Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 122-129, Ağustos 2020
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 122-129, August 2020
Copyright © 2020 EJOSAT
Research Article

Turkish Sentiment Analysis System via Ensemble Learning

Saed Alqaraleh

Hassan Kalyoncu University, Computer Engineering Department, GaziAntep, Turkey (ORCID: 0000-0002-7146-3905)

(Bu yayın 26-27 Haziran 2020 tarihinde HORA-2020 kongresinde sözlü olarak sunulmuştur.)

(**DOI:** 10.31590/ejosat.779181)

ATIF/REFERENCE: Alqaraleh, S. (2020). Turkish Sentiment Analysis System via Ensemble Learning. *European Journal of Science and Technology*, (Special Issue), 122-129.

Abstract

Nowadays, sentiment analysis (SA) also known as opinion mining (OM) is widely used and has an impressive effect in many fields, such as marketing, politics, and even, company's products are now adjusted based on users' opinions. In this paper, a new efficient sentiment analysis system that supports the Turkish language has been introduced. In addition, as Turkish is an agglutinative language, which requires special processing, an efficient preprocessing model was also implemented as a part of the developed system.

Several experiments using the challenging and benchmark "The Turkish movie reviews" dataset have been conducted, and it is obvious that the constructed approach can efficiently support the Turkish language and can achieve a quite good performance.

Keywords: Sentiment Analysis; Text Classification; Opinion Mining; Turkish Language; Ensemble learning; Natural Language Processing.

1. Introduction

Due to the rapid growth of the Internet, it is very common to share our opinions about government policy, people, products, movies, hotels, research, etc. Transferring the opinions into useful information (knowledge) is known as sentiment analysis (SA). In other words, sentiment analysis is the process of discovering the feelings, opinions, and subjectivity behind the text.

In general, SA is a set of techniques that process the opinions (usually text) to find out whether it illustrates a positive or negative sentiment [2]. The main approaches used in SA tasks are machine learning and Lexicon-based [3, 4], related to the first type, if labeled dataset(s) are used for training the classifiers it is referred to as supervised machine learning approaches. On the other hand, unsupervised machine learning methods work on unlabeled datasets. Related to the lexicon-based approach, it uses seed words, which is a set of predefined words or phrases where each one has a membership score for each of the three related classes(categorize), i.e., positive, negative and neutral[5].

1.1 SA Challenging Problems

One of the main well-known SA problems is that most of the current SA techniques and studies have been done on a limited number of languages such as English and chines. These studies have shown that the techniques used for SA changed from a language to others. For instance, the sentence-based SA systems are more efficient and fast for English, however, the character-based SA systems are more suitable and efficient for chains. In addition, most sentiment analysis resources (e.g., polarity lexicons, parsers) are well established for English while, due to lack of resources and complexity, very few researches have been done for other languages especially the agglutinative morphology languages such as Turkish, French, and Korean. As a result, most of the researches for such Languages are not mature yet, and even some have been done using the English sentiment analysis resources. For instance, in [6], French movie reviews have been classified by using linguistic features and supervised learning through translating the reviews French words to English to obtain its semantic orientation using the SentiWordNet [7].

Related to the Turkish language, sentiment analysis attracts researchers interest in the last decade, however, most of the existing studies are preliminary, and there is still a lot to do in the field. In the following, some details about challenging problems related to SA for supporting the Turkish language are summarized:

1) Turkish is one of the agglutinative languages: Some new words can be generated by adding suffixes to the root word, and these suffixes may change the semantic orientation of the word. Overall, building a lexicon of all variants of the Turkish words is practically limited and a hard task.

2) Negations: Multiple ways can negate the Turkish words. For instance, using words such as “değil" or “yok", or with the affixes siz/sız or me/ma, where the sentiment polarity might be changed from a way to others.

3) Turkish Alphabet: The Turkish language has some characters that English does not have, i.e., “ğ”, “ç”, “ı", “ö”, “ş”, and “ü”. However, people are used to substituting these characters by the closest ASCII ones, where “s” is used instead of “ş “and “o" instead of “ö”, etc. Overall, the semantic analysis of Turkish is more risky to suffer from erroneous writings.

2. Related Work

In this section, the recent researches, developments, and solutions related to the proposed system have been summarized. As mentioned before Turkish language sentiment analysis has attracted research interest in recent years.

In [8], which is considered as one of the first studies in Turkish sentiment analysis. The Turkish reviews have been classified using an SVM classifier and n-grams. In addition, they have studied the effect of part-of-speech tagging, spell-checking, and stemming. Overall, around 85% accuracy was obtained on the binary sentiment classification by the approach of [8].

