(t,D)$ value is calculated using formula 3. In this formula, D indicates the set of documents, N is the size of D, and $|d \in D: t \in d|$ is the number of documents in the D that the term t appears.

$$IDF(t,D) = log\frac{N}{|d \in D: t \in d|} \qquad (3)$$

Input Layer
1×V

[1] Term Frequency-Inverse Document Frequency
[2] Latent Semantic Analysis

2-2- The LSA Model

One of the methods for improving the text representation model is to use semantic information [18]. This idea is based on the assumption that words with lexical differences which are frequently used in similar documents have similar meanings. Semantic information can be computed using the LSA model. This model is a technique in text processing that can be used for modeling the conceptual relationship among a number of documents based on their set of words [20].

To represent a series of documents using the LSA model, a matrix (A) with the dimensions of $m \times n$ is created where n is the number of documents and m is the number of terms in all documents. Each column in A indicates a document (d) and each row indicates a term (t) (Fig. 2). There are various techniques for computing the values of the cells of this matrix, which indicate the frequency of the terms in the document. These methods include the TF-IDF method and the entropy method.

	Document 1	Document 2	…	Document n
Term 1				
Term 2				
…				
Term n				

Fig. 2 The Used Matrix in LSA Model

2-3- The Word2Vec Model

The Word2Vec is one of the models related to text processing where each term in the text is transformed into a vector [21]. This model uses a two-layer neural network and it's the most important idea in converting the words into vectors and transferring those to the vector space where the processing of words and documents can be easily done through machine learning tools. The output of the Word2Vec model is a vocabulary of words in the initial document along with their vector representation in an n-dimensional vector space. Related words or word groups are placed close to each other in this space, that this causes the semantic relationship between words to be maintained. The architecture of the Word2Vec model is shown in Fig. 3.

Hidden Layer
1×N

Output Layer
1×V

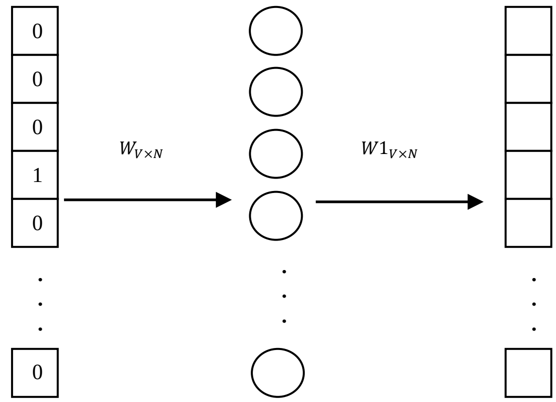

Fig. 3 The Architecture of Word2Vec Models [21]

2-4- The Doc2Vec Model

The Doc2Vec model is a developed model of the Word2Vec that applies to a sentence, paragraph, or text and creates a vector for it. The Word2Vec model produces high-quality word vectors; however, it doesn't have a specific method for combining these vectors to represent a text in the form of vectors. The Doc2Vec model was proposed for vector representation of a document. This model converts all the words in a sentence into a vector and puts the sentence label as a word inside this vector [22]. The vectors produced by the Doc2Vec model can be used to find semantic similarities between texts. The architecture of the Doc2Vec model is shown in Fig. 4.

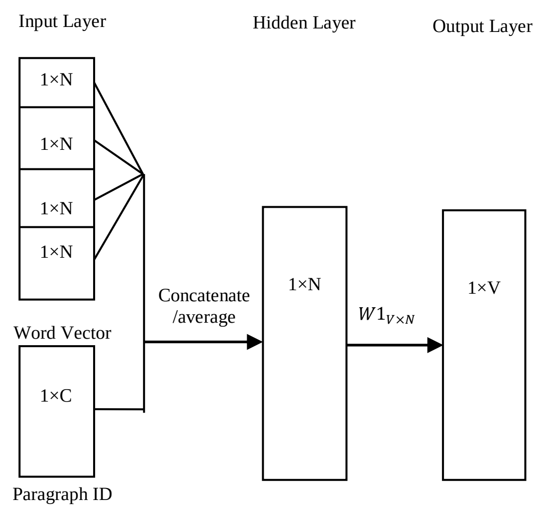

Fig. 4 The Architecture of Doc2Vec Model [22]

