Int. J. Adv. Eng. Pure Sci. 2023, 35(1): 125-141
DOI: 10.7240/jeps. 1249586

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Improving the Performance of Sentiment Analysis by Ensemble Hybrid Learning Algorithm with Nlp and Cascaded Feature Extraction

Ayşe Berna ALTINEL GİRGİN [1], Sema ŞAHİN [2]

[1,2] Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye

Abstract
Sentiment analysis is a challenging problem in Natural Language Processing since every language has its own character within several difficulties such as ambiguity, synonymy, negative suffixes…etc. Since words with ambiguity can have different sentiment scores depending on the meaning they have in their corresponding context, we accomplished a study on Turkish language to determine whether the polarity scores of these polysemous words may differ according to their meaning. For a word with ambiguity, we first made a polarity calculation module to calculate its polarity score. This way, we calculated the polarity scores of 100 Turkish polysemous words. Then, since negation directly affects the correct meaning of the word in the sentiment analysis, a negation handler module is also implemented. After that, we prepared a sentiment polarity corpus which consists of 159,876 Turkish words including 100 Turkish polysemous words. Actually, the main purpose of this study is to detect sentiment polarity of Turkish texts by considering and building a specialized module for polysemous words. In short, we built a system for Turkish sentiment polarity detection task including these modules: Pre-processing, Polarity Calculation Module, Negation Handling Module, Feature Generation Module, and Classification Module. According to our knowledge, this is the first study which includes all of these modules in one Turkish sentiment analysis task. Finally, we conducted this corpus using an ensemble hybrid regularized learning algorithm on two self-collected Twitter-datasets. Experimental results show that the suggested approach improves the classification performance on Turkish sentiment analysis task.
Keywords: Sentiment analysis, word ambiguity, machine learning, hybrid learning algorithm, LSTM

I. INTRODUCTION

A natural language may contain semantic confusion in words due to its nature. Polysemous words can have different meanings depending on their intentional usage in the context. The purpose of the Word Sense Disambiguation (WSD) [1,2] problem is to determine the sense of a word with ambiguity in a sentence. The problem of ambiguity, which people solve with the help of their cognitive processes while communicating, is one of the important and current issues discussed in the field of NLP so that machines can solve it with algorithms.

While the studies on WSD problem were mostly done in English, we conducted this study for Turkish language, which is harder to study due to the complexity of Turkish morphology, syntactic structure, and being an agglutinative language. In the literature, there are some studies on WSD for Turkish language, for example: Açıkgöz et al. [3] studied on semantic ambiguity for Turkish language and measured the performance of the given features for different classification algorithms. Orhan and Altan [4], conducted experiments to find effective features to eliminate the ambiguity of meaning in Turkish verbs and summarized the results.

Sentiment analysis (SA) goals to determine the sentiment polarity of a word as positive or negative with the help of some outer resources such as corpus, dictionary and by using algorithms. It is known that words with multiple meanings, for instance, polysemous words, may have different sentiments (i.e., positive, negative) depending on the meaning they have. According to the studies in the literature, it is seen that polysemous words are often not taken into account or the average of the sentiment values of all meanings of ambiguous words is calculated as the sentiment value of that word [5].

Negation is an essential concept in NLP. In SA, finding the correct meaning of the word is crucial and since negation directly affects the sentiment, it should be detected prior to the analysis. In Turkish, negation usually appears in two forms. It can appear either in a word form or can appear in a suffix form.

Corresponding Author: Ayşe Berna ALTINEL GİRGİN, Tel: 0216 777 3813, e-posta: berna.altinel@marmara.edu.tr

Submitted: 09.02.2023, **Accepted:** 28.02.2023

The prepossessing phase of the NLP process includes a stemming step. When stemming is applied to the word that includes negation, the meaning of the word is lost. To prevent these types of situations, negation must be handled before the stemming is applied to preserve the negated meaning of the word that includes negation. Although the studies for Turkish are few, it is seen that in most of the studies, the words are stemmed directly from the root and the sentiment polarity values of these roots are found. It is observed that there are very few studies which handle negation in the sentiment polarity classification task [5-6].

By considering that there is a lack of studies in literature, the purpose of this study is to make a preliminary study on whether the polarity values of Turkish words with ambiguity differ according to their meanings and to prepare a negation handler module, so that we have taken into account both forms of negation of Turkish language in the polarity classification.

Actually, the main purpose of this study is sentiment analysis of Turkish texts by considering and building a specialized module for polysemous words. Since words with ambiguity can have different sentiment scores depending on the meaning they have in their corresponding context, we accomplished a sub-module in our sentiment analysis system on Turkish language to determine whether the polarity scores of these polysemous words may differ according to their meaning. We think that this is very important since the polarity scores of polysemous words could change from their con-text to context meanwhile they can have different meanings in different contexts; which directly have the capability of changing the polarity score of the text.

Our contributions in this work are as follows:

- One of the most important purposes of this study is to show that polysemous words, which have the capability of having different meanings based on their context, could have different sentiment polarity scores according to their context. As far as we know, this is the first effort for Turkish language. By making statistical calculations and morphological analysis, we calculated different scores for the corresponding different meanings of about 100 Turkish polysemous words. This study contributes to the literature due to its unique property to be the first in kind.
- We prepared Turkish sentiment polarity corpus based on some statistical calculations instead of using translated dictionaries. This extended dictionary consists of 159,876 words including 100 polysemous words and publicly available for other researchers upon their request. As far as we know, this study again is the first attempt on building such a dictionary for Turkish which has sentiment scores of polysemous words. As we see from the literature, there is no such dictionary including different scores for ambiguous words.
- We extracted features $\{F_1\text{-}F_{10}\}$ extracted from the extended GDELT based dictionary, which clearly seems to have an improvement on the classification task in comparing to using features extracted from conventional TF. As far as we know, there is very little effort for such a comprehensive feature extraction module for Turkish sentiment analysis task.
- We built a system for Turkish sentiment polarity detection task including these modules: 1) Pre-processing, 2) Polarity Calculation Module, 3) Negation Handling Module, 4) Feature Generation Module, and 5) Classification Module. According to our knowledge, this is the first study which includes all of these modules in one Turkish sentiment analysis task.
- The suggested algorithm in this study also includes an ensemble-based architecture, which is very valuable since there is very little effort on using ensemble-based learning algorithms for Turkish SA problem.
- By conducting several experiments in our experimental environment, we tried to show the effectiveness of the proposed algorithm over the state-of -the-arts algorithms on the sentiment polarity detection task. By using the proposed algorithm, we observed a very promising performance on Turkish sentiment analysis, which is very appreciated since there are rare significant results on Turkish sentiment analysis.
- All materials in this study (i.e., the extended dictionary, datasets and implementation) are publicly available for other researchers upon their request. This is a great advance for Turkish, since there is very little publicly available source for Turkish language.

The remainder of the paper is organized as follows: Section 2 presents related work on WSD and sentiment analysis for Turkish language. The proposed methodology, including polarity calculation, feature generation module, and negation handler module are explained in Section 3. The experimental setup, dataset, polarity calculation analysis results for polysemous words, and the corresponding experimental results of the sentiment polarity detection task are presented in Section 4. Finally, Section 5 gives the conclusion and future directions.

II. RELATED WORK

2.1 Related Work About Word Sense Disambiguation

Açıkgöz et al. [3] studied on semantic ambiguity for Turkish language and aimed to measure the effect of the performance of the given features for different classification algorithms. A randomly obtained dataset from Penn Treebag Corpus was translated into Turkish, and there were 1400 sentences in the translated dataset. C4.5, k-nearest neighbors (KNN), Random Forests(RF), Naive Bayes(NB), multilayer Perceptron, Rocchio, and Linear classification algorithms were used. In addition, SkipGram and Continuous Bag of Words (CBOW) models were used. According to their experimental results, there were some important observations: 1) SkipGram generally outperformed CBOW. 2) Increasing the window size does not increase the model performance. 3), increasing or decreasing the vector size, or increasing the corpus size did not have a remarkable effect on the success of the model.