In [9], a Turkish lexicon-based SA system has been proposed. This system uses the SentiStrength which is a lexicon-based library that calculates a sentiment score for each word in a given text [10]. The approach of [9] was tested using the same dataset of [8], and achieved an accuracy of 76%.

Another lexicon-based framework that can handle the simple negation and multi-word expressions also introduced in [11]. This system has achieved 79.0% and 75.2% accuracy on a movie and Twitter datasets, respectively.

In [12], a Turkish sentiment analysis system using three different levels (aspect, sentence, and document) was introduced. In addition, some linguistic issues such as intensification and conjunction were investigated. Overall, accuracies range from 60% to 79% for both ternary and binary classification tasks were reported in [12].

In [13], a comparison of Lexicon based and Machine Learning-based sentiment analysis methods on Turkish social media was performed. In addition, their lexicon was formed with 1) an English opinion lexicon (translated to Turkish), and 2) multi-words expressions. They applied both approaches for binary (positive/negative) classification, and then, these two approaches were evaluated. The results showed that Machine Learning based sentiment analysis gives a good performance, however, lexicon-based sentiment analysis is still preferable in some cases.

In addition, in [14] and [15], the performances of maximum entropy, Naive Bayes, SVM, and the character n-gram were investigated using a Turkish political news dataset. Results of [14] showed that maximum entropy and the n-gram achieved 76-77% as accuracy and outperformed other techniques. In [15], the work of [14] was further improved by implementing transfer learning into the existing framework.

3. The proposed System

In the following the mechanism and components of the developed SA system that fully supports the Turkish language are explained in details:

3.1 Preprocessing

Preprocessing is critical in terms of classifier performance and it works on improving the quality of the used data and eliminating useless information through some steps such as:

1) Tokenization: also known as text segmentation or lexical analysis, it works on splitting longer text strings into smaller pieces such as sentences, words, or characters. In this work, tokenization is used to split the text data into words.

2) Cleaning, i.e., removing stop words, special characters, URLs, and irrelevant text. In addition, we have tested the effect of a series of other related tasks such as converting numbers to word equivalents, case folding, removing punctuation, etc.

3) 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 a) Correcting each word individually: de-ASCIIfication, which converts the ASCII of the English characters to the equivalents Turkish one, to find the possible candidate words has been investigated in this study. b) Correcting the word based on the whole sentence: In general, correcting a single misspelled word can produce several possible words, 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. However, the implementation of the second type is kept as future work.

4) Stemming: In general, the morphological structure of the Turkish language is very rich, and there are over fifty suffixes that can be affixed to verbs [17], and multiple suffixes can be used at once. Hence, we can have different words from a single stem when appending a sequence of suffixes. For instance, the word "Geliyormuşsunuz" means "You had been coming". The stem of the word is "gel-" and it takes three different suffixes -(i)yor, -muş, and -sunuz. In [17], an efficient stemming algorithm for the Turkish was developed, which has been integrated into our developed approach.

5) Detection of negation: as stated in [8] and [18], related to the Turkish language, this step needs extra treatment as the negation may be realized within the word with affixation rather than a separate individual word.

3.2 Feature Extraction

In general, text represents categorical and discrete features. Hence, we need to map the textual data into real-valued vectors. This process can be done by representing the textual information using the Vector Space Model (VSM). VSM is a model that converts the text into a vector of words and then transforms the words vector into a numerical format. This process consisted of the following steps:

First: create a dictionary of terms presented in the text collection (in our case it is the dataset of movie review). Briefly, all terms from the collection are alphabetically sorted in the vector space, where each word has a unique id. It worth mentioning that this process is done after preprocessing all the movie reviews. Hence, all irrelevant and stop words have been deleted previously.

Second, obtaining the representation of each term, which will be added to the vector space. The following are some of the methods that can be used to represent text terms in the vector space:

1) Term Frequency (TF): It refers to the frequency of the document's terms (in our case it represents the frequency of each term in the vocabularies of the created vector of words). Term frequency can be calculated using Equation 1.

$$\text{TF } (W_i) = \frac{O(W_i)}{N} \qquad \text{Equation (1)}$$

Where $O(W_i)$ is the number of occurrence of the i^th word in the previously created VSM, and N is the total number of words existed in the VSM.

