International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

NOVEL OPINION MINING SYSTEM FOR MOVIE RVIEWS IN TURKISH

Abdul Hafiz ABDULHAFIZ

Submitted: 10/02/2020 Accepted: 11/05/2020

Abstract: Opinion Mining (OM) works on transferring the online available opinions into useful knowledge. In this paper, a novel opinion mining system of reviews in Turkish has been presented. The proposed system utilizes Word2Vec, which is one of the states of the art text feature extraction method, along with an ensemble learning algorithm for classification. The challenging and benchmark "IMDB Movies Reviews" dataset has been used for conducting the experimental comparison and verification. In addition, the performance of the proposed method is compared to some of the well-known machine learning algorithms like Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Naive Bayes (NB). The tested ensemble methods are the Random Forest (RF), AdaBoost Classifier, and Gradient-Boosting Classifier (GBC). The results of the conducted experiments using the dataset have shown that the performance of SVM, KNN, and NB are comparable. However, the performance, robustness, and stability of the system have been significantly improved by adapting the RF ensemble learning, along with the Word2Vec feature vector, and suitable pre-processing operations on the data. In addition, the proposed method is compared to one of the states of art ensemble methods and have shown superior performance with respect to it.

Keywords: *Ensemble Learning, Opinion Mining, Sentiment Analysis, Text Classification*.

1. Introduction

Companies have rapidly increased the usage of social media in marketing their products and services in the last few years [1]. It is motivated by the available reviews given by customers. This is besides the shared thoughts and opinions about the products and services. In other words, companies are able to improve their products and services based on users' opinions.

Opinion mining, also called Sentiment analysis (SA), is a collection of methods, techniques, and tools that mainly focus on opinions that convey or indicate a certain sentiment that can either negatively or positively classified [2]. Machine learning plays a main role in the approaches to the OM. Lexicon-based methods and hybrid approaches are also presented in the literature [3, 4]. The lexicon-based approach uses a set of predefined words or phrases known as seed words to define whether the text is positive or negative.

The machine learning approach is either supervised or unsupervised learning. The supervised learning uses a labeled (structured) dataset to train classifiers to determine whether the tested text is positive or negative. In contrast, unsupervised machine learning methods use an unstructured dataset. In addition, the hybrid approach combines both lexicon-based and machine learning approaches [5]. The available large size datasets and the improvements in machine learning techniques in recent years attract the attention of many researchers to work on improving the OM systems.

This paper presents a novel opinion mining algorithm for Turkish movie reviews. The proposed algorithm utilizes a random forest-based classifier along with a Word2Vec feature extraction stage to classify the movie reviews into positive or negative classes. In addition, this work is the first to use a data set of size 53,400, while previous works reported their experiments using a dataset with a size close to 5000 that is ten times smaller than the one used here.

The remaining of this paper is organized as follows. The next section presents the most related work. It concludes with a presentation of the novelty of our work and how it is different from the reviewed previous works. Section 3 builds the background required for the machine learning algorithms presented in this paper. Section 4 presents the main algorithm and explain its structure. The experimental works are evaluated in Section 5, while our concluding remarks are presented in Section 6.

2. Related work

In this section, we review several opinion mining works built on machine learning algorithms and compare them to our proposed algorithm. They utilize algorithms like the k-means, KNN, SVM, NB algorithms, and a variety of ensemble learning methods including Bagging, Boosting, and random forests RF. In addition, we summarize in the following some of the recent studies and achievements in the field of Turkish language sentiment analysis.

The work presented in [6] is one of the recent sentiment analysis works. It is built based on the k-means clustering algorithm. It classifies the customer review at the phrase level. The terms with high-frequency are extracted using a keyword extraction technique and used for extracting keywords from each document, while the intensity of sentiment polarity is calculated by measuring its strength. However, the experimental work of [6] has revealed that the proposed approach has categorized the majority of reviews as neutral, i.e., it fails in classifying a large number of the positive and negative documents correctly.

The sentiment analysis for a collection of election tweets has been

[1]*Department of Computer Engineering, Hasan Kalyoncu University, Gaziantep, Turkey.*
ORCID ID: 0000-0002-1908-5521
** Corresponding Author Email: abdul.hafez@hku.edu.tr*

studied [7]. To do so, an approach that is based on building a dynamic dictionary of sentiments or words by polarity using a small set of positive and negative hashtags related to a predefined topic, where the case study of this paper was the 2016 US presidential election. Overall, results showed that using a dynamic dictionary leads to outperform other SA tools, i.e., IBM Watson text analytics, Rapidminer, Meaning cloud, and StreamCrab [7]. SentiWordNet's approach that uses a machine-learning approach for sentiment classification was presented in [8]. The SentiWordNet approach is a lexical resource for sentiment analysis that firstly introduced in [9] and [10], and now publicly available as a lexical resource for opinion mining tasks. The results of this study, which is done using a dataset of hotel reviews, showed that the supervised ML algorithm can outperform the SentiWordNet based approaches.

The method presented in [11] combines both NB and SVM classifiers were proposed. Their experimental results have shown that the proposed method outperformed the NB classifier, but it has achieved close to one of SVM classifiers alone. Turkish language sentiment analysis research has attracted interest in recent years. For example, the work presented in [12] can be considered as one of the first studies in Turkish sentiment analysis, the SVM classifier and n-grams were used to classify some Turkish reviews. The effect of part-of-speech tagging, spell checking, and stemming is also studied in this work. The system has achieved around 85% accuracy on the binary sentiment classification. In [13], a sentiment analysis system for Turkish using different sentiment levels such as aspect, sentence, and document was introduced. The obtained accuracy was between 60% to 79% for both ternary and binary classification tasks.