3- Ensemble Sentiment Classifier for Evaluation the Reviewed Models

Sentiment classification is the determining of the polarity of opinion texts in two or more classes. This can be done using machine learning methods, knowledge-based methods, or a combination of them. Machine learning methods provide a high level of accuracy, while knowledge-based methods provide a better generalization capability [15]. In machine learning methods, classification algorithms are used for separating sentiments and opinions into a number of separate classes.
In recent years, researchers suggested that ensemble classifiers provide a good capability for classification [23, 24, and 25]. Studies on using a combination of a number of basic classifiers show that using ensemble classifiers can improve the accuracy of sentiment classification [26].
In general, there are three common methods for combining classifiers including bagging, boosting, and random subspace [23]. In the bagging method, each one of the learners is created using independent random copies of the training dataset and the final results are calculated using simple majority polling. In the boosting method, basic learners are created from the weighted versions of the training dataset, which depend on the results of previous basic learners, and the final result is calculated using simple polling or weighted majority polling. In the random subspace method, basic learners are created based on the training dataset divided into a number of segments based on specific characteristics.
In this paper, we used the existing bagging method for sentiment classification. The used ensemble classifier system for evaluating text representation models is shown in Fig. 5. As can be seen from this figure, first the selected data for analysis is acquired from the dataset. Then, the necessary preprocessing tasks are performed on the dataset, and at the end of this step, the data is converted into a text representation model. Next, the prepared data is given to the classifiers, and the results obtained from the three classifiers are combined using majority polling. Finally, the result of the vote is evaluated.
Various classifiers were used in the experiments and finally based on the performance of these classifiers, Logistic Regression, K-NN and SVM classifiers were selected to be used in the ensemble system. Moreover, the preprocessing operations include tokenization, lemmatization, removing punctuation, ignoring usernames and hyperlinks, and replacing letters with a sequential repetition of more than twice in one word with two letters.
As shown in Fig. 5 and mentioned above, before sentiment classification, first the texts are converted to the format of the reviewed models, then the classification is done on the texts. Given that, the purpose of this paper is to evaluate the efficiency of four models, a fixed ensemble classifier is used, but the text display models are changed. In this way, the sentiment classification is done once with the TF-IDF model, once with the LSA model, once with the Word2Vec model and once with the Doc2Vec model. As a result, classification efficiency reflects the impact of each model on sentiment classification. In this way, the efficiency of each model in sentiment analysis is evaluated. In the next section, the experiments and the results are being explained.

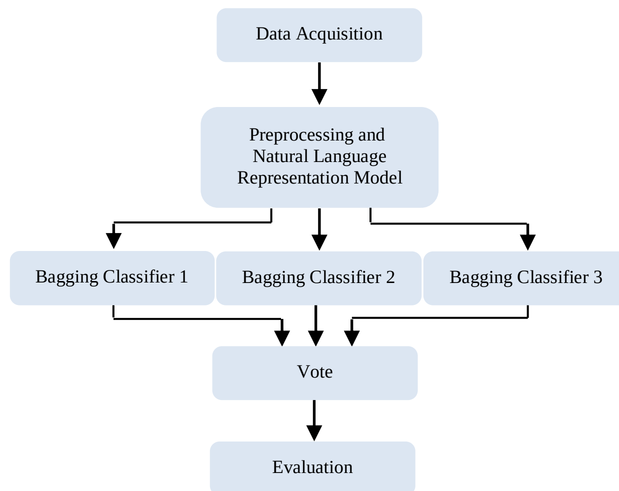

Fig. 5 The Used Ensemble Classifier

4- Experiments and Evaluation

In this section, the performed simulations for evaluating the selected models are presented. The evaluation includes measuring, comparing and analyzing the obtained results. To realize this goal, the ensemble classifier system explained in section 3 was applied on a number of datasets and the effects of various models on sentiment classification were investigated by applying these models in the preprocessing step. The simulations were done in Python 3 and the used datasets, metrics for measuring the efficiency, and the results of the simulations are presented below.