2) Term Frequency-Inverse document frequency (TF-IDF): Generally, the TF method assigns the highest score to the most frequent words. In other words, the highest score is assigned to the word that occurs frequently in the VSM. On the other hand, the IDF measures how important a term is. Hence, the frequent words are not always the important ones, for instance, certain terms, such as "Yapacak", "Ederim", "Hatta", "olduklarını", etc., appear a lot of times in most texts, but they have little importance. Hence, in the case of IDF, the highest score is assigned to rare words, and a low score is assigned to the frequent words. Inverse document frequency can be calculated using Equation 2.

$$\text{IDF } (W_i) = \log \frac{N}{T} \qquad \text{Equation (2)}$$

Where T refers to the number of movie reviews that contain the i^th word. Finally, the TF-IDF can be obtained by multiplication of TF and IDF values:

$$\text{TF-IDF } (W_i) = \text{TF } (W_i) * \text{ IDF } (W_i) \qquad \text{Equation (3)}$$

3) Word Embedding: It is very popular as it can efficiently preserve the contextual similarity while representing the words in low dimensional vector space [4], [7]. In addition, word embedding produces a similar representation for words that have a similar meaning. However, to achieve such advantages a very large dataset(s) must be used in the training process. Word2Vec [19 and 20], GloVe [21], and FastText [22] are examples of the popular embedding approaches. In this paper, we have used the Word2Vec that was trained on one million Turkish common crawls and Wikipedia documents. In this study, the input of Word2Vec is the movie reviews and a feature vector for each word in the collection is produced as its output (shown in Equation (4)).

$$\text{Word2vec } (W_i) = [F_1, F_2, F_3, \ldots\ldots\ldots F_m] \qquad \text{Equation (4)}$$

Where, m is set to 300, and F is a float number. Hence, a vector of 300 float numbers is produced to represent each word. For more details about Word2Vec, the reader is referred to [19] and [20].

3.3 Classification

In this work, we have investigated the performance of AdaBoost, Random Forest, and GradientBoosting, which are well-known ensemble learning approaches and can be used as the system classifier, i.e., classifying the movie reviews into positive or negative sentiments classes. The details about these approaches are summarized below.

I. AdaBoost Classifier (AdaBoost)

AdaBoost or Adaptive Boost is an ensemble classifier. It is an iterative ensemble method that uses the weak classifiers within the ensemble structure in order to boost its performance. In this case, the ensemble's classifiers are added one at a time, where each subsequent classifier is trained using the data that previous ensemble members have failed to classify it correctly. In other words, AdaBoost selects the training set for trains the current learning model based on the estimation of the last training [23-25].

II. Random Forest (RF)

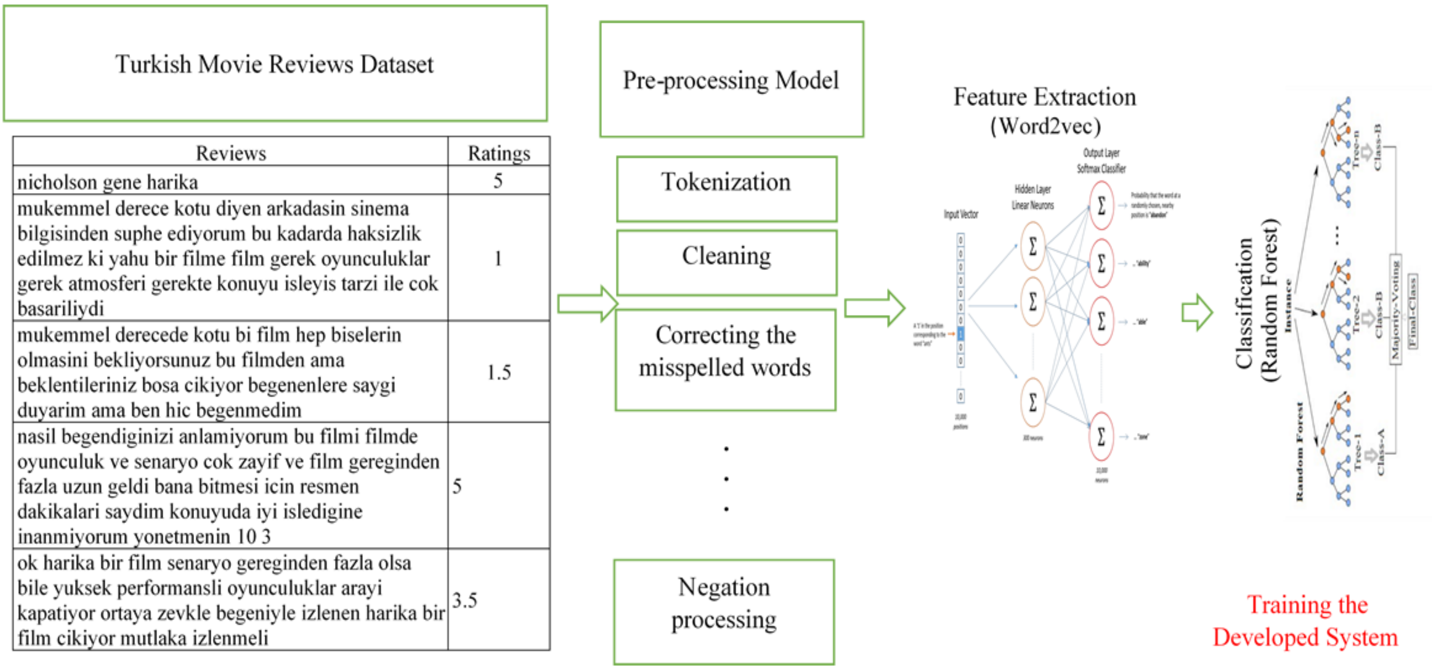
Similar to the AdaBoost, RF is another ensemble classifier that is based on combining multiple decision tree models [23-25]. In other words, RF is an ensemble that contains some relatively uncorrelated individual decision trees working together, where each data sample is passed to all the trees to predicts its class, and the class with the most votes is selected as the prediction of the RF model.