A lexicon-based sentiment analysis system has been proposed in [14]. It uses the SentiStrength library [15]. This Turkish sentiment analysis framework has been tested on the same dataset of [12], and report an accuracy of 76% for positive/negative classification. In [16], a comparison of machine learning and lexicon-based sentiment analysis methods on Turkish social media was performed. The lexicon is formed by translating an English opinion lexicon into Turkish. Machine learning-based sentiment analysis was able to outperform the lexicon-based one.

Turkish sentiment classification system that uses ensemble learning was proposed in [17]. It integrates Naive Bayes, Support Vector Machine (SVM), and Bagging classifiers. In this work, the SVM is used as the base classifier of the Bagging, while the parameters of the Naive Bayes and the second SVM are tuned using the "CVParameterSelection" parameter optimization algorithm. The N-grams model is used to generate the text's feature vector and the majority-voting rule was used for finding the class of each sample. The results of this study showed that the proposed system outperformed the used individual classifiers. In conclusion, this system was able to achieve quite good results; however, it is a complicated system, as the processes of Bagging and SVM parameter optimization are time-consuming. An approach that hierarchically combines RF and SVM algorithms was introduced in [18]. In this, the text's features are first classified by the RF classifier into two classes, i.e., positive and negative. Then the features of all texts that have been classified as negative are once again classified by the SVM classifier. This process was built on the assumption that SVM can beat the performance of RF when both are required to classify negative texts. The results of this study showed that the proposed hierarchically system outperforms the individual performance of the used classifiers.

Two main Turkish SA resources are nowadays existed, which may help in advancing all the Turkish SA research. The first resource is SentiTurkNet [19], which is a Turkish lexicon that includes 15,000 synsets with their Part-of-Speech Tagging. It also includes three associated polarity values, i.e., positive, negative, and neutral/objective. In other words, the polarity scores, in this approach, refers to the measurement of negativity, objectivity, and positivity, and sum up to one. The second resource is Turkish datasets that include movie reviews along with their linked binary sentiment polarity labels, i.e. either Positive (P) or Negative (N). These datasets are presented in [20] and [21], and considered as benchmarks for sentiment classification. The dataset of [20] contains 5331 positive and 5330 negative movie reviews. The one presented in [21] contains 53,400 movie reviews divided equally into the two classes, i.e., negative and positive. To form this dataset, rated Turkish movie reviews on a scale of 0 to 5 were collected from the Turkish movie website "beyazperde.com". The reviews rated by 1 or 2 stars were classified as negative, the reviews rated by 4 or 5 stars classified as positive, and the reviews rated by 3 were ignored.

The main contribution of the work presented in this paper can be summarized as follows

1) Developing an approach that takes the advantages of both Word2Vec and RF to produce an efficient system that has more robustness and scalability as compared with approaches that use a single classifier such as approaches of [11-13].
2) Using RF and word2vec for Turkish movie reviews has resulted in a simpler system with better performance compared to other systems such as [17] and [18]. Furthermore, a second classifier has used in [18] to enable the system to classify negative statements; our system is able to efficiently classify them using simple RF classifiers.
3) Up to our knowledge, this paper is the first Turkish research that uses the complete dataset of reviews available [21], while other studies use a small fraction of movie reviews that have been humanly preprocessed. By testing the system using this number of files and the 10-fold cross-validation, we ensure the accuracy of the results and avoid overfitting.

It can be concluded from the experimental works below that the proposed system has significantly outperformed other commonly used classifiers and other well-known ensemble machine-learning algorithms for movie reviews analysis and opinion prediction in particular.

3. Background on Machine Learning Approaches

The proposed method employs the RF learning approach for classification, while the performance is compared to other ensemble methods like AdaBoost and GBC classifiers, also compared to basic methods like KNN, NB, SVM. In this section, we briefly introduce the basic concept of each of them. For more details, readers are referred to the text cited in the corresponding section.

3.1. Basic methods

K-Nearest Neighbor (KNN): K-nearest neighbors is a simple algorithm that stores all existing training samples [22]. It classifies the new data (test sample) according to a selected similarity criterion. The similarity is measured as the inverse of a distance function. The most commonly used distance functions are Manhattan, Euclidean, and L_{inf} *norm* distances [23]. KNN is a powerful non-parametric technique for statistical estimation and pattern recognition since the early 1970s. In our work, the tweet, i.e. the opinion text, is classified by assigning to it the label of the majority of its nearest K neighbors.

Naïve Bayes (NB): Naive Bayes classifier is a probabilistic classifier derived from Bayes theorem with naive independence assumption. NB classifier is easy to implement with simple parameter estimation like the maximum likelihood [24]. This in fact makes it useful particularly for very large datasets. Naive Bayes operates by assuming independence, i.e., the presence of some feature will not affect the other features [25].

Support Vector Machine (SVM): Support Vector Machine is a common supervised machine learning algorithms [26]. It searches for an optimal boundary (called hyperplane) that maximizes the margin (distance) between the classes. It works by representing each training data sample as a point in an n-dimensional space, where n refers to the number of features used to represent the sample. Then, the classification process is performed by detecting the hyper-plane that most discriminates the classes while maximizing the margin between training data points in deferent classes. Maximizing the margin provides some reinforcement so that future data points can be classified with more confidence.