Orhan and Altan [4], implemented some algorithms and conducted experiments to find effective features to eliminate the ambiguity of meaning in Turkish verbs. METU-Sabancı Turkish Tree Bank was used in this study. The verb "gelmek (English translation: come)" was used in the experiments and its meanings were examined gradually. By using WEKA tool, AODE, IBk, and J48 algorithms were run on the dataset. Experimental results reported in this study show that there was only a slight difference between the classification performance of the used algorithms. Moreover, it has been deduced that feature selection has a significant effect on the experiment results.

Çetiner, Yıldırım, Onay and Öksüz [7] studied on resolving the ambiguities of multiple-sense words that have the same morphological structure. Turkish KeNet word network was used to capture the meaning of the words. Segmentation of sentences was provided by Turkish BERT segmentation model. First of all, word representation vectors of BERT were created for each term in the corpus. After that, the vectors obtained from KeNet are selected according to their cosine similarity scores, and then the related term is associated with the relevant meaning in KeNet and indexed. The study was conducted on a corpus of 130 thousand news, the results of 4 queries were given as an example. The queries were performed without using the WSD module and according to the experimental results, there was no difference observed in direct indexing. A noticeable increase was detected in the queries made using the WSD module.

Mert and Dalkılıç [8] adapted Lesk and Simplified Lesk algorithms to Turkish in order to solve Turkish WSD problem. While Lesk method compares the definitions of the indefinite word and the definitions of other words in the sentence in order to remove ambiguity of the words; on the other hand, Simplified Lesk method compares the stems or roots of the words in the sentence with the definitions of the indefinite word, instead of comparing the definitions of all words. These methods were tested with 10 sample sentences and the average of those 10 runs were reported as the success rate of the model. As a result, it was seen that Lesk-like methods produced better classification results in comparing to the simplified lesk-like methods.

Aslan, Arıcan, Bayrak, Özbek, and Yıldız [9] aimed to prepare the most accurate corpus of manual tagging for tourism domain by using a large amount of Turkish data. The data set was created by selecting 14,000 comments had written by the customers on the internet, and a total of 20,000 sentences were processed with an unsigned data set. While marking the words and word groups, they were first processed according to the tourism dictionary, and the words that did not have a meaning were marked with their meanings in the general Turkish word network (KeNet). Experimental results in this study show that; 93,653 words out of 20,000 sentences were marked semantically, 1737 of the meanings came from the general Turkish word network, while the remaining 111 meanings came from the tourism word network. The authors stated this study to be a resource for future studies.

Tüysüz and Güvenoğlu [10] applied machine learning algorithms to the dataset of Semeval-2007 workshop for Turkish, and they compared their experimental results with the experimental results shared in the workshop. Within the scope of the research, test and evaluation data consisting of six words including the types classified as ten in the noun type, ten in the verb type, and other types were prepared. The prepared dataset has been tested with Semeval, NB Algorithm and Decision Tree Algorithm methods by using WEKA Tool. Due to the small number of samples in Semeval-2007, it resulted in low performance. In machine learning algorithms, more successful results were obtained compared to Semeval-2007 workshop.

Arslan, Orhan and Tahiroğlu [11] proposed a solution for Turkish WSD task. The semantic graph of Turkish lemmas was created by using the co-occurrence relation at the sentence level. First, lemmas were added to the graph database. Then, the lemmatization process was applied to the sentences and connections were established between the lemmas. By ignoring inflected words with more than one lemma alternative, a graph without ambiguity is obtained. All lemmas collected in this graph are linked by the 'COOCCUR' relationship. NB classifier was used for pattern classification. For each lemma sequence, NB calculations were made using lemma relationship statistics. When the test results were evaluated in a controlled manner, the success rate was 68.42%, and it has been observed that this success rate increases when trained with more data sets.

Aydın and Kılıçaslan [12] improved a corpus based WSD application by using Inductive Logic Programming (ILP). WSD application was implemented with a ILP system, ALEPH (A Learning Engine for Proposing Hypotheses). Three data files were required to build the theories in ALEPH. In practice, the background information of each word's training set sentences was given to the ALEPH system by taking positive examples and negative examples, and a model was created. In the evaluation phase of this model, files containing the background information of the test set, positive samples, and negative samples were created. Finally, these files were given to the system and the test process was carried out. They concluded that most WSD techniques fail to detect the relationship between information from different lexical sources for clarification. However, ILP was successful in displaying relational information and can create a different structure with data from various sources.

Selamet and Eryiğit [13] proposed a semisupervised context based WSD approach for data augmentation for low-resource languages (LRLs). The proposed model was tested on the English dataset in order to demonstrate the accuracy of this study. The suggested semi-supervised method used context embedding and seed set. The study was tested in 9 different context-based language models (BERT, ELMo, RoBERTa, etc.) and their effects were examined. As a result, a performance increase of 28% has been achieved, thus it was conducted that the initial findings are promising. In addition to the original study, the maximum and average similarity models of the seed set were expanded according to a certain threshold value and tested in the highest performing language model.

2.2 Related Work About Sentiment Analysis

A recent study in [14] aims to guide both public and private enterprises. AutoTrain technology and bert model were used in this study. Thanks to AutoTrain, analysis can be made without knowing any statistics or mathematics. The Bert model, unlike other models, evaluates the input from both the right and left. Thus, the margin of error of the output is minimized. According to the experimental results, the methodology presented in this study achieved 90% success.

Çılgın et al. [15] attempted to learn the public's emotional perspective on the vaccines developed during the Covid-19 epidemic. In this study, a majority learning architecture was developed using Support Vector Machines, Naive Bayes, K-nearest neighbor, Logistic regression, Random Forest and XGBoost algorithms. In order to form a dataset in this study, Turkish tweets with Covid-19 and vaccine tags between April and August 2021 were collected. According to the experimental results of the study, the percentage of people who are not antivaccine was very low.

Another study [16] aims to perform sentiment analysis on Turkish language datasets gathered from Twitter. In this study, first of all, tweets were collected using the Twitter API. After that, two different libraries were used for morphological analysis. These libraries are Zemberek Library and Snowball Library. TF-IDF technique was used as the text representation for both the texts preprocessed after Zemberek library and snowball Library. SVM, Logistic Regression, Random Forest, NB, LSTM, and SGD algorithms were applied on both datasets for classification. The analysis reported in this study states; these machine learning algorithms on these two datasets give up to 87% classification performance which is very significant and promising for Turkish sentiment analysis task.

Kirelli and Arslankaya [17] worked on sentiment analysis of tweets about global warming. Data was collected from Twitter with hashtags such as #kureselısınma, #iklimdegisikligi, and #iklimetkisi using the Twitter API. In the preprocessing step; numbers, special characters, punctuation marks, and stop words have been filtered. Then, the stemming process was applied on this cleared dataset with Zemberek library [18]. After that, the data set was classified with K-NN, SVM, and NB(Bayes) algorithms. According to their reported results, using N-gram with K-NN increased the classification success rate.

The purpose of another very recent study [19] is to make a sentiment analysis of the criticisms in the social media towards teachers and health workers during Covid-19 epidemic, especially from the perspective of children and parents. Turkish tweets containing the hashtags of health workers and teachers were collected by the Twitter API for the period of 11 March 2020 and 2 July 2021. In this dataset, for the “healthcare professionals” hashtag there are tweets from 10324 different users and totally 15368 tweets, and for the "teachers" hashtag there are tweets from 1685 different users and totally 2956 tweets. After pre-processing of the data, dictionary-based sentiment analysis was performed at the sentence level. As a result of the analysis, 70% of the tweets shared by Twitter users to their teachers were detected as sentiment positive, and 61% of the tweets shared on their health lives were detected as sentiment positive. Moreover, a very similar recent study was accomplished by Kandıran et al. [20]. The authors attempt to understand what the society thinks about online education during the Covid-19 epidemic by doing sentiment analysis on their social media posts. The data set contains 8545 Turkish tweets. In order to get tweets from Twitter, 28 different hashtags related to online education have been used.