III. GradientBoosting Classifier (GBC)

In general, the prediction model of the GBC classifier is produced by sequentially fitting the base learner to current “pseudo”-residuals, which is the gradient of the loss functional being minimized in respect to the model values for each training sample evaluated at the current step [23-25].

3.4 Structure of the Developed System

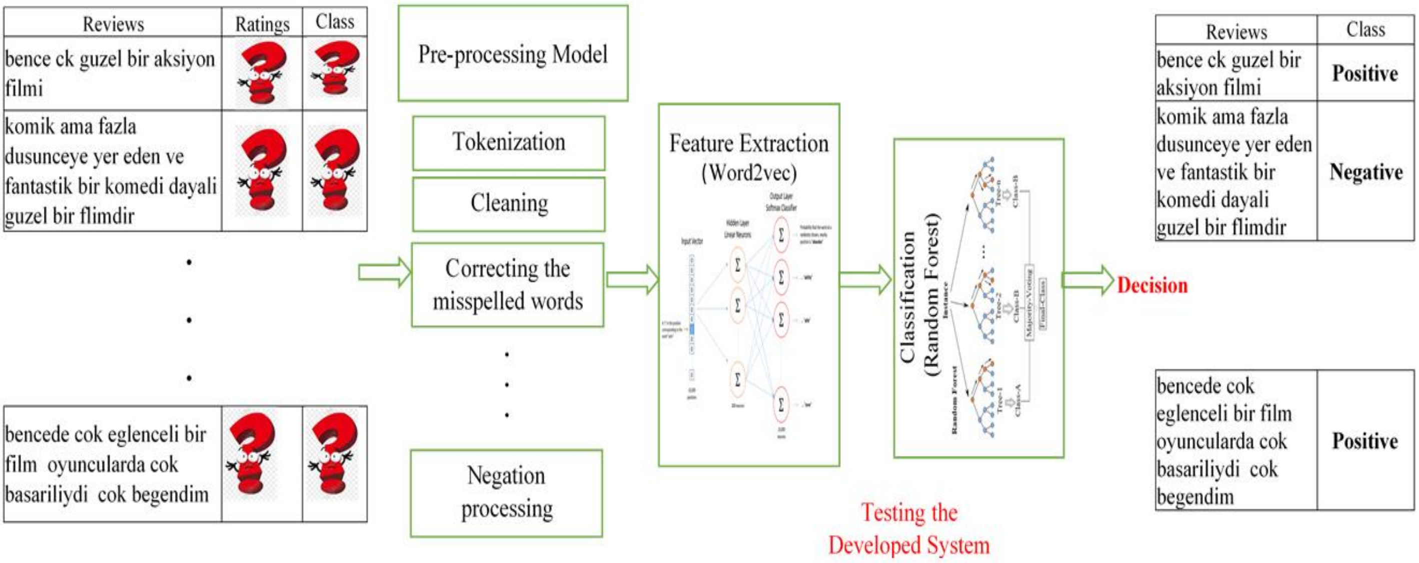
In general, the developed approach consists of the developed pre-processing model followed by the word2vec embedding system, then the random forest ensemble approach is used as a classifier. More details about the developed system are shown in Figures 1 and 2.



Turkish Movie Reviews Dataset

Reviews	Ratings
nicholson gene harika	5
mukemmel derece kotu diyen arkadasin sinema bilgisinden suphe ediyorum bu kadarda haksizlik edilmez ki yahu bir filme film gerek oyunculuklar gerek atmosferi gerekte konuyu isleyis tarzi ile cok basariliydi	1
mukemmel derecede kotu bi film hep biselerin olmasini bekliyorsunuz bu filmden ama beklentileriniz bosa cikiyor begenenlere saygi duyarim ama ben hic begenmedim	1.5
nasil begendiginizi anlamiyorum bu filmi filmde oyunculuk ve senaryo cok zayif ve film gereginden fazla uzun geldi bana bitmesi icin resmen dakikalari saydim konuyuda iyi isledigine inanmiyorum yonetmenin 10 3	5
ok harika bir film senaryo gereginden fazla olsa bile yuksek performansli oyuncuklar arayi kapatiyor ortaya zevkle begeniyle izlenen harika bir film cikiyor mutlaka izlenmeli	3.5

Figure 1. Main Components while Training Process of the Developed System.