4-1- Datasets

To evaluate the performance of various models of text representation and comparing them to each other, it is required that the conditions of the experiments to be identical and the used data to be standard. Therefore, the experiments were performed on the existing four datasets known as HCR [27], Sanders [28], SemEval [29], and STS [30]. These datasets are common standard datasets for use in sentiment analysis and are used in many papers. They are collected by API from tweets on Twitter. In order to improve confidence, these tweets were manually labeled based on the sentiments in each one. These datasets are related to different issues and they have two classes include positive and negative classes. From each of these datasets, 1000 tweets with positive and negative sentiments were selected, where 700 tweets were used for training and 300 tweets were used for testing. Among the selected tweets, half of them contained positive sentiments and the other half contained negative sentiments.

4-2- Performance Evaluation Measures

As mentioned above, the goal of the current study is to evaluate the efficiency of the selected text representation models by using them in sentiment classification. In fact, the model with the higher efficiency will increase the efficiency of the classifier system. Therefore, various measures must be selected for evaluating the used classifier system. Evaluating the efficiency of the sentiment classification system involves estimating how well a classifier system can predict the classes of texts with positive or negative sentiments. In the experiments, important measures were used for evaluating the efficiency of the sentiment classification system, and these include:

- Accuracy: This measure indicates what percentage of predictions is performed correctly and it is calculated using Formula 3.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

- Precision: This measure indicates what percentage of positive results is actually positive, and it is calculated using Formula 4.

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

- Recall: This measure indicates what percentage of the positives is predicted correctly, and it is calculated using Formula 5.

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

- F-Measure: This measure is the harmonic average of accuracy and recall, and it is calculated using Formula 6.

$$F-Measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (6)$$

As can be seen from Formula 3-6, to calculate the values of the evaluation measures, we have to extract the values for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These values are calculated as follows:

- TP: The number of positive tweets correctly predicted as positive.
- TN: The number of negative tweets correctly predicted as negative.
- FP: The number of negative tweets incorrectly predicted as positive.
- FN: The number of positive tweets incorrectly predicted as negative.

4-3- Results of the Experiments

In this section, the results related to each one of the selected models are shown based on the measured metrics on the used datasets. Fig. 6 shows the results related to the accuracy of various datasets and models. As can be seen, the best performance among all the models and on all the datasets is achieved by the Doc2Vec model. On average, the accuracy value obtained for this model on all the datasets is 0.67. Moreover, the best value of accuracy obtained for different datasets is achieved by this model, and it is 0.72. On various datasets, other models have benefits over each other, and considering the average values for each model on all the datasets, after the Doc2Vec model, the LSA model with an average accuracy of 0.59 is achieved the best performance. Then, the Word2Doc model provided an average accuracy of 0.58. Finally, the weakest performance is achieved by the TF-IDF model, with an average accuracy of 0.57. Considering the average accuracy obtained from all the models on each dataset, on the HCR dataset, the average accuracy is 0.59; on the Sanders dataset, the average accuracy is 0.62; on the SemEval dataset, the average accuracy is 0.63; and on the STS dataset, the average accuracy is 0.56.