3.2. Ensemble methods

In addition to the above algorithms, some of the state of the art ensemble-based algorithms have been investigated, hence they are briefly summarized below.

Random Forest (RF): It is an ensemble classifier that is designed based on integrating multiple decision tree models [27]. In other words, it consists of a set of individual decision trees working as an ensemble. Each tree in the random forest predicts a class for a certain test sample, then the class is selected based on the majority voting. However, being the processing time is much larger for larger number of samples, is one of its drawbacks. Random forest applies and uses the bagging technique (bootstrap aggregating) [28], it works well on relatively large datasets, but on the account of increasing the execution time when processing a large number of samples.

AdaBoost Classifier (AdaBoost): The AdaBoost or Adaptive Boost classifier is an iterative ensemble method that works on boosting the performance of weak classifiers. In more detail, the classifiers are added to the ensemble one at a time, and the newly added classifier is trained on data that the previous member(s) was not able and has difficulties to classify it correctly. Hence, it trains the model by selecting the training set based on the estimation of the last training [29].

GradientBoosting Classifier (GBC): Gradient boosting is an ensemble learning technique that is very frequently used for regression and classification problems. It produces a prediction model by sequentially fitting the base learner to current "pseudo"-residuals, which are the gradient of the loss functional being minimized by least-squares at each iteration [30].

It is worth nothing mentioning that the major difference between the AdaBoost and the Gradient Boost algorithm is the method used for determining the weaknesses of weak learnings, where gradient boosting identifies weaknesses using gradients in the loss function, while the AdaBoost model carries this by using high-weight data points [30].

4. The Proposed System

The main aim of the proposed system is to classify the newly available reviews about a movie given movie reviews along with their linked binary sentiment polarity labels. The IMDB Movies Reviews Dataset is used for training the proposed RF classifier. The data is represented to the classifier input in the form of Word2Vec features, which are produced by a feature extraction stage. The feature extraction is preceded by a suitable pre-processing stage to ensure the suitability of the data representation and improve its quality.

The main components of the proposed OM system are: Data pre-processing, feature extraction, and classification. These components and data flow through the system are depicted in Figure 1. They are also detailed in the following subsections.

4.1. Data pre-processing:

The aim of the pre-processing stage is to improve the quality of the used data and eliminating useless information. The pre-processing stage in the proposed system includes tokenization, cleaning, detecting and correcting misspelling words, stemming, and normalization. In this paper, the dataset of [21] that contains thousands of movie's reviews have been used. For instance, "cok iyi film izleyyyin :)!!!! :) :)", which means "It is very good movie to joyfully watch" is one of the movie reviews. Tokenization process will produce ['cok', 'iyi', 'film', 'keyifle', 'izleyyyin', ':)!!!!', ' :) :)']. In addition, by cleaning the text we will have ['iyi', 'film', 'keyifle', 'izleyyyin']. Then, by applying the process of detecting and correcting the misspelled words so we will have ['iyi', 'film', 'keyifle', 'izleyin']. Finally, stemming produces the review representing vector ['iyi', 'film', 'keyif', 'izley'].

Tokenization also known as text segmentation or lexical analysis is the step of splitting text such as paragraphs into smaller pieces or tokens. Large text sets can be divided into sentences, words, or characters. Cleaning the text by removing stop words, special characters, URLs, and irrelevant text. Hence, such words have no sentiment and do not help the processes of OM.

Detecting and correcting the misspelled words: misspelled word can ruin the understandability of the whole sentence. This indicates the importance of this step. Mainly this step can be done through correcting individual words or correcting the whole sentence. Correcting each word individually can be done by finding the possible candidate words and obtain their polarities from the lexicon, then, select the highest possible candidate. Correcting the word based on the whole sentence is to consider several possible words when correcting a single misspelled word can produce. However, only one of these candidates is correct regarding the context in the sentence. Hence, the correct word can be found using the relations with other words in the sentence.

Stemming is basically a method that finds the root of each word in the processed text. Finally, Normalization consists of multiple sub-steps such as converting all text to the same letter size (upper or lower), converting numbers to word equivalents.