A new SA system implemented by Yüksel and Tan in [21], a sentiment model was proposed to analyze and classify restaurant reviews as positive, negative or neutral. The dataset used in this study was collected from the Foursquare application. From Foursquare a total of 7086 Turkish reviews from 128 different restaurants were extracted to form the dataset which was later used both in training and testing of the proposed system. As the first process, the typos of all the data in the dataset were corrected along with rooting all the words in the data to access the base form of words. For these preprocessing steps, ITU Turkish NLP Web Service and Zemberek tool were used. Then to perform classification on the reviews, first, the sentiment values were determined. As the second step, the same reviews were translated from Turkish to English using the Google Translate API. Then the sentiment values of these translated reviews were found as well using the Text Analysis API. The reason for translating all the reviews to English and finding the sentiment values of these reviews is to compare the classification results of the same reviews both in Turkish and English. For classification, three different techniques were used which are Naive Bayes (NB), the Social Information Discovery Algorithm (SIDA) proposed in this study, and Text Analysis API. In the classification approach of SIDA algorithm, the positive reviews were tagged as +1 while the negative reviews were tagged as -1 and the neutral reviews were tagged as 0, these neutral reviews were determined due to not including any words related to the concepts of restaurants, foods, etc. The classification was performed using all these three classifiers. Then the proposed approach, NB and Text Analysis API was evaluated for both Turkish and English reviews. According to the results, the SIDA algorithm performed the best in English reviews with an 84,49% of accuracy rate. Following that, again the SIDA algorithm achieved 81,97% accuracy rate in Turkish reviews. The NB algorithm achieved 78% accuracy rate in English reviews and 73% accuracy rate in Turkish reviews. The Text Analysis API can only be evaluated for English reviews because it only works in English input. And according to that evaluation, the Text analysis API received a 59.38% accuracy rate.

Köksal and Özgür in [22] offered a transformer-based architecture that uses the by hand labelled BounTi dataset which contains Turkish tweets about particular universities in Turkey. This dataset has characteristic features of social media texts such as emojis, slang, and typo. Multilingual and Turkish transformer models such as MBERT, XLM-Roberta, and BERTurk are evaluated. According to experimental results, the proposed model reaches a macro-averaged recall score of 72,9%.

In another recent study proposed by Güran et al. in [23], sentiment analysis was performed on the social media data. Mostly the effects of optimizing the Support Vector Machine on sentiment analysis were analyzed. It has been seen that SVM techniques give the most successful results when compared with other machine learning methods in the literature such as Naive Bayes (NB), and Random Forest (RF). The selection of the appropriate kernel function and the determination of the appropriate parameters of the kernel function are of great importance in SVM success rates. The grid search method is used to find the most suitable states of the parameters that affect the mentioned performance rate. In a basic sense, unlike other machine learning methods, SVM separates 2 classes by drawing 2 optimum lines and creates a maximum-sized corridor space in the middle of the separation, instead of fitting and separating a linear line in the classification graph. For the dataset, three different data sets named VS1(3 classes, 3000 data), VS2(4 classes, 157 data), VS3(3 classes, 105 data) were used. The proposed model was tested on these sets based on the number of state separations in the datasets. The effect of revealing the most appropriate parameters for modeling on the results is also shown numerically, by observing a large difference between the highest and lowest performance in the VS1, VS2, and VS3 datasets tested. As a result, the proposed model received an average of 75.2% accuracy rate. In future studies, the authors plan to determine parameters using heuristic methods.

III. METHODOLOGY

The main purpose of our study is to attempt to develop a dictionary-based ensemble learning design for Turkish sentiment analysis task by handling both WSD and negation. The general architecture of the proposed system is given in Figure 1. Our study has four main modules: 1) Pre-processing, 2) Polarity Calculation Module, 3) Negation Handling Module, 4) Feature Generation Module and 5) Classification Module. The details of these modules are given in the below sections.

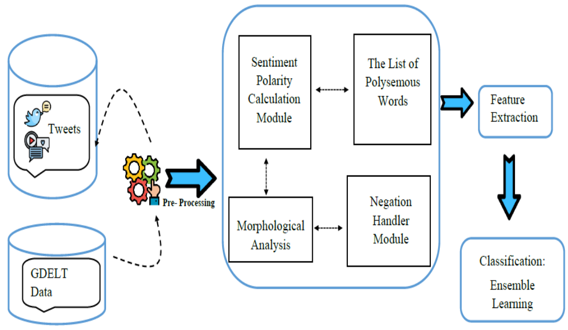
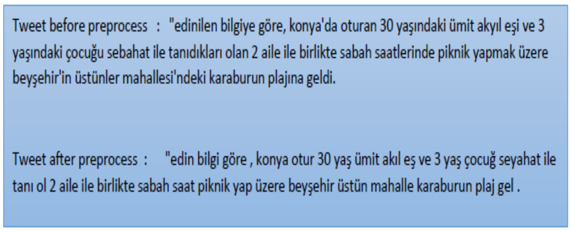


Figure 1. The general architecture of the proposed system

3.1 Pre-processing module

The first step in our system is the preprocessing phase. This step comprises of a cleaning phase which filters unwanted links, symbols, hashtags, emoji, and numbers from the tweets. Moreover, the roots of the words are found in the Zemberek library [18] as shown in Figure 2.



Tweet before preprocess : "edinilen bilgiye göre, konya'da oturan 30 yaşındaki ümit akyıl eşi ve 3 yaşındaki çocuğu sebahat ile tanıdıkları olan 2 aile ile birlikte sabah saatlerinde piknik yapmak üzere beyşehir'in üstünler mahallesi'ndeki karaburun plajına geldi.

Tweet after preprocess : "edin bilgi göre , konya otur 30 yaş ümit akıl eş ve 3 yaş çocuğ seyahat ile tanı ol 2 aile ile birlikte sabah saat piknik yap üzere beyşehir üstün mahalle karaburun plaj gel .

Figure 2. Example text in the preprocessing module using the Zemberek library [18].

3.2 Polarity calculation module

In order to calculate the sentiment polarity values of words, the methodology presented in [24] was taken into consideration. This study uses GDELT[1] (Global Data on Events, Languages, and Tone) Project to get Turkish documents. GDELT actually is a very large, on-going, and comprehensive open database of the real world. It captures and analyses the world's news media in many formats in over 100 languages including Turkish. The sentiment polarity calculation has been done by Equation 1 in [24]:

$$S_w = x = \frac{\Sigma_{x=1}^{n}(f_i * d_i)}{\Sigma_{x=1}^{n} f_i} \quad (1)$$

where n shows the total number of documents the word is in, d represents polarity score of the document, f_i is the number of occurrences of the word in the document and S_W represents the calculated polarity score of the word.

Since the news in GDELT Project they have been annotated as positive or negative for their sentiment polarities, Equation 1 takes advantage of these annotated documents in order to discover the sentiment polarity scores of the words occurring in these documents. The logic behind Equation 1 is that positive sentiment polarity words commonly occur in positive sentiment polarity documents, while negative sentiment polarity words commonly occur in negative sentiment polarity documents [24].

In our previous study in [25], in order to achieve sentiment polarity values of Turkish words, we were inspired by the methodology in [24]. We prepared our sentiment polarity dictionary which contains about 84,744 Turkish words [25]. There are two stages during the building of this dictionary. 1)In the first stage, we downloaded Turkish documents from GDELT Project using the BeautifulSoup library implemented in Python. Then a preprocessing step has been performed as mentioned in Pre-processing module. 2) In the second stage, the sentiment polarity values of words occurring in these documents have been calculated according to Equation 1.

In this study, we extended our existing dictionary in [25] by downloading more documents from GDELT Project. Finally, the dictionary has about 120,000 Turkish words. A sample output of the polarity calculation module is shown in Table 1.

Table 1. Sample output of polarity calculation module

Word	Polarity (Tone) Value	Sentiment Polarity
tanı (diagnosis)	0.051187981	positive
kılıç (sword)	-0.391961507	negative
ölü (dead)	-0.104723823	negative
tatil (holiday)	0.677150085	positive
risk (risk)	-0.92443375	negative
misafir (guest)	0.115154259	positive
pahalı (expensive)	-2.307692308	negative
büyü (magic)	-0.993990332	negative
güçlü (strong)	0.538346933	positive
oyun (play)	0.333479786	positive

1 https://www.gdeltproject.org

3.3 Negation handler module

In Turkish, negation usually appears in two forms. It can appear either in a word form or can appear in a suffix form. The words that cause negation are words like "güzel" (nice), "değil" (not), "hiç" (any). The suffixes that cause negation are the ones that cause negative meaning to the adjectives and verbs such as "-me", "-ma", "-sız", "-siz", "suz" and "-süz".