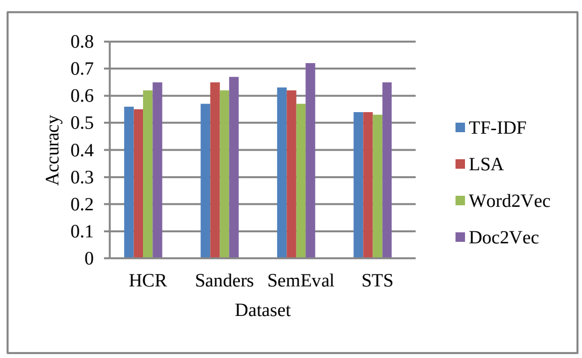

Fig. 6 Results for the Accuracy Measure

Fig. 7 shows the results for the precision measure on different datasets and models. The highest precision is achieved 0.72 by the LSA model on the Sanders dataset. On the other datasets, the Doc2Vec model has better performance, and the average precision for the Doc2Vec model on all the datasets is achieved 0.66, which is higher than the average values for all the other models. With regards to the average values of precision for each model on all the datasets, the Doc2Vec model is followed by the LSA model with an average precision of 0.60. Then, the TF-IDF model, with an average precision of 0.58 has the best performance, followed by the weakest performance by the Word2Vec model with an average precision value of 0.57. With regards to the average precision value for all the models on each dataset, on the HCR dataset, the average precision is achieved 0.58; on the Sanders dataset, the average precision is achieved 0.65; on the SemEval dataset, average precision is achieved 0.63; and on the STS dataset, average precision is achieved 0.56.

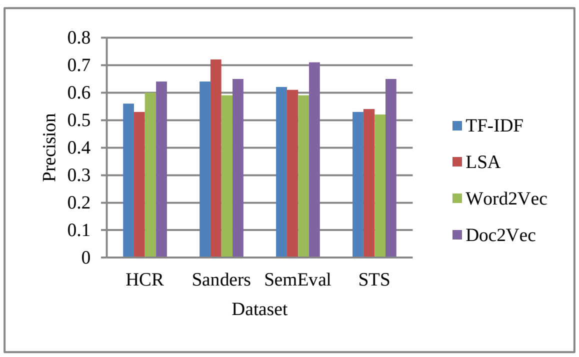

Fig. 7 Results for the Precision Measure

Fig. 8 shows the results for the F-Measure metric on different datasets and models. As can be seen, the Doc2Vec model achieves better performance compared to other models, and it has the best F-Measure value for various datasets equal to 0.72. On average, the F-Measure obtained for this model on all the datasets is 0.67. With regards to the average value for each model on different datasets, the Doc2Vec model is followed by the LSA model with an average F-Measure value of 0.58, followed by the Word2Vec model with an average F-Measure value of 0.57, followed by the TF-IDF model with the weakest performance and an average F-Measure value of 0.54. With regards to the average F-Measure value obtained for all the models on each dataset, on the HCR dataset the average F-Measure value is achieved 0.58; on the Sanders dataset the average F-Measure value is achieved 0.57; on the SemEval dataset, the average F-Measure value is achieved 0.64; and on the STS dataset, the average F-Measure value is achieved 0.56.

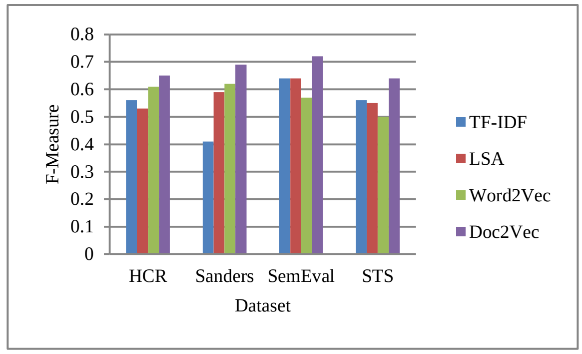

Fig.8 Results for the F-Measure Metric

Fig. 9 shows the results of the recall measure for various datasets and models. As can be seen, the best recall value for different datasets is achieved 0.73 for the Doc2Vec model on the Sanders dataset. The Doc2Vec model obtains the best performance among all the models and datasets, and on average, the recall value obtained for this model on all the datasets is achieved 0.68. With regards to the average value of recall for each model on various datasets, after the Doc2Vec model, the Word2Vec model with an average recall of 0.57 has the best performance. This model is followed by the LSA model with an average recall value of 0.56, and the TF-IDF model provides the weakest performance with an average recall value of 0.54. With regards to the average recall value obtained for all the models on each dataset, on the HCR dataset, the average recall is achieved 0.59; on the Sanders dataset, the average recall is achieved 0.54; on the SemEval dataset, the average recall is achieved 0.65; and on the STS dataset, the average recall is 0.56.