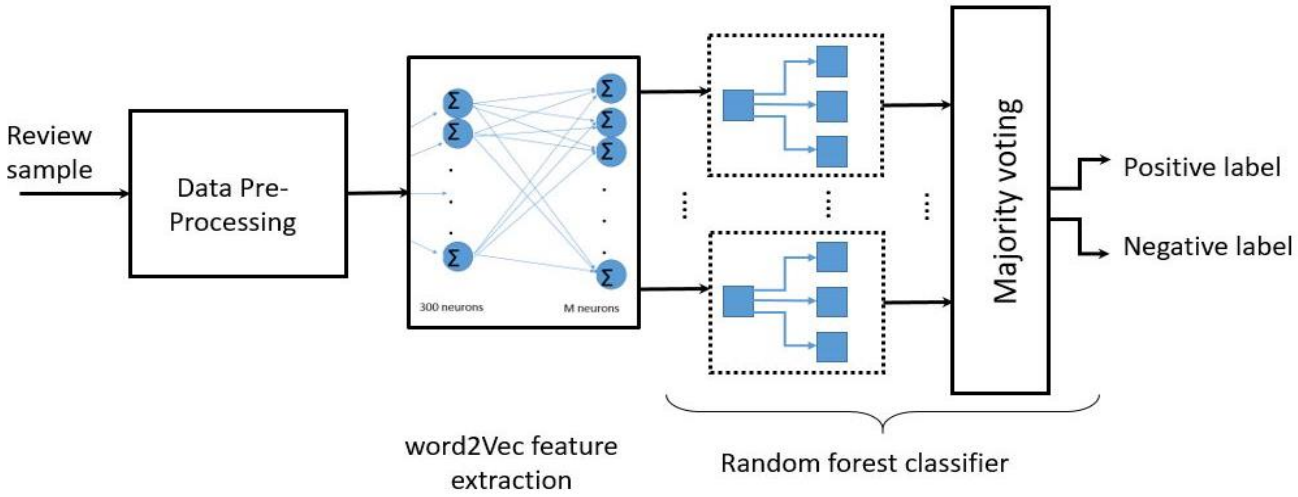


Fig. 1. The main components of the proposed system for both training and testing phases. The input is a review sample while the output is a label. The review sample is preproccessed and then a 300-vector feature is produced by the Word2Vec stage. The feature vector is supplied to each of the decision trees. Decisions of all trees are ensempled using major voting to produce either positive or negative final label.

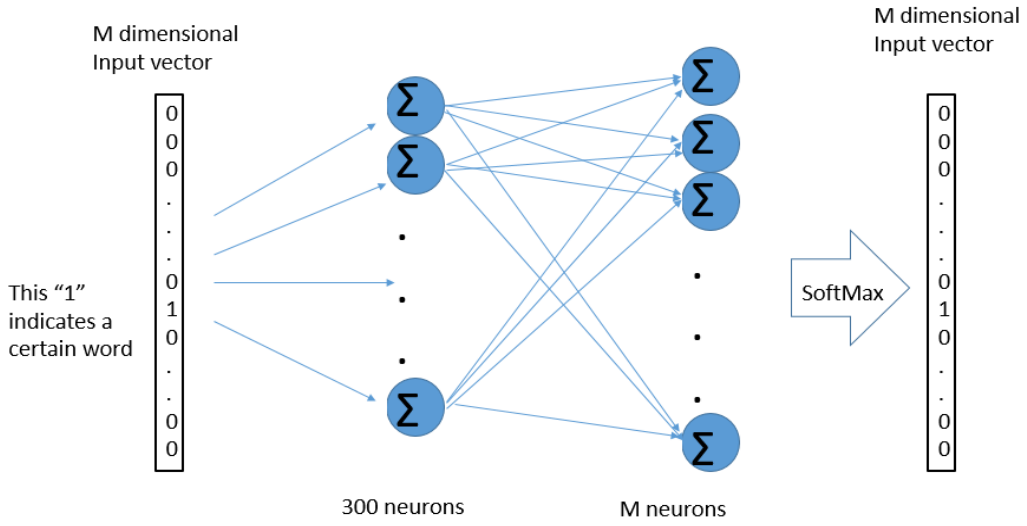


Fig. 2. The Word2Vec feature extraction process. All of the stages indicated in the figure are used in the training phase, the testing phase use the output of the 300 neuron as a Word2Vec and provide them as an input to the RF classifier.

4.2. Feature Extraction

Our data is a textual data and consists of a set of discrete words, which in turn represent a categorical set of features. Hence, we need to map the textual data into real-valued vectors. This process can be done by converting the textual representation of information into a Vector Space Model (VSM).

The vector space model is an algebraic model that converts the text into a vector of words. After that, the words vector is transformed into a numerical format. To represent each element in the vector space, one can use some traditional techniques such as Term Frequency (TF), and Term Frequency-Inverse document frequency (TF-IDF) [31]. In addition, the word embedding representation method can be considered. It is the most commonly used and state of the art technique. The word embedding represents the set of words in a low dimensional vector space, while efficiently preserving the contextual similarity [32]. Its main advantage is that it allows words with similar meaning to have a similar representation. On the other hand, it requires a very large amount of text data (millions) to ensure that useful embeddings are learned. The Word2Vec features [32] and [33], GloVe [34], and FastText [35] are the most popular word embedding approaches.

As presented in [32] and [33], Word2Vec feature extraction consists of two main architectures that can be used for obtaining the distributed representations of words in a corpus. These architectures are: Continuous bag-of-words (CBOW) and Continuous skip-gram (CSG). CBOW predicts the current word using a window of its surrounding words, and it has been proven that the prediction results still the same independent from the order of context words. CSG architecture summarizes the words of each sentence in order to estimate the neighbors around the input word. It assumes that nearby context words are more important than further context words.

We have adopted the Word2Vec feature representation, the CSG in particular, in our work. It has a pre-trained word vector for the English language trained on 1 million common crawl and Wikipedia documents. Word2vec is a two-layer neural net. In our case, its input is the collection of movie reviews and its output is a feature vectors for each word in the collection is represented by a vector of 300 float numbers.

We adopted the CSG architecture in our work. Figure 2 depicts the

processes involved in forming the CSG feature. The Word2Vec architecture feature consists of two layers in addition to the input layer. The figure assumes a vocabulary that contains a set of M words. Here, M is in the scale of 10000 in our application. The input layer contains M neurons connected to the middle layer that contains 300 neurons. Again the middle layer is connected to the output layer with M neurons. After the training is completed every word is represented using a 300-vector of float numbers. This vector is supplied as an input to the RF classifier presented later in the next section. For more details about Word2Vec, the reader is referred to [32] and [33].