To give an example for the first form of negation, which are the word form negations in the sentence "Burası güzel değil" (this place is not nice), the word "güzel" (nice) alone contains a positive sentiment but when the "değil"(not) word is used, the meaning of the sentence is changed. Using the "değil" (not) word combined with the word "güzel" (nice) together causes negation in the given sentence. To generalize, words like "yok" (none), "değil" (not), "hiç" (any) came after the words. It negates the previous word in the sentence and together it causes negation. To give an example for the second form of negation which are suffix form negations, for instance, the verb "yapmamak" (not to do) has negative sentiment and includes negation due to including the suffix "-ma". The stem of the word "yapmamak" (not to do) is "yap" (do) has a neutral sentiment, but when the "-ma" suffix is added to the word the word becomes negated. To generalize, when the suffixes "-me", "-ma", "-sız", "-siz", "suz" and "-süz" are added to the verbs and adjectives, they become negated.

Negation is an important concept in NLP. In sentiment analysis, finding the correct meaning of the polysemous word is very crucial and since negation directly affects the sentiment and it should be detected prior to the analysis. Generally, in most NLP tasks, the preprocessing is an important step. Usually preprocessing includes steps like stemming and stop, word filtering, etc. In the second form of negation, when stemming is applied to the word that includes negation, the meaning of the word is lost. To prevent these types of situations, negation must be handled before the stemming is applied to preserve the negated meaning of the polysemous word that include negation.

A negation module in our study is implemented which handles both forms of negation. For the negation module, a rule-based approach was used. To handle negation, first the morphological analysis of sentences is found using a context based morphological analyzer tool. The output of a morphological analysis of a sample sentence using the ITU NLP tool and sentiment polarity scores of the words are given below in Table 2.

Table 2. Sample Output from Nlp Morphological Analyzer Tool and Sentiment Polarity Scores of the Words

Sentence: "Sinemaya gitmemiz ne kadar güzel oldu" (How nice it was that we went to the cinema)

Word	Polarity (Tone) Value	Sentiment Polarity
"sinemaya"	0.051187981	positive
(to the cinema)	-0.391961507	negative
"gitmemiz" (we go)	-0.104723823	negative
"ne" (how)	0.677150085	positive
"kadar" (much)	-0.92443375	negative
"güzel" (nice)	0.115154259	positive
"oldu"(it was)	-2.307692308	negative

From the morphological analysis, the stems and tags of the words can be found separately. In this study, the negation handler module deals with both forms of negation by using rule-based approaches. Different rules were applied to solve each form of negation since the handling process for each form of negation differs Word-based negation, which is the first form of negation, is caused by words such as "yok" (none), "değil" (not), and "hiç" (any) in Turkish. In the negation model, a negation list containing these words is created, which is called negation_list1, and this list is used to detect the word-based negations. To detect all negations in a given sentence, the first and the roots of each word are checked from the negation_list1. If the word in the sentence is in negation_list1, the module finds the word that comes before the negation word and combines it with the negation word. After the combination process, the combined word is tagged with the "neg_" keyword. To give an example; in the "Orası güzel değil" (it's not nice) sentence, the word "değil" (not) together with the "güzel" (nice) word causes negation. In the negation handler module, the root of each word in the given sentence is checked from the negation_list1, and since the "değil" (not) word is in the negation_list1, the word comes before the "değil" (not) word which is "güzel" (nice) word is combined with the "değil" (not) word and tagged with the "neg_" keyword to handle the negation. After this process, the sentence became "Ora neg_güzel-değil değil". The structure of the sentence was not corrupted; only the negated word is tagged with this process.

To handle the second form of negation which is caused by suffixes, different rules are applied for different word tags. When "-me", "-ma", "-sız", "-siz", "suz" and "-süz" suffixes are used in verbs, nouns and adjectives they cause negation. For example, for "gel" (come) word which is a verb, when “-me” suffix is used on this verb, the verb becomes “gelme” (not come). and the verbs become negated. In the negation handler module, a negation list is created, which is called “negation_list2”. This list includes the suffixes that cause negation in verbs, nouns, adjectives, which this list will be used for detecting these forms of negation. In the negation module, when the second form of negation is detected, the word that contains the negation is tagged with “neg_” keyword to represent the negation. To detect the second form of negation, the suffixes given above are checked inside the words, if any word includes these suffixes, the word will be tagged as negation. In the negation module, different rules are applied to detect negation for different word types such as verbs, adjectives, and nouns. First, the morphological analysis of each word is derived by applying the morphological analysis to the given sentence. From the morphological analysis, the tag of words is detected. The negation module is executed for each word in the sentence given to the negation module.

The module checks the type of the word. According to the type of the word, the negations are handled. If the type of the word is detected as a noun, the suffixes of the word are checked. If the word contains any suffix specified in negation_list2, the noun is tagged as negation. To give an example, the “Sevinçsiz kaldım” (I'm without joy) sentence is considered as shown in Table 3:

Table 3. Morphological Analysis of the Sample Sentence for Noun Type Negations

Sentence: “Sevinçsiz kaldım” (I'm without joy)	
Word	Morphological Analysis
“sevinçsiz” (without joy)	sinema+Noun+A3sg+Pnon+Dat
kaldım” (I‘m)	git+Verb+Pos^DB+Noun+Inf2+A3sg+P1pl+ Nom

As it can be seen from Table 3, the word “Sevinçsiz” (without joy) is a noun and it contains the suffix “-siz” which is one of the suffixes that causes negation found in the negation_list2. As a result, the word “Sevinçsiz” (without joy) is detected as a negation word and tagged using the “neg_” keyword. The output is “neg_sevinç kal”. As it can be seen from the example, besides detecting the negation, the roots of the words in the sentence are also found in this process, that is, why in the output the stems of the words appear instead of the word itself.

A similar rule is applied for adjective word types to detect negation. According to the morphological analysis of the word, if the tag of the word is an adjective, the suffixes that cause negation found in the negation_list2 are searched within the word. If the word contains any of the suffixes from the negation_list2, it is tagged with the “neg_” keyword. For example, when the “İnançsız olmak” (to be unbeliever) sentence is given to the negation module, from the morphological analysis of each word in the sentence the tag of words are checked.

The morphological analysis of the word “inançsız” (unbeliever) is shown in Table 4. According to the morphological analysis of the words, the word “inançsız” (unbeliever) has an adjective tag and the word also includes the “-sız” suffix which is one of the suffixes that cause negation in adjective type words. In the negation module, the word “inançsız” (unbeliever) is tagged as a negation. As an output, when the negations are handled, the sentence becomes “neg_inanç or”.

Table 4. Morphological Analysis of the Sample Sentence for Adjective Type Negations

Sentence: “İnançsız olmak” (to be unbeliever)	
Word	Morphological Analysis
“inançsız”	inançsız+Adj
(unbeliever)	ol+Verb+Pos^DB+Noun+Inf1+A3sg+Pnon+Nom

The last type of word negation is verb type negation; a different approach was used to detect verb type negation. To detect the negation of the words in verb form, the properties of the ITU NLP morphological analyzer were used. With the morphological analyzer, the negativity of words can be detected. For example, when the “Gelmemek senin tercihin” (It's your choice not to come) sentence is morphologically analyzed, according to Table 5, the word “gelmemek” (not to come) has the verb as a tag. And if the verb is negative, it includes the “Neg^” tag.

Table 5. Morphological Analysis of the Sample Sentence For Verb Type Negations

Sentence: “Gelmemek senin tercihin” (It's your choice not to come)	
Word	Morphological Analysis
“gelmemek” (not to come)	gel+Verb+Neg^DB+Noun+Inf1+A3sg+Pnon+ Nom
“senin” (your)	sen+Pron+Pers+A2sg+Pnon+Gen
“tercihin” (your choice)	tercih+Noun+A3sg+Pnon+Gen

Therefore, verb-type words can be classified as negation if they contain the tag "Neg^" in their morphological analysis. So to handle verb type negation. The "Neg^" tag is searched if the word has the verb tag. If the verb includes the specified tag, it is tagged as a negation. For example, we can consider the sample word given in Table 5. According to the sample sentence, the word "gelmemek" (not to come) has the tag and includes the "Neg^" tag, therefore it will be tagged as negation using the "neg_" keyword. As an output after the negation handling process, the output when the sample sentence in Table 5 is given, which is "neg_gel sen tercih".