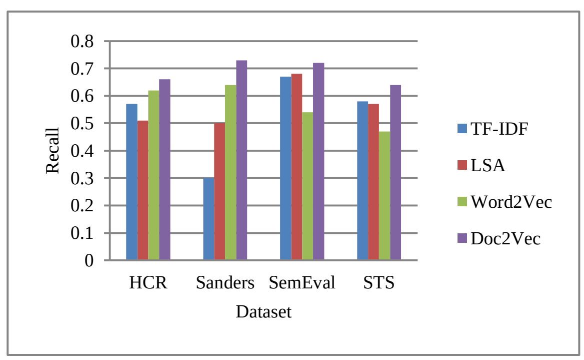

Fig. 9 Results for the Recall Measure

Based on Figs 6-9 and the results explained above, it can be concluded that the Doc2Vec model provides the best performance in sentiment analysis among the four tested models. Based on the calculated averages as well as the four measured metrics and considering the high significance of accuracy and precision measures, it can be said that the LSA model has second performance in sentiment analysis. After this model, the Word2Vec model comes next in terms of performance and the weakest performance is achieved by the TF-IDF model. Moreover, based on each dataset and the overall average values obtained in the experiments, it can be said that the performance of all the models is better on the SemEval dataset compared to all the other datasets.

5- Conclusions

Text representation models play an essential role in sentiment analysis. Therefore, various models have so far been proposed for performing this task. These models have different efficiency in different applications, as well as many of these models are used in sentiment analysis researches. Therefor the main purpose of the current study was to compare the efficiency of common and famous text representation models in this field in sentiment analysis by experiments. For this goal, an ensemble classifier system was used for sentiment classification and at the preprocessing step of this system, each of the models including TF-IDF, LSA, Word2Vec, and Doc2Vec were used. Four different standard datasets of tweets were selected, and the efficiency of these models was evaluated on these datasets. The simulation results show that in general, the Doc2Vec model provides better performance compared to other models based on basic evaluation measures for sentiment classification. It seems that using this model in other applications of sentiment analysis can lead to better results since the classification of opinions is a primary task in sentiment analysis and the efficiency of classification has a significant direct impact on other possible operations as well.

References

[1] A. Montoyo, P. Martínez-Barco, and A. Balahur, “Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments”, Decision Support Systems, Vol. 53, 2012, pp. 675–679.

[2] J. A. Balazs, and J. D. Velasquez, “Opinion Mining and Information Fusion: A survey”, Information Fusion, Vol. 27, 2016, pp. 95–110.

[3] F. A. Pozzi, E. Fersini, E. Messina, and B. Liu, Sentiment Analysis in Social Networks (Chapter 1), Science Direct, 2017.

[4] V. M. Mika, G. Daniel, and K. Miikka, “The evolution of sentiment analysis—A review of research topics, venues, and top cited papers”, Computer Science Review, Vol. 27, 2018, pp. 16-32.

[5] E. Cambria, “Affective computing and sentiment analysis”, IEEE Intelligent Systems,Vol. 31, No. 2, 2016, pp. 102-107.

[6] P. Gupta, R. Tiwari, and N. Robert, “Sentiment Analysis and Text Summarization of Online Reviews: A Survey”, in International Conference on Communication and Signal Processing, 2016, pp. 241-245.

[7] J. A. Balazs, and J. D. Velasquez, “Opinion Mining and Information Fusion: A survey”, Information Fusion, Vol. 27, pp. 95–110, 2016.

[8] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydar, “Detection of review spam: A survey”, Expert Systems with Applications, Vol. 42, 2015, pp. 3634–3642.

[9] A. Qazi, R. Raj, G. Hardaker, and C. Standing, “A systematic literature review on opinion types and sentiment analysis techniques: tasks and challenges”, Internet Research, Vol. 27, 2017.

[10] F. Bisio, L. Oneto, and E. Cambria, Sentiment Analysis in Social Networks (Chapter 5), Science Direct, 2017.

[11] S. Sun, C. Luo, and J. Chen, “A Review of Natural Language Processing Techniques for Opinion Mining Systems”, Information Fusion, 2016.

[12] P. Sobkowicz, M. Kaschesky, and G. Bouchard, “Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web”, Government Information Quarterly, Vol. 31, 2012, pp. 470–479.