4.3. RF based Classification

The RF adopted classifier is depicted in Figure 1 which shows the main components of the developed system. The RF classifier used here consists of a collection of decision trees. This model uses a set of individual decision trees that are relatively uncorrelated and combined together to work as an ensemble. Each of which receives the same input vector from the feature extraction module. The input to each decision tree is the same 300 vector.
In more detail, RF predicts the class of each sample by sending this sample to each tree in the random forest that predicts a class. As the problem is a two class problem, the output of the RF classifier is easily calculate using a majority rule as depicted in Figure 1. Then the class with the most votes becomes the prediction of the model. The RF classifier produces the decision P, stands for positive, which is the decision of majority of decision trees in the forest. RF classifier is implemented using the function "RandomForestClassifier" from the library "sklearn.ensemble". Readers are referred to [36] for more details on DT and RF classifiers.

It can be concluded, as shown in [36], Random Forest is simpler in computation and less sensitive to outliers when compared to other machine learning methods, such as the support vector machine and the artificial neural network.

5. EXPERIMENTS

In this section, we present our experimental works that were carried out to prove the superiority of our proposal. Four experiments were conducted. The first experiment measures the performance of the basic learning methods, i.e. KNN, NB, and SVM. The second experiment compares the performance of different ensemble methods like RF and GBC using conventional features, it includes an ensemble method that combines the three basic classifiers, KNN, NB, and SVM. In the third experiment, the performance of a system that was built based on the RF ensemble learning method and Word2Vec feature is evaluated. The fourth one compares the performance of the proposed method with the one of [17]. The later uses an ensemble of SVM, NB, and Bagging along with the N-gram feature vector, while ours uses RF ensemble classifier along with word2vec features.

These experiments were performed using the dataset of [21]. We have reconstructed four balanced sub-datasets from this dataset. Related to balancing these sub-datasets, balanced means having the same number of positive and negative samples, this process was done by randomly sampling without replacement. This was done by selecting samples and stopping whenever the number of samples in the two-class is the same and the sum of these two groups is equal to the desired number of samples. In addition, the first dataset contains 4000 training and 1000 testing movie reviews. The second dataset contains 8000 training and 2000 testing movie reviews. The third dataset contains 16000 and 4000 training and testing movie reviews respectively, and finally, the fourth dataset contains 25000 and 5000 training and testing movie reviews respectively. It is worth mentioning that our main aim of constricting multiple datasets with different sizes is to investigate the system's robustness and scalability while increasing the number of processed samples. In addition, by testing the system using multiple datasets and using the 10-fold cross-validation, we ensure the accuracy of the results and avoid overfitting.

The following main standard evaluation metrics are used in our experiments. They are well known for evaluating classification systems:

A) *Accuracy* refers to the ratio of the number of correctly classified tweets divide by the total number of tweets

$$\text{Accuracy} = \frac{TP+TN}{\text{N}} \quad (1),$$

B) B) *Precision* refers to the ratio between the correct predictions and the total predictions and can be obtained using Equation (2).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2),$$

C) The *recall* represents the ratio of the correct predictions and the total number of correct tweets in the dataset.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3),$$

D) *F1 score*, a weighted average of Precision and Recall that considers both false positives and false negatives into account

$$F1 = \frac{2*\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4).$$

Where the total number of tweets is represented by N. True Positive (TP) is the number of correctly predicted as positive reviews. True Negative (TN) is the number of correctly Negative reviews. The number of positive reviews which are predicted as negative reviews (False Negative) is represented by FN.

5.1. Experiment 1: Investigating the Performance of the KNN, NB, and SVM

In this experiment, the performance of the mentioned techniques was investigated using the four balanced sub-datasets. Also, in addition to the accuracy, the Precision, Recall, and F1 score, which as mentioned before considers both false positives and false negatives into account has been calculated. Table 1 shows the results of this experiment. The results can be summarized as follows.
1) In general, none of the studied algorithms can be used as the system classifier, as none was able to achieve good performance.
2) Overall, NB achieved the worst performance. In addition, KNN, was the best for the first two datasets, however, as the number of samples was increased in the last two datasets, SVM was able to outperform the others.

5.2. Experiment 2: Investigating the Performance of Ensemble-Based Classifiers

In this experiment, the performance of the ensemble system consisted of KNN, NB and SVM in addition to the Random Forest (RF), AdaBoost Classifier (AdaBoost), and Gradient-Boosting Classifier (GBC) the mentioned techniques were investigated using the four sub-datasets. Also, in addition to the accuracy, the Precision, Recall, and F1 score, which as mentioned before considers both false positives and false negatives into account has been calculated. Table 2 shows the results of this experiment. It is obvious from Tables 1 and 2 that Ensemble-Based Classifiers have

significantly outperformed the performance of individual classifiers. In addition, undebatable the RF is the best choice for the developed system as it has beaten all others for all the datasets. Hence, it is the most stable classifier even when the number of samples was increased.

5.3. Experiment 3: Performance of the Proposed System

In this experiment, based on the results of the previous two experiments, a system that uses the Word2Vec and the RF. The robustness and scalability of this system were investigated using all the constricted datasets. Also, in addition to the accuracy, the Precision, Recall, and the F1 score have been calculated. Figure 3 shows the results of this experiment, and the results can be summarized as follows. 1) Using all the datasets, the developed system achieved very good results, i.e., on average the accuracy, precision, recall, and F1 were above 0.85. 2) Related to the Robustness and Scalability while increasing the number of processed samples, the system showed that it has stability and can handle all the datasets efficiently.