To conclude, the negation handler module detects two forms of negation. After the negations are detected, each word is also stemmed. Different rule-based approaches were used to detect different forms of negation. The negation module takes sentences as input and according to the morphological analysis of sentences, each word in the sentence is checked and tagged if the word has negation, it is tagged accordingly. Each word is checked if it includes any form of negation. If the word does not have the first form of negation, the word is checked if it includes the second form of negation. In the second form of negation, there are different negation types for different types of words such as verbs, nouns, adjectives. Each word is checked if it includes any type of the second form of negation. To understand how the negation module performs, 10 sample sentences were given to the negation module as an input. The results can be seen from Table 6.

Table 6. Testing of the Negation Module with 10 Sample Sentences

Input Sentence	Output
"Bu insanların akılsız olması hiç güzel değil" (It's not nice that these people are mindless)	bu insan neg_akılsız neg_ol-hiç hiç neg_güzel-değil değil
"Geliyor değilim" (I'm not coming)	neg_gel-değil değil
"Değilim seninle iyi" (I'm not fine with you)	neg_değil sen iyi
"Çalışmaman bizi çok üzüyor" (It makes us very sad that you don't work)	neg_çalış biz çok üz
"Gelmemek ya da gitmemek senin tercihin" (It's your choice to not to come or not to go)	neg_gel ya da neg_git sen tercih
"Bu işleri yapmaman senin suçun" (It's your fault for not doing these works)	bu iş neg_yap sen suç
"Beni sevmiyorsun"(You do not love me)	ben neg_sev
"Böyle inançsız olarak nasıl yaşıyorsun" (How do you live without faith like this)	böyle neg_inanç olarak nasıl yaşa

3.4 Feature Generation Module

We prepare our tweet vectors by using the sentiment polarity scores of the words in the tweets. In other words, the feature vectors in our classification system comprises of sentiment polarity scores of the words gathered from our extended sentiment polarity score dictionary, which includes about 120 words.

We represent tweets in our test dataset using 10 features. These features are given in Equations 2-12:

F_1 shows the number of words which have positive sentiment polarity in a given text as specified in Equation 2:

$$F_1: \sum_{i=1}^{n} PositiveSentiment\_word_i \quad \textbf{(2)}$$

F_2 represents the number of words which have negative sentiment polarity in a given tweet as specified in Equation 3:

$$F_2: \sum_{i=1}^{n} NegativeSentiment\_word_i \quad \textbf{(3)}$$

F_3 shows the number of total occurrences of positive sentiment polarity words in the text as specified in Equation 4:

$$F_3: \sum_{i=1}^{n} TF * PositiveSentiment\_word_i \quad \textbf{(4)}$$

F_4 demonstrates the number of total occurrences of negative sentiment polarity words in the text as specified in Equation 5:

$$F_4: \sum_{i=1}^{n} TF * NegativeSentiment_{word_i} \quad \textbf{(5)}$$

F_5 indicates the average positive sentiment polarity score in a given text as specified in Equation 6:

$$F_5: \frac{1}{n} \sum_{i=1}^{n} PositiveSentiment_{score_i} \quad \textbf{(6)}$$

F_6 represents the average negative sentiment polarity score in a given text as specified in Equation 7:

$$F_6: \frac{1}{n} \sum_{i=1}^{n} PositiveSentiment\_score_i \quad \textbf{(7)}$$

F_7 demonstrates the number of positive seed words in a given text as specified in Equation 8:

$$F_7 : \sum_{i=1}^{n} PositiveSeed\_word_i \quad \textbf{(8)}$$

F_8 shows the number of negative seed words in a given text as specified in Equation 9:

$$F_8 : \sum_{i=1}^{n} NegativeSeed\_word_i \quad \textbf{(9)}$$

F_9 indicates the number of total positive seed word occurrences in a given text as specified in Equation 10:

$$F_9 : \sum_{i=1}^{n} TF * PositiveSeed\_word_i \quad \textbf{(10)}$$

F_{10} demonstrates the number of total negative seed word occurrences in a given text as specified in Equation 11:

$$F_{10} : \sum_{i=1}^{n} TF * NegativeSeed\_word_i \quad \textbf{(11)}$$

In our text representation methodology, we attempt to represent text instances using Equation 12.

text$_i$: { $F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}$ }
(12)

where $text_i$ is a tweet instance in our test dataset and $F_{1\text{-}10}$ are the features explained above with Equations 2-11.

The list of seed words and their corresponding seed scores are represented in Table 7.

Table 7. The list of seed words

Seed Word	English Translation	Seed Score
Acayip	Strange	2
Az	Little	-1
Azıcık	Few	-1
aşırı	Extreme	2
Bayağı	Pretty	1
Büyük	Large	1
Cidden	Really	1
Devamlı	Continuous	1
En	Most	2
Fazla	More	2
Gayet	Plenty	1
gerçekten	Truly	1
Hakkaten	Really	1
Hep	Always	1
Kesinlikle	Definitely	1
Kuşkusuz	No doubt	1
Muhteşem	Wonderful	2
Nadiren	Rarely	-1
Sürekli	Continually	1
Tam	Full	1
Tamamen	Completely	1
Yeterince	Enough	1
Çok	A lot	2
özellikle	Especially	1

3.5 Classification module

In this module; textual datasets which include polysemous words, are classified according to their sentiment polarity. The purpose of this classification step is to decide the sentiment polarity of the test samples, whether they are positive or negative. This classification task is performed on two self-collected datasets. The information about these datasets is given in Section 4.2.

The classification module in this study includes an ensemble form of the SVM, NB, RF, and LSTM classifiers. First of all, we implemented SVM, NB, RF, and LSTM classifiers in Python using Python Data Analysis Library (pandas v1.3.5), Machine Learning Libraries (scikit learn 1.0.2, sklearn.ensemble and Pytorch 1.13.0).

Then we implemented a majority voting scheme in the classification phase to determine the output of the ensemble algorithm, which combines the decisions of SVM, NB, RF, and LSTM classifiers. A brief description of classifiers is given below:

<u>Support Vector Machine (SVM):</u>
SVM is a very popular supervised machine learning algorithm. The main purpose of this commonly used algorithm is to decide the labels of the data using support vectors [26]. It figures out some support vectors in order to use a certain maximum margin which then used as a decision boundary when an unlabeled data instance enters into the classification system as it is represented in Figure 3. SVM can be used as both binary categorization systems and multiclass categorization systems with "one-against-one" and "one-against-the-rest" techniques.

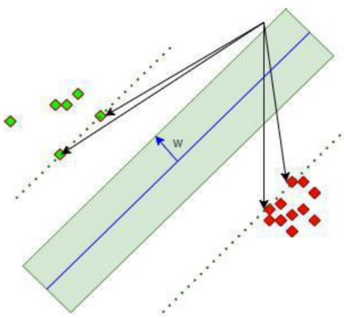


Figure 3. Representation of maximum margin in SVM

Naive Bayes Classifier (NB):
NB is again a very popular machine learning algorithm which can be easily implemented [27]. Its origin is the Bayes Theorem and uses statistical-based calculations to assign appropriate class labels for the classification tasks as it is given in Equation (13).

$$P(Class|Data) = \frac{P(Data|Class) \cdot P(Class)}{P(Data)} \quad (13)$$

where P($Data|Class$) shows the likelihood probability of the predictor given the class, P(Class) represents the prior probability of the class (y output), P(Data) signifies the prior probability of the predictor (X features) and P(Class|Data) denotes the posterior probability of the class given the predictor [28].

Random Forest (RF):
Random Forest is a machine learning model based on decision trees. It makes output predictions by combining the results of a set of regression decision trees. Each tree is independent of each other. All trees in the forest have the same distribution and all trees in the forest depend on the random vector sampled from the input data [29].