[13] M. Thelwall, “Sentiment Analysis for Tourism”, Big Data and Innovation in Tourism, Travel, and Hospitality, 2019, pp. 87-104.

[14] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, “A survey of sentiment analysis in social media”, Knowledge and Information Systems, 2018.

[15] K. Ravi, and V. Ravi, “A survey on opinion mining and sentiment analysis: Tasks, approaches and applications”, Knowledge-Based Systems, Vol. 89, 2015, pp. 14–46.

[16] Z. Jianqiang, and G. Xiaolin, “Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis”, IEEE Access, Vol. 5, 2017, pp. 2870 – 2879.

[17] L. W. Manod, P. J. Wiliam, and T. R. Michael, “Variable segmentation and ensemble classifiers for predicting dairy cow behavior”, Biosystems engineering, Vol. 17, 2019, pp. 156-167.

[18] T. Brychcin, and M. Konopik, “Semantic spaces for improving language modeling”, Computer Speech and Language, Vol. 28, 2014, pp. 192–209.

[19] A. Aizawa, “An information-theoretic perspective of tf–idf measures”, Information Processing and Management, Vol. 39, 2003, pp. 45-65.

[20] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis”, Journal of the American Society for Information Science, Vol. 41, 1990, pp. 391–407.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space”, Proc. Workshop at ICLR, 2013.

[22] Q. Le, and T. Mikolov, “Distributed representations of sentences and documents”, in Proceedings of the 31st International Conference on Machine Learning (ICML 2014), 2014, pp. 1188–1196.

[23] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, “Sentiment classification: The contribution of ensemble learning”, Decision Support Systems, 2013.

[24] I. Furxhi, F. Murphy, M. Mullins, and C. A. Poland, “Machine Learning Prediction of Nanoparticle In Vitro Toxicity: A Comparative Study of Classifiers and Ensemble-

Classifiers using the Copeland Index", Toxicology Letters,Vol. 312, 2019, pp. 157-166.
[25] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, and E. Agirre, "Word n-gram attention models for sentence similarity and inference", Expert Systems With Applications, Vol. 132, 2019, pp. 1-11.
[26] N. F. F. Da Silva, E. R. Hruschka, and J. E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles", Decision Support Systems, 2014.
[27] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph", in Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP, 2011.
[28] W. Deitrick, and W. Hu, "Mutually enhancing community detection and sentiment analysis on twitter networks", Journal of Data Analysis and Information Processing, Vol. 1, 2013, pp. 19-29.
[29] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter", in Proceedings of the 7th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2013.
[30] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision", CS224N Project Report, Stanford, 2009.

Sajjad Jahanbakhsh Gudakahriz received the B.S. degree in Computer Engineering from Azad University, Ardabil Branch, Iran in 2008, and M.S. degree in Computer Engineering from Azad University, Qazvin Branch, Iran, in 2011. Currently he is Ph.D. Candidate in Azad University, Qazvin Branch, Iran. His research interests include Sentiment Analysis, Sentiment Summarization, Data mining and Computer Networks.

Amir Masoud Eftekhari Moghadam was born in Tehran, Iran in 1964. He received B.E. and M.Sc. degree in computer engineering from Iran University of Science and Technology in 1992 and 1995, respectively and Ph.D. degree in computer engineering in 2002 from Science and Research branch, Islamic Azad University, Tehran, Iran. He is associate professor at the Islamic Azad University, Qazvin, Iran. His research interests include image processing, image/video coding, machine learning, data mining, Big data analysis and semantic video analysis.

Fariborz Mahmoudi received BS, MS, and Ph.D. degree in computer engineering. From 2002 to 2013, he had been an assistant professor in Electrical, Computer, and IT engineering faculty at the Qazvin Azad University, Iran. From 2013 until 2016, he had continued his researches in the image analysis lab of Henry Ford Health System (Detroit, MI, USA). Since Jan. 2017, he is a data scientist in General Motors company (Warren, MI, USA). His research interests include machine learning, machine vision, and information (image and text) retrieval. Dr. Mahmoudi has published more than 100 papers in the international scientific journals and conferences proceedings.