5.4. Experiment 4: Performance of the proposed system vs. other ensemble approach.

In this experiment, the performance of the developed approach was compared to the approach presented in [17] that is the state of the art ensemble work for opinion mining. In general, we have re-implemented this approach which its details are given in section 2. It is worth mentioning that this approach was originally built using Weka, and we have re-implemented using python and we used the GridSearchCV function which is the python version of CVParameterSelection to tune the approach parameters. As mentioned before, the approach of [17], integrates the SVM as the base classifier of the Bagging as the first classifier, while the Naive Bayes and another SVM are used as the second and the third classifiers.

Figure 4 shows the results of the developed approach, individual classifiers used in [17], and the ensemble system presented in [17]. It is clear that the developed approach has outperformed both the approach of [17] and its individual components. In more detail, related to the value of AUC, the developed system was able to get 0.9 for the first dataset, a 0.8 for other datasets, while the approach of [17] has achieved 0.7 for the four datasets. In addition, it can be noticed in Figure 4 that the performance of each of the individual classifiers used in [17] is almost the same as its ensemble system for the used datasets. This means the performance of [17] is sensitive to different datasets as well.

6. Conclusions and Future Works

In this paper, a novel RF-based opinion mining system for movie review application is proposed. The performance of several basic machine learning is tested using the constructed data sets and compared to the proposed one. The experiments have empirically proven that using ensemble learning methods along with the Word2Vec features has considerably outperformed the basic learning methods like the K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB). In addition, using RF along with Word2Vec has superior performance with respect to other ensemble methods like GBC and AdaBoost.

As future work, we plan to work on modifying the Word2Vec features representation to reduce the processing time. In addition, the performance of the feature vector Word2Vec will be improved by training it using a larger dataset and using better object function.

Table 1. The Accuracy, Precision, Recall, and F1 Score for the studied algorithms using four different size datasets.

Dataset	Algorithm	Accuracy	Precision	Recall	F1
1st	KNN	0.5900	0.5512	0.3520	0.4253
	NB	0.1990	0.5707	0.1650	0.2008
	SVM	0.4390	0.5718	0.2170	0.2962
2nd	KNN	0.5125	0.5553	0.2955	0.3717
	NB	0.1830	0.5695	0.1545	0.1680
	SVM	0.4910	0.5659	0.2625	0.3307
3rd	KNN	0.4987	0.5519	0.2752	0.3516
	NB	0.1887	0.5704	0.1530	0.1697
	SVM	0.5990	0.5714	0.3415	0.4155
4th	KNN	0.5218	0.5455	0.2922	0.3649
	NB	0.1966	0.5665	0.1542	0.1727
	SVM	0.6564	0.5720	0.3772	0.4481

Table 2. The Accuracy, Precision, Recall, and F1 Score for the studied ensemble systems using four different size datasets.

Dataset	System	Accuracy	Precision	Recall	F1
1st	{ KNN, NB and SVM}	0.5930	0.5679	0.3170	0.3993
	RF	**0.8350**	**0.7771**	**0.8350**	**0.8350**
	AdaBoost	0.7765	0.7771	0.7765	0.7764
	GBC	0.7935	0.7935	0.7937	0.7935
2nd	{ KNN, NB and SVM}	0.5600	0.5669	0.3040	0.3789
	RF	**0.8350**	**0.7771**	**0.8350**	**0.8350**
	AdaBoost	0.8200	0.8200	0.8202	0.8200
	GBC	0.7765	0.7771	0.7765	0.7764
3rd	{ KNN, NB and SVM}	0.5693	0.5713	0.3185	0.3944
	RF	**0.8340**	**0.7826**	**0.8340**	**0.8340**
	AdaBoost	0.8115	0.8112	0.8133	0.8115
	GBC	0.7825	0.7826	0.7825	0.7825
4th	{ KNN, NB and SVM}	0.5950	0.5705	0.3312	0.4086
	RF	**0.8526**	**0.7976**	**0.8526**	**0.8526**
	AdaBoost	0.8192	0.8190	0.8207	0.8192
	GBC	0.7974	0.7976	0.7974	0.7974

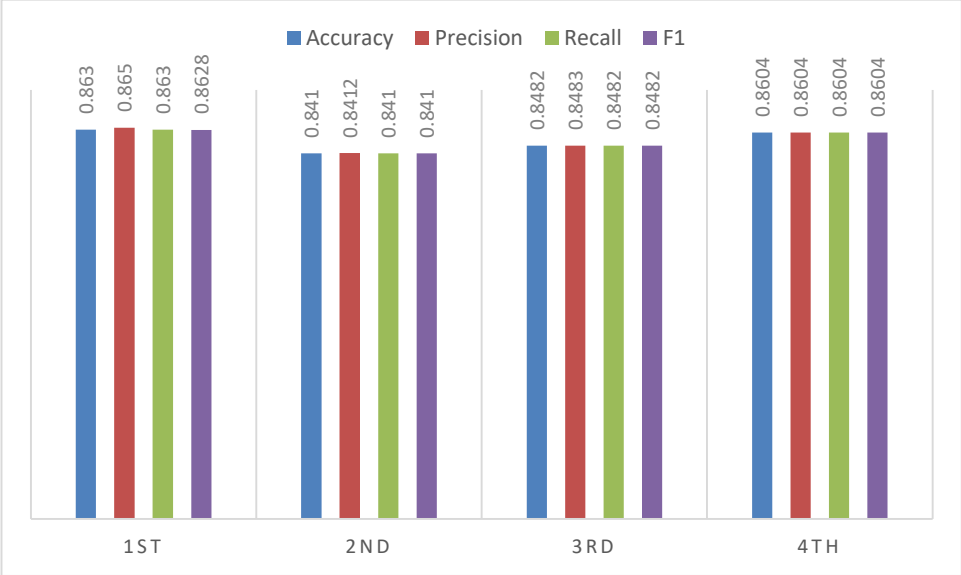


Fig . 3. The Average of Accuracy, Precision, Recall, and F1 for the developed system that uses the Word2Vec and the RF. It is clear that the developed system is able to achieve quite good results using all the datasets, i.e., on average the accuracy, precision, recall, and F1 were above 0.85.