Long Short-Term Memory (LSTM):
LSTM is a branch of the Recurrent Neural Network (RNN) architecture [30]. LSTM was developed as a result of the search for a solution to the disappearing gradient problem encountered in RNN networks [31]. The general structure of LSTM consists of cell, forget, input, and output sections. Cell is the memory of the system. Input and output are the sections where the inputs and outputs to the cell are determined. The Forget section is the critical part that determines which information will be kept or deleted using the sigmoid function. A general LSTM architecture is shown in Figure 4.

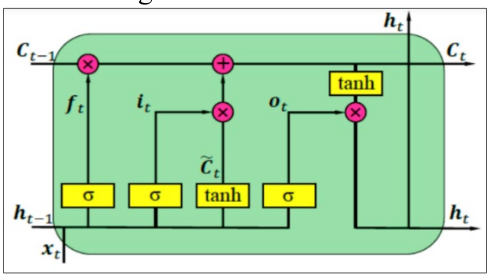


Figure 4. Representation of LSTM algorithm [35]

In order to improve the implemented LSTM model, we also apply a few regularization techniques to the output of LSTM before the softmax layer. We implemented L2 regularization [32], Manifold regularization [33] and Discrete Regularization [34] to the weight of the softmax layer.

IV. EXPERIMENTS

4.1 Experimental setting and computer specs

We apply 10-fold Cross-Validation (CV) on our dataset and report the average of those 10 runs. All experiments presented in this paper are carried out on a computer with Intel(R) CPU at 4.70 GHz with 64 GB of memory. BeautifulSoup library is used for python during the data collection from GDELT project. SVM, NB, RF, and LSTM classifiers have been implemented in Python using Python Data Analysis Library (pandas v1.3.5), Machine Learning Libraries (scikit learn 1.0.2, sklearn.ensemble and Pytorch 1.13.0). We run SVM with ‘scikit learn’ library with LinearSVC classifier with ‘C’=1, ‘dual’=’false’, ‘penalty’ = ‘l2’ parameters. Besides, LSTM’s hyperparameters used in this implementation are shown in Figure 5.

```
# Build the model
print('Build model...')
model = Sequential()
model.add(LSTM(hidden_nodes, return_sequences=False, input_shape=(word_vec_length,
char_vec_length)))
model.add(Dropout(0.6))
model.add(Dense(2))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')
batch_size=1000
model.fit(train_x, train_y, batch_size=batch_size, epochs=10, validation_data=(validate_x,
validate_y))
```

Figure 5. Hyperparameters of LSTM

4.2 Dataset

Two different datasets were used for Sentiment Polarity detection task. These datasets are collected by using 100 Turkish polysemous words as the keywords from Twitter. These polysemous words are selected from Turkish Language Institute (TDK[1]). Twelve of the selected keywords are listed in Table 8:

Table 8. Twelve of Turkish polysemous words which were used as keywords when gathering data from Twitter.

Turkish Word	English Translation
Yüz	Face
Aç	Hungry
Bin	Thousand
Doğru	True
Gül	Rose
Kara	Black
Kız	Girl
Sağ	Right
Yaş	Age
Yaz	Summer
Dil	Tongue
Yağ	Oil

The steps of this process are as follows: 1) We collected tweets for each keyword (i.e., polysemous words in our list), 2) We labelled these tweets according to the meaning of that polysemous word. 3) Then we labelled the same tweets according to their sentiment polarity as they are positive or negative.

In other words, two different labelling process (i.e., labelling according to the appropriate meaning of the polysemous word in the tweet and labelling according to the sentiment polarity of the tweet) has been performed on both datasets by human experts.

For the first dataset, we collected 30 tweets for each polysemous word in our polysemous-words list, and the first dataset finally includes 3000 tweets. For the second dataset, we collected 100 tweets for each polysemous word in our polysemous-words list, and the second dataset finally includes 10000 tweets.

Twitter-3k dataset: This dataset includes 3000 tweets, and of these, 47% are sentimentally positive and the remaining 53% are sentimentally negative.
Twitter -10k dataset: This dataset includes 10,000 tweets, and of these, 42% are sentimentally positive and the remaining 58% are sentimentally negative.

1 https://sozluk.gov.tr/

Details of these datasets are given in Figure 6.

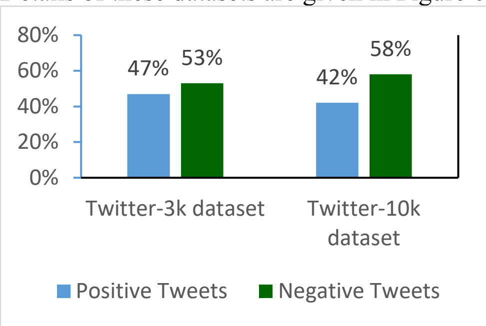


Figure 6. Properties of datasets

4.3 Sentiment-polarity corpus: GDELT Dictionary

In order to achieve sentiment polarity values of Turkish words, we took into account the dictionary which has been prepared for the study in [25]. This dictionary contains about 84,744 Turkish words. There are two stages during the building of this dictionary. 1)In the first stage, we downloaded Turkish documents from GDELT[1] (Global Data on Events, Languages, and Tone) Project using the BeautifulSoup library implemented in python. Then a cleaning process has been performed by filtering unwanted links, symbols, etc., in the text. After that, morphological analysis has been done with Zemberek library [18]. 2) In the second stage, the sentiment polarity values of words occurring in these documents have been calculated according to the Eq. (1).

In this study, we extended the dictionary in the study in [25] by downloading more documents from GDELT Project. Finally, the dictionary has about 120,000 Turkish words. The visualization of these steps are shown in Figure 7.

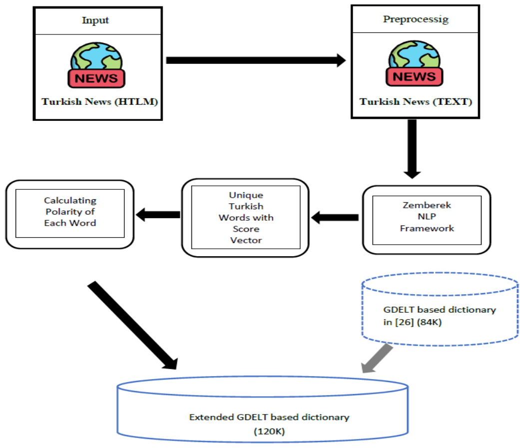


Figure 7. The visualization of building GDELT based dictionary

4.4 The list of polysemous words

We prepared a list of polysemous words which contains 100 Turkish polysemous words. Ten of them are as: “Yüz” (face), “Kız” (Girl), “Bin” (thousand), “Yaş” (Age), “Gül” (Rose), “Kara” (Black), “Sağ” (Right), “Aç” (Hungry), “Doğru” (True) and “Yaz” (Summer) are examined. The details of these polysemous words are given in Table 9. It shows the number of different meanings for each word. For instance, for the word “Yüz” (face), we examined 5000 different sentences, which were collected from GDELT Project and contained “Yüz” (face) and labeled sentences according to the meanings of the word. We detected 4 main meanings for “Yüz” (face): Hundreds (Numbers), Faces (parts of body), Surface (the top of any item), and Swimming (activity in the sea). The details about these different meanings of “Yüz” (face) are shown in Table 10.

Table 9. Number of different meanings of polysemous words

Word	# Sentences Examined	#Different Meanings
“Yüz” (Face)	5000	4
“Aç” (Hungry)	5000	2
“Bin” (Thousand)	5000	2
“Doğru” (True)	5000	2
“Gül” (Rose)	5000	2
“Kara” (Black)	5000	2
“Kız” (Girl)	5000	2
“Sağ” (Right)	5000	2
“Yaş” (Age)	5000	2
“Yaz” (Summer)	5000	2
“Dil” (Tongue)	5000	5
“Yağ” (Oil)	5000	6

4.5 Sentiment analysis of polysemous words

We found the meaning of Turkish polysemous words in their corresponding context and calculated the polarity value according to the related meaning. And the results obtained for each word are shown in Tables 10-19. For all meanings of the word "Yüz" (Face), we expect positive polarity values or polarity values that are close to 0 because neither of them has a negative meaning. According to the obtained results which can be seen in Table. 10, all four meanings of "Yüz" (Face) get both positive sentiment values.

Table 10. Details about the different meanings of "Yüz" (Face).