Reviews	Ratings	Class
bence ck guzel bir aksiyon filmi	?	?
komik ama fazla dusunceye yer eden ve fantastik bir komedi dayali guzel bir flimdir	?	?
. . .		
bencede cok eglenceli bir film oyuncularda cok basariliydi cok begendim	?	?

Reviews	Class
bence ck guzel bir aksiyon filmi	**Positive**
komik ama fazla dusunceye yer eden ve fantastik bir komedi dayali guzel bir flimdir	**Negative**
bencede cok eglenceli bir film oyuncularda cok basariliydi cok begendim	**Positive**

Figure 2. The Mechanism of Testing and/or using the System in Real-Time.

3. Experimental Work

In this section, multiple experiments to evaluate the main components of the developed system were performed. In addition, as shown in the following sub-section, multiple databases that we used to ensure the robustness of the obtained results, have been constructed and used in this study. It is important to note the best value of the “number of estimators” for the RF classifier has been set using the Grid Search, and the tested values were {50, 100, 150, and 200}.

A. Datasets and Evaluation

In this work, the last version of the Turkish movie reviews dataset introduced in [16], which is publicly available and composed of 34990 positive and negative movie reviews in Turkish is used. In more detail, the movie reviews were collected from "Beyazperde", which is a movie site that allow users to write reviews about movies and give ratings of 0 to 5. In this dataset, the ground-truth (labels for evaluation) is represented by the ratings of the reviews. It is worth mentioning that in our work, reviews of ratings >3 are taken as positive reviews; reviews of rating <3 are considered negative, and reviews with ratings equal to 3 are considered as neutral/objective. In addition, we have reconstructed four datasets from this dataset, the first dataset contains 3000 positive and 3000 negative movie reviews, the second dataset contains 5000 positive and 5000 negative movie reviews, the third dataset contains 10000 positive and 10000 negative movie reviews and the fourth dataset contains all the reviews, i.e., 34990.

Related to the evaluation, we have used the following main standard metrics that are well known used for evaluating classification systems:

1) Accuracy, which represents the ratio of correctly classified movie reviews and can be obtained using Equation (5).

2) Precision, can be calculated using Equation (6), and it is the ratio between the correct predictions and the total predictions.

3) Recall obtains the ratio of the correct database samples that were identified correctly and can be obtained using Equation (7).

4) F1 score, which is a weighted average of Recall and Precision can be observed using Equation (8).

$$\text{Accuracy} = \frac{TP+TN}{\text{Total}} \quad \text{Equation (5)}, \qquad \text{Precision} = \frac{TP}{TP+FP} \quad \text{Equation (6)}$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{Equation (7)}, \qquad \text{F1 Score} = \frac{2*\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad \text{Equation (8)}$$

B. Experiments and Results Analysis

Experiment 1: Performance of the RF, AdaBoost, and GBC

In this experiment, we have investigated the performance of each of the studied techniques, i.e., the RF, AdaBoost, and GBC. The results of this experiment can be seen in Figure 3. Overall, the performance of all studied techniques for processing the Turkish language was not good enough (at most 70% as accuracy). On the other hand, the results using all the datasets indicated that the Random Forest outperformed the others.