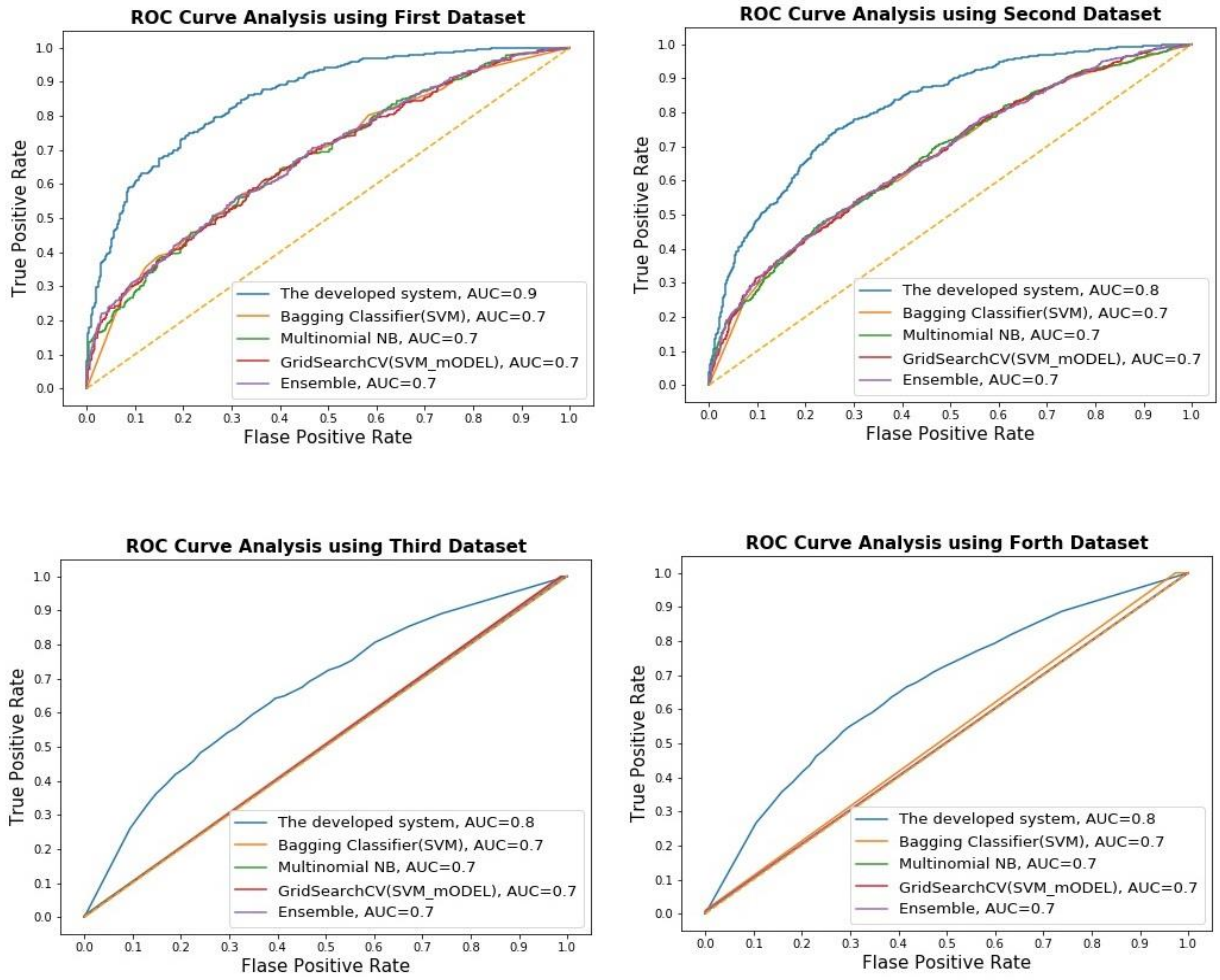


Fig. 4. Performance of the proposed system vs. the approach of [17]. The figure clearly shows that the developed approach has outperformed the approach of [17] and the other three individual classifiers used in [17], i.e., the SVM as the base classifier of the Bagging, the Naive Bayes and a second copy of the SVM using all the constricted datasets.

References

[1] Social Media Examiner, "2018 Social media marketing industry report", Social media examiner, 2019 [Online]. Available: http://www.socialmediaexaminer.com/ report2016/.[Accessed: 20.1.2019]

[2] Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In mining text data. Springer, Boston, MA, 415-463.

[3] Mäntylä, M. V., Graziotin, D., & Kuutila, M. (2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers.Computer Science Review, 27, 16-32.

[4] Pradhan, V. M., Vala, J., & Balani, P. (2016). A survey on Sentiment Analysis Algorithms for opinion mining. International Journal of Computer Applications, 133(9), 7-11.

[5] Bing Liu. (May 2012). Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers.