Class	Meaning	#Different Meanings
1	Hundred	4
2	Face	2
3	Surface	2
4	Swimming	2

As a result of the classification study, for the word "Aç" (Hungry), for the first meaning, we expect a negative sentiment polarity value because it has a negative meaning, and for the second meaning, we expect a positive polarity value because it has not a negative meaning. According to the calculated results which can be seen in Table 11, the first class gets negative polarity value and the second class gets positive polarity value as expected.

Table 11. Details about the different meanings of "Aç" (Hungry).

Class	Meaning	Average Score	Polarity
1	Hungry	-0.6157	
2	Opening	1.4096	

"Bin" (Thousand) has 2 meanings: Thousands and Get on. For all meanings of the word "Bin" (Thousand), we expect positive polarity values or polarity values that are close to 0 because neither of them has a negative meaning. According to the obtained results which can be seen in Table 12, both meanings have positive polarity score.

Table 12. Details about the different meanings of "Bin" (Thousand).

Class	Meaning	Average Score	Polarity
1	Thousand	0.2103	
2	Get on	0.3050	

As a result of the classification study for the word "Doğru" (True), for both meanings, we expect positive polarity values or polarity values that are close to 0, because neither of them has a negative meaning. Still, we human beings expect the positive sentiment polarity score of the first meaning will be higher than the positive sentiment polarity score of the second meaning, such as consciousness. According to the calculated scores which are reported in Table 13, the first meaning of the word "Doğru" (True) has 3.2103 positive polarity score, while the second meaning of the word "Doğru" (True) has 0.1700 positive polarity score.

Table 13. Details about the different meanings of "Doğru" (True).

Class	Meaning	Average Score	Polarity
1	True (not lie)	3.2103	
2	a Line in Mathematics	0.1700	

As a result of the classification study, for the word "Gül" (Rose), for the first meaning, we expect a positive polarity value or polarity value that is close to 0, and for the second meaning, we expect a positive polarity value, because neither of them has a negative meaning. According to the obtained results which are listed in Table 14, both meanings get positive sentiment polarity score. However, the first meaning of the word "Gül" (Rose) has 1.2103 positive polarity score while the second meaning of the word "Gül" (Rose) has 3.1700 positive polarity score.

Table 14. Details about the different meanings of "Gül" (Rose).

Class	Meaning	Average Score	Polarity
1	Rose	1.2103	
2	Laughing	3.1700	

As a result of the classification study for the word "Kara" (Black), we human beings expect a neutral sentiment polarity score which is very close to zero since the first meaning reflects neither negative or positive feeling to the readers. According to the experimental results listed in Table 15, the second meaning of "Kara" (Black) has a negative sentiment score, while the first meaning of "Kara" (Black) has a positive sentiment score.

Table 15. Details about the different meanings of "Kara" (Black).

Class	Meaning	Average Polarity Score
1	The part of the earth not covered by the sea, soil	0.2103
2	Black	-2.1700

As a result of the classification study for the word "Kız" (Girl), we human beings expect a neutral sentiment polarity score which is very close to zero since the first meaning reflects neither negative or positive feeling to the readers. Besides, we human beings expect a negative sentiment polarity score since the second meaning reflects a negative feeling to the reader. According to the experimental results listed in Table 16, the second meaning of "Kız" (Girl) has a negative sentiment score, while the first meaning of "Kız" (Girl) has a positive sentiment score.

Table 16. Details about the different meanings of "Kız" (Girl).

Class	Meaning	Average Polarity Score
1	Girl	0.2103
2	Being angry	-4.1700

"Sağ" (Right) has 2 meanings and we expect positive polarity values or polarity values that are close to 0 because neither of them has a negative meaning. According to the obtained results which can be seen in Table 17, both meanings have positive polarity score.

Table 17. Details about the different meanings of "Sağ" (Right).

Class	Meaning	Average Polarity Score
1	Right -indicating direction, side	0.2103
2	Alive	1.1700

As a result of the classification study for the word "Yaş" (Age), we human beings expect a neutral sentiment polarity score which is very close to zero since the first meaning reflects neither negative or positive feeling to the readers. Besides, we human beings expect a negative sentiment polarity score since the second meaning reflects a negative feeling to the readers. According to the experimental results listed in Table 18, the second meaning of "Yaş" (Age) has a negative sentiment score, while the first meaning of "Yaş" (Age) has a positive sentiment score.

Table 18. Details about the different meanings of "Yaş" (Age).

Class	Meaning	Average Polarity Score
1	Age	0.2103
2	Wet	-2.1700

As a result of the classification study for the word "Yaz" (Summer), we human beings expect a neutral sentiment polarity score which is very close to zero since the second meaning reflects neither negative or positive feeling to the readers. Besides, we human beings expect a positive sentiment polarity score since the first meaning reflects a positive feeling to the readers. According to the experimental results listed in Table 19, the second meaning of "Yaz" (Summer) has a negative sentiment score, while the first meaning of "Yaz" (Summer) has a positive score.

Table 19. Details about the different meanings of "Yaz" (Summer).

Class	Meaning	Average Polarity Score
1	Summer	2.2103
2	Writing	0.1700

4.6 Experimental results and discussion

In order to show the effectiveness of the presented approach, we conducted some experiments on Twitter-3k and Twitter-10k datasets. As it is reported in Section 4.2, all datasets used in the experiments are not balanced and we list the experimental results with F1 score as it is given in Equation 14:

$$\text{F1 Score} = (2 * Precision * Recall)/(Precision + Recall) \quad \mathbf{(14)}$$

where $Precision$ shows the Precision value, $Recall$ represents the Recall value and basically *F1 score* is the harmonic mean of precision and recall.

Table 20 presents the experimental results of the extended GDELT dictionary-based ensemble classifier with extracted features $\{F_1\text{-}F_{10}\}$, SVM, NB, RF, and LSTM classifiers on Twitter-3k and Twitter-10k datasets. According to Table 20, the F1 scores of the proposed ensemble classifier, SVM with TF, SVM with extracted features $\{F_1\text{-}F_{10}\}$, NB with TF, NB with extracted features $\{F_1\text{-}F_{10}\}$, RF with TF, RF with extracted features $\{F_1\text{-}F_{10}\}$, LSTM with TF and LSTM with extracted features $\{F_1\text{-}F_{10}\}$ are 91.56%, 87.12%, 89.24%, 86.34%, 87.93%, 89.67%, 90.78%, 90.18% and 91.45% on Twitter-3k dataset; respectively. Additionally, the F1 scores of the proposed ensemble classifier, SVM with TF, SVM with extracted features $\{F_1\text{-}F_{10}\}$, NB with TF, NB with extracted features $\{F_1\text{-}F_{10}\}$, RF with TF, RF with extracted features $\{F_1\text{-}F_{10}\}$, LSTM with TF and LSTM with extracted features $\{F_1\text{-}F_{10}\}$ are 92.63%, 88.22%, 89.71%, 87.40%, 88.56%, 90.75%,91.80%,91.84% and 92.11% on Twitter-10k dataset; respectively. The superiority of the proposed ensemble algorithm over other algorithms could be explained as with several reasons: 1) Using features $\{F_1\text{-}F_{10}\}$ extracted from the extended GDELT based dictionary, which clearly seems to have an improvement on the classification task in compared to using features extracted from conventional TF, 2) The extended capacity of GDELT dictionary with approximately 159,876 Turkish words, 3) Majority-voting scheme based ensemble classifier, 4) WSD module in which different meaning of Turkish ambiguous words are taken into consideration, 5) Negation handling module.