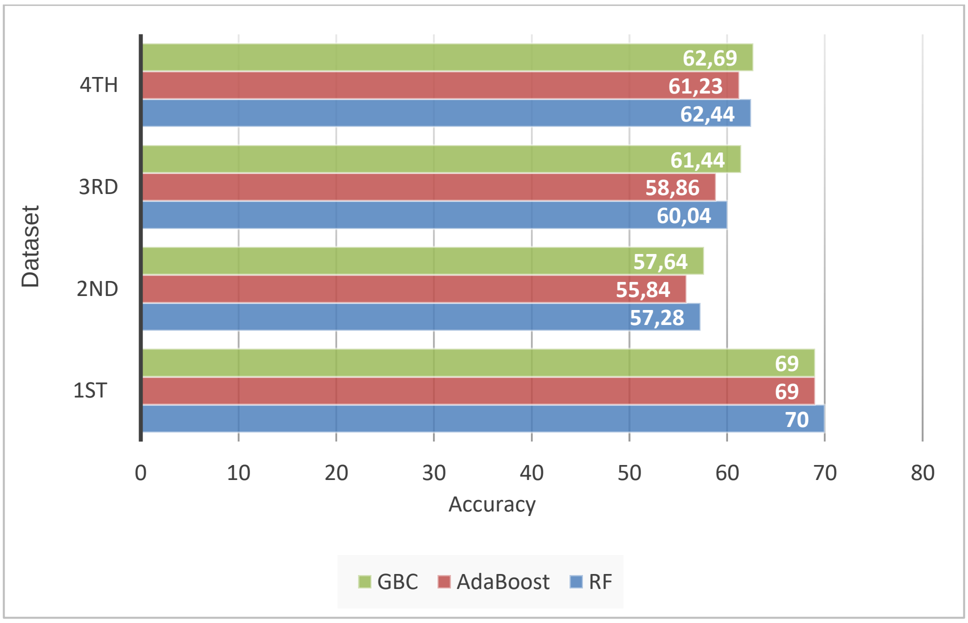


Figure 3. Performance of the Random Forest (RF), AdaBoost Classifier (AdaBoost), GradientBoosting Classifier (GBC).

Experiment #2: The Effect of Pre-Processing the Turkish Movie Reviews

In this experiment, we have investigated the effect of pre-processing the Turkish movie reviews. In general, it is expected that pre-processing will always improve the performance of any system, however, this process requires extra time. Hence, the decision of whether to use or not using the pre-processing model can be taken based on the percentage of performance improvement. The accuracy after integrating the developed preprocessing model with the studied classifiers is shown in Figure 4, and Figure 5 shows the improvement percentage for the studied algorithms using the developed pre-processing model. As shown in Figures 4 and 5, it is clear that the developed pre-processing model has significantly improved the performance of all the techniques, and on average the improvement percentage was around 30%. As a result, by introducing this model we are able to process and build an efficient approach for the Turkish language.

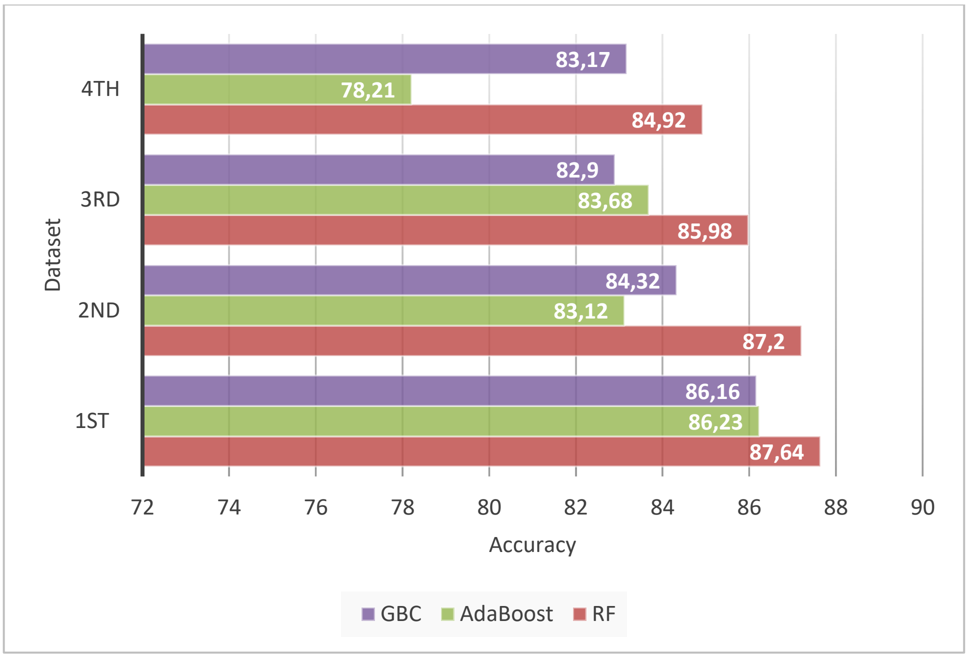


Figure 4. The Accuracy of the Studied Classifiers when the Developed Pre-processing Model is Used.