[6] Riaz, S., Fatima, M., Kamran, M., & Nisar, M. W., "Opinion mining on large scale data using sentiment analysis and k-means clustering", Cluster Computing, 22(3), pp.7149-7164, 2019.

[7] Aishwarya, R., et al, "A Novel Adaptable Approach for Sentiment Analysis", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5 (2), 2019.

[8] Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2015). Attitude Sensing in Text Based on A Compositional Linguistic Approach. Computational Intelligence, 31(2), 256–300.

[9] Esuli, A., & Sebastiani, F. (2006, May). Sentiwordnet: A publicly available lexical resource for opinion mining. In LREC, 6, 417-422.

[10] Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In Lrec 10 (2010), 2200-2204.

[11] Korovkinas, K., Danėnas, P., & Garšva, G. (2017). SVM and Naïve Bayes Classification Ensemble Method for Sentiment Analysis. Baltic Journal of Modern Computing, 5(4), 398-409.

[12] Eroğul, U. (2009). Sentiment analysis in Turkish. Master's thesis. Middle East Technical University, Ankara.

[13] Dehkharghani, R., Yanikoglu, B., Saygin, Y., & Oflazer, K. (2017). Sentiment analysis in Turkish at different granularity levels. Natural Language Engineering, 23(4), 535-559.

[14] Vural, A. G., Cambazoglu, B. B., Senkul, P., & Tokgoz, Z. O. (2013). A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish. In Computer and Information Sciences III (pp. 437-445). Springer, London.

[15] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2011). Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology, 62(2), 419.

[16] Türkmenoglu, C., & Tantug, A. C. (2014, June). Sentiment analysis in Turkish media. In Proceedings of Workshop on Issues of Sentiment Discovery and Opinion Mining, International Conference on Machine Learning (ICML), Beijing, China.

[17] Catal, C. and Nangir, M., 2017. A sentiment classification model based on multiple classifiers. Applied Soft Computing, 50, pp.135-141.

[18] Shehu, H. A., & Tokat, S. (2019, April). A hybrid approach for the sentiment analysis of Turkish Twitter data. In The International Conference on Artificial Intelligence and Applied Mathematics in Engineering (pp. 182-190). Springer, Cham.

[19] Dehkharghani, R., Saygin, Y., Yanikoglu, B., & Oflazer, K. (2016). SentiTurkNet: a Turkish polarity lexicon for sentiment analysis. Language Resources and Evaluation, 50(3), 667-685.

[20] Demirtas, E., & Pechenizkiy, M. (2013, August). Cross-lingual polarity detection with machine translation. In Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining (p. 9). ACM.

[21] Ucan, A., Naderalvojoud, B., Sezer, E. A., & Sever, H. (2016, January). SentiWordNet for new language: automatic translation approach. In 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 308-315). IEEE.

[22] CUNNINGHAM, Padraig; DELANY, Sarah Jane. k-Nearest neighbour classifiers. Multiple Classifier Systems, 2007, 34.8: 1-17.

[23] NIKHATH, A. Kousar; SUBRAHMANYAM, K.; VASAVI, R. Building a K-Nearest Neighbor Classifier for Text Categorization. International Journal of Computer Science and Information Technologies, 2016, 7.1: 254-256.

[24] FRANK, Eibe; BOUCKAERT, Remco R. Naive bayes for text classification with unbalanced classes. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, 2006. p. 503-510.

[25] DIETTERICH, Thomas G. Ensemble methods in machine learning. In: International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000. p. 1-15.

[26] DADGAR, Seyyed Mohammad Hossein; ARAGHI, Mohammad Shirzad; FARAHANI, Morteza Mastery. A novel text mining approach based on TF-IDF and Support Vector Machine for news classification. In: 2016 IEEE International Conference on Engineering and Technology (ICETECH). IEEE, 2016. p. 112-116.

[27] ONAN, Aytuğ; KORUKOĞLU, Serdar; BULUT, Hasan. Ensemble of keyword extraction methods and classifiers in text classification. Expert Systems with Applications, 2016, 57: 232-247.

[28] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[29] RODRIGUEZ, Juan José; KUNCHEVA, Ludmila I.; ALONSO, Carlos J. Rotation forest: A new classifier ensemble method. IEEE transactions on pattern analysis and machine intelligence, 2006, 28.10: 1619-1630.

[30] FRIEDMAN, Jerome H. Stochastic gradient boosting. Computational statistics & data analysis, 2002, 38.4: 367-378.

[31] C Hans, M Agus, and D Suhartono,"Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)," ComTech: Computer, Mathematics and Engineering Applications, vol. 7, no. 4, pp. 285-294, 2016.

[32] Mikolov, T., Chen, K., Corrado, G. S., Dean, J., Sutskever, L., & Zweig, G. (2013). word2vec. URL https://code.google.com/p/word2vec.

[33] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. arXiv preprint arXiv:1712.09405.

[34] P Jeffrey, R Socher, and C Manning,"Glove: Global vectors for word representation," In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.

[35] J Armand, E Grave, P Bojanowski, M Douze, H Jégou, and T Mikolov,"Fasttext. Zip: Compressing text classification models," arXiv preprint arXiv: 1612.03651, 2016.

[36] Rodriguez-Galiano, V. F., Chica-Olmo, M., Abarca-Hernandez, F., Atkinson, P. M., & Jeganathan, C. (2012). Random Forest classification of Mediterranean land cover using multi-seasonal imagery and multi-seasonal texture. Remote Sensing of Environment, 121, 93-107.