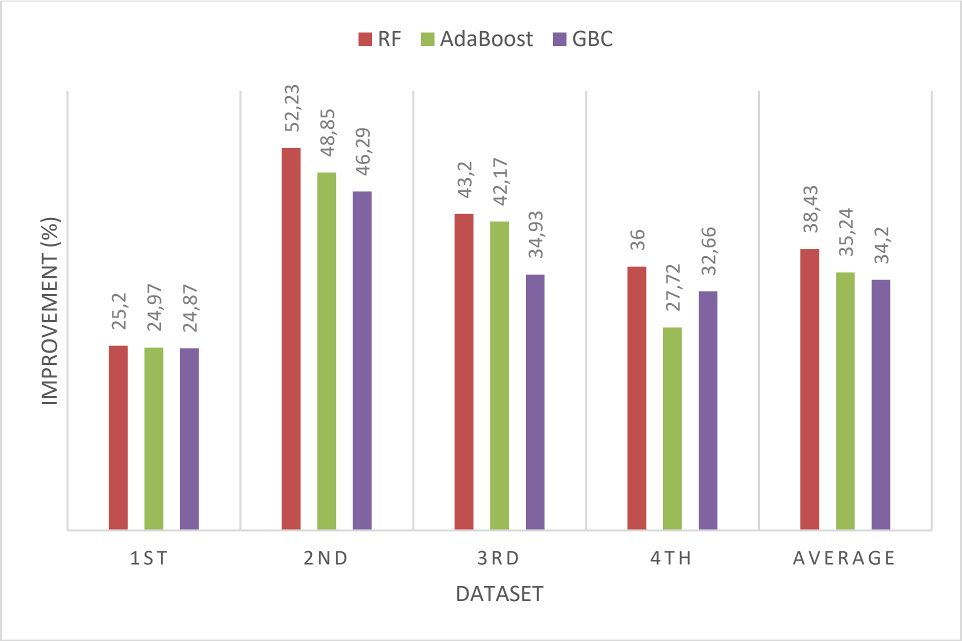


Figure 5. Percentage of Performance Improvement for the Studied Classifiers when Using the Developed Pre-Processing Model.

Experiment 3: Finding the Suitable Feature Extraction Method for the Turkish Language

In this experiment, the performance of the Term Frequency (TF), Term Frequency-Inverse document frequency (TF-IDF) and Word2Vec methods that can be used to represent the text in the vector space was studied. In more detail, each of these methods was integrated into the developed system and its performance was investigated to find out which one is more suitable for the Turkish language. As shown in Table 1, as expected with the improvement that was introduced by word embedding approaches, we have proven that Word2Vec is the most appropriate approach compared to the other two for building an effacing system for processing the Turkish language.

Table 1. Accuracy of TF, TF-IDF, and Word2Vec approaches when processing the Turkish language.

Dataset	***Accuracy***		
	TF	***TF-IDF***	***Word2Vec***
1st Dataset	*69.91*	*66.75*	***87.50***
2nd Dataset	*66.72*	*65.64*	***86.8***
3rd Dataset	*69.8*	*69.86*	***85.86***
4th Dataset	*65.4*	*65.25*	***83.93***

Experiment 4: The Overall Performance, Robustness and Scalability of the Developed Approach

In this experiment, both robustness and scalability of the developed system were investigated using all the constructed datasets by obtaining the Accuracy, Precision, Recall, and F1 score.

As depicted in Figure 6, the main results of this experiment are:

1. Using all the datasets, the developed system achieved very good results, i.e., on average the accuracy, precision, recall, and F1 were above 86, 87, 85, and 86 respectively.
2. The system showed that it has a stable performance while processing all the datasets, even when the number of processed samples is increased.

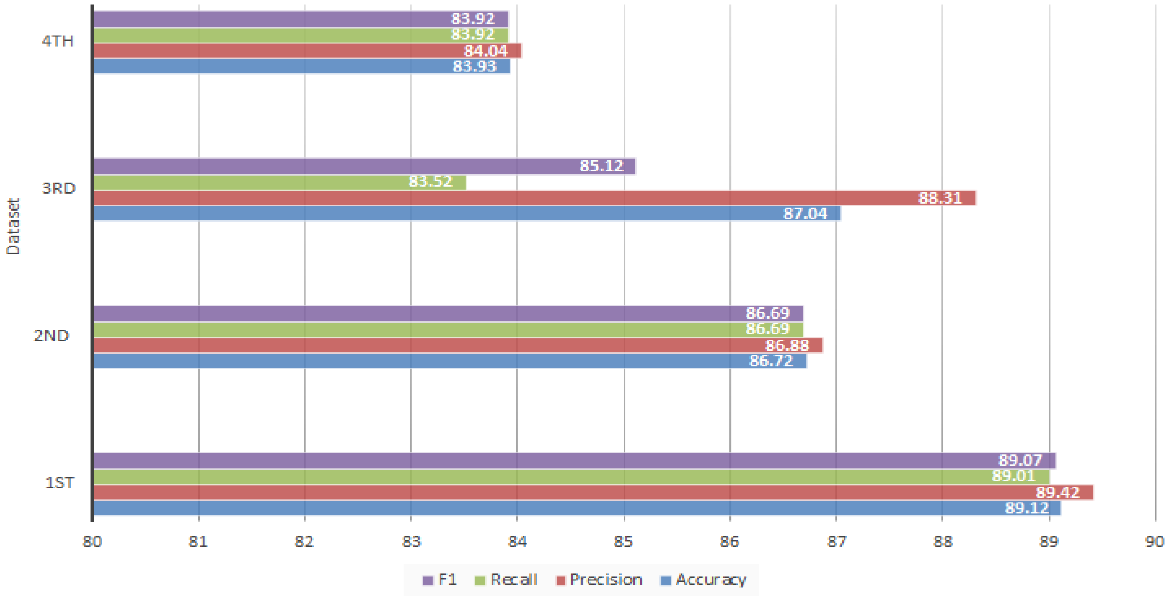


Figure 6. The accuracy, Precision, Recall and F1 score of the Developed Approach using All the Datasets.

4. Conclusions and Future Works

Nowadays, social media analytics is an important research field and has led to improve internet-based systems. In this work, an efficient sentiment analysis system for the Turkish language was developed. The system was built after investigating the performance of some well-known ensemble systems, i.e., Random Forest (RF), AdaBoost Classifier (AdaBoost), GradientBoosting Classifier (GBC). Results show that preprocessing the text is a must step. In addition, the developed pre-processing model was able to significantly improve the performance of all the studied classifiers. Furthermore, it has been shown that the proposed system has a stable performance while processing all the datasets.

As future work, this study can be expanded in many directions starting by taking advantage of existing CNN models which can be integrated and considered as very good choices for improving sentiment analysis systems. A second direction is to investigate the performance of the state-of-the-art word embedding approaches such as BERT, ElMo, and XLNet when used for the Turkish language.

References

[1] Social Media Examiner, "2019 Social Media Marketing Industry Report ", 2020 [Online]. Available: https://www.socialmediaexaminer.com/social-media-marketing-industry-report-2019,[Accessed: 20.1.2020].

[2] Hussein, D. M. E. D. M. (2018). A survey on sentiment analysis challenges. Journal of King Saud University-Engineering Sciences, 30(4), 330-338.

[3] Kaur, H., & Mangat, V. (2017, February). A survey of sentiment analysis techniques. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 921-925). IEEE.

[4] Wang, H., & Zhai, C. (2017). Generative models for sentiment analysis and opinion mining. In A practical guide to sentiment analysis (pp. 107-134). Springer, Cham.

[5] Liu, R., Shi, Y., Ji, C., & Jia, M. (2019). A Survey of Sentiment Analysis Based on Transfer Learning. IEEE Access, 7, 85401-85412.

[6] Ghorbel, H., & Jacot, D. (2011). Sentiment analysis of French movie reviews. In Advances in Distributed Agent-Based Retrieval Tools (pp. 97-108). Springer, Berlin, Heidelberg.

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

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

[9] 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.

[10]Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. Journal of the American society for information science and technology, 61(12), 2544-2558.

[11]Tofighy, S., & Fakhrahmad, S. M. (2018). A proposed scheme for sentiment analysis. Kybernetes.

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

[13]Türkmenoglu, C., & Tantug, A. C. (2014, June). Sentiment analysis in Turkish media. In International Conference on Machine Learning (ICML).

[14]Kaya, M., Fidan, G., & Toroslu, I. H. (2012, December). Sentiment analysis of turkish political news. In 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (Vol. 1, pp. 174-180). IEEE.

[15]Kaya, M. (2013). Sentiment analysis of Turkish political columns with transfer learning. Diss, Middle East Technical University.

[16]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 (pp. 1-8).

[17]Eryiğit, G., & Adalı, E. (2004, February). An affix stripping morphological analyzer for Turkish. In Proceedings of the IASTED international conference artificial intelligence and applications (pp. 299-304).

[18]Yıldırım, E., Çetin, F. S., Eryiğit, G., & Temel, T. (2015). The impact of NLP on Turkish sentiment analysis. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 7(1), 43-51.

[19]Rong, X. (2014). word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.

[20]Wang, H. (2014). Introduction to Word2vec and its application to find predominant word senses. URL: http://compling. hss. ntu. edu. sg/courses/hg7017/pdf/word2vec% 20and% 20its% 20appli cation% 20to% 20wsd. pdf.

[21]Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[22]Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.

[23]Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. Expert Systems with Applications, 57, 232-247.

[24]Freeman, E. A., Moisen, G. G., Coulston, J. W., & Wilson, B. T. (2016). Random forests and stochastic gradient boosting for predicting tree canopy cover: comparing tuning processes and model performance. Canadian Journal of Forest Research, 46(3), 323-339.

[25]Kuncheva, L. I., & Rodríguez, J. J. (2007, May). An experimental study on rotation forest ensembles. In International workshop on multiple classifier systems (pp. 459-468). Springer, Berlin, Heidelberg