Table 20. F1 Scores of GDELT dictionary-based ensemble classifier with extracted features $\{F_1\text{-}F_{10}\}$, SVM, NB, RF, and LSTM classifiers on Twitter-3k and Twitter-10k dataset

Dataset	Classifier	F1% score
Twitter-3k	*Extended GDELT Dictionary based ensemble classifier with extracted features $\{F_1\text{-}F_{10}\}$ and MV*	***91.56***
	SVM with TF	87.12
	SVM with extracted features $\{F_1\text{-}F_{10}\}$	89.24
	NB with TF	86.34
	NB with extracted features $\{F_1\text{-}F_{10}\}$	87.93
	RF with TF	89.67
	RF with extracted features $\{F_1\text{-}F_{10}\}$	90.78
	LSTM with TF	90.18
	LSTM with extracted features $\{F_1\text{-}F_{10}\}$	91.45
Twitter-10k	*Extended GDELT Dictionary based ensemble classifier with extracted features $\{F_1\text{-}F_{10}\}$ and MV*	***92.63***
	SVM with TF	88.22
	SVM with extracted features $\{F_1\text{-}F_{10}\}$	89.71
	NB with TF	87.40
	NB with extracted features $\{F_1\text{-}F_{10}\}$	88.56
	RF with TF	90.75
	RF with extracted features $\{F_1\text{-}F_{10}\}$	91.80
	LSTM with TF	91.84
	LSTM with extracted features $\{F_1\text{-}F_{10}\}$	92.11

There are two important outcomes of these experimental results: 1) It is possible to improve the classification performance by using multiple learning algorithms in a majority-voting scheme-based ensemble format instead of using these learning algorithms in a conventional way (i.e. using those algorithms independently from each other in a singular way). 2) Text representations with the features $\{F_1\text{-}F_{10}\}$ which are extracted from the extended GDELT based dictionary to compare to text representation with traditional TF, seem to have a positive effect on the sentiment analysis task.

In WSD module, different sentiment polarity scores for each different meaning of polysemous words are calculated. Consequently, this process discriminates classes successfully and results in high classification performance.

V. CONCLUSION AND FUTURE DIRECTIONS

By considering that there is a lack of studies on Turkish sentiment analysis for polysemous words, we prepared a polarity calculation module to calculate the polarity values of the polysemous words so that we examined whether the polarity values of this words differ according to the meaning they have. Then, since negation directly affects the correct meaning of the word in the sentiment, we prepared the negation handler module to take into account both forms of negation of Turkish language in the polarity classification.

We also prepared the sentiment polarity corpus which consists of 159,876 including 100 polysemous words. Negation in Turkish usually appears in two forms. It may appear in a word form or may appear in a suffix form. A negation module is implemented to handle both forms of negation. A rule-based approach was used for the negation module.

In other words, we built a system for Turkish sentiment polarity detection task including the following modules: 1) Pre-processing, 2) Polarity Calculation Module, 3) Negation Handling Module, 4) Feature Generation Module and 5) Classification Module. According to our knowledge, this is the first study which includes all of these modules in one Turkish sentiment analysis task. Actually, the basic goal of this study was to show that the polarity scores of polysemous words could change from their context to context since they can have different meanings in different contexts. As far as we know, this is the first effort for Turkish language. By making statistical calculations and morphological analysis, we calculated different scores for the corresponding different meanings of about 100 Turkish polysemous words. This study contributes to the literature due to its unique property to be the first in kind.

We conducted several experiments: In the first set of experiments, we tried to show the different meanings of polysemous words could have different sentiment polarity scores. This way, we attempted to observe the efficiency of the extended sentiment polarity dictionary for polysemous words over the traditional sentiment polarity dictionary on the classification task. In the second set of experiments, we attempted to show the effectiveness of the generated features from GDELT dictionary are superior to traditional TF features for text representation on the sentiment analysis task. We also aimed to analyse the MV technique-based ensemble algorithm advancing the system performance.

The F1 scores gathered from the experiments, shown in Table 20, motivated us that the extended sentiment polarity dictionary with polysemous words has the ability to increase the classification performance. This encouraged us to extend the sentiment polarity dictionary from polysemous words to increasingly polysemous words. According to the experimental results in Table 20; it seems possible to improve the classification performance by using multiple learning algorithms in a majority-voting scheme-based ensemble format instead of using these learning algorithms in a conventional way (i.e. using those algorithms independently from each other in a singular way). Besides, it could be deduced from the experimental results in Table 20 that the text representation with the features $\{F_1\text{-}F_{10}\}$ which are extracted from the extended GDELT based dictionary to compare to text representation with traditional TF seems to have a positive effect on the sentiment analysis task.

In the future, to make further analysis of Turkish sentiment analysis for polysemous words, we will examine more polysemous words, increase the size of the dataset, and use more different morphological analyzer tools. Furthermore, rule-based methods will be studied for both classification and negation handler modules.

ACKNOWLEDGEMENT
This work is supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) grant number [120E187]. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of TÜBİTAK.

REFERENCES

[1] Navigli, R., Word sense disambiguation: A survey. ***ACM Comput Surv***, 41(2), 1-69, (2009).
[2] Boyd-Graber, J., Blei, D. & Zhu, X.A. Topic model for word sense disambiguation. ***In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning*** (EMNLP-CoNLL), *Prague*, (2007).
[3] Açıkgöz, O., Gürkan, A. T., Ertopçu, B., Topsakal, O., Özenç, B., Kanburoğlu, A. B., & Yıldız, O. T. All-words word sense disambiguation for Turkish. ***In International Conference on Computer Science and Engineering (UBMK)***, *Antalya, Turkey,* (2017).
[4] Orhan, Z., & Altan, Z. Effective features for disambiguation of Turkish verbs. ***Int J. Comp and Inf Eng***, 1(7), 2264-2268, (2007).
[5] Gezici, G., & Yanıkoğlu, B. Sentiment analysis in Turkish. ***Turkish natural language processing***, 255-271, (2018).
[6] Türkmenoglu, C., & Tantug, A. C. Sentiment analysis in Turkish media. ***In International Conference on Machine Learning (ICML)***, *Beijing, China,* (2014).
[7] Çetiner, M., Yıldırım, A., Onay, B., & Öksüz, C. Word Sense Disambiguation using KeNet. ***In 29th Signal Processing and Communications Applications Conference (SIU),*** *Istanbul, Turkey*, (2021).
[8] Mert, E., & Dalkilic, G. Word sense disambiguation for Turkish. ***In 24th International Symposium on Computer and Information Sciences,*** *Cyprus,* (2009).
[9] Aslan, D. B., Arıcan, B. N., Bayrak, T. A., Özbek, E. E., & Yıldız, O. T. Turizm Alanına Özgü Anlamsal Olarak Isaretlenmis Bir Derlem. ***An All-Words Sense Annotated Corpus for Tourism Domain***. [cited 2023 March 1].
[10] Tüysüz, M. A., & Güvenoğlu, E. Türkçe için karşılaştırmalı bir kelime anlamı belirginleştirme uygulaması. ***In Proceedings of Akademik Bilişim Conferene***, *Mersin*, (2014).
[11] Arslan, E., Orhan, U., & Tahiroğlu, B. T. Serbest Sırada Birliktelik İstatistiklerinin Kullanımıyla Türkçe'nin Biçimbirimsel Belirsizliği'nin Giderilmesi. ***Gümüşhane Üniversitesi Fen Bilimleri Dergisi***, 46-52, (2018).
[12] Aydın, Ö., & Kılıçaslan, Y. Tümevarımlı Mantık Programlama İle Türkçe İçin Kelime Anlamı Belirginleştirme Uygulaması. ***In Proceedings of Akademik Bilişim Conferene***, *Muğla, (2010).*
[13] Selamet, D. T., & Eryiğit, G. C. Veri Artırımı için Yarı-Denetimli Bağlamsal Anlam Belirsizliği Giderme. ***Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi***, 14(1), 34-46, (2021).
[14] Taşar, D. E., Özcan, C., & Koruyan, K. Autotrain Yaklaşımı ile Duygu Analizi. ***In Proceedings of USBILIM 2nd International Conference on Artifical Intelligence Studies***, *Ankara*, (2022).
[15] Çılgın, C., Gökçen, H., & Gökşen, Y. Twitter'da COVID-19 aşılarına karşı kamu duyarlılığının çoğunluk oylama sınıflandırıcısı temelli makine öğrenmesi ile duygu analizi. ***Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi***, 38(2), 1093-1104, (2023).
[16] Balli, C., Guzel, M. S., Bostanci, E., & Mishra, A. Sentimental analysis of Twitter users from Turkish content with natural language processing. ***Comp Intel and Neurosci***, 2022(3),1-18, (2022).
[17] Kirelli, Y., & Arslankaya, S. Sentiment analysis of shared tweets on global warming on twitter with data mining methods: a case study on Turkish language. ***Comp Intel and Neurosci***, 2020(1),1-9, (2020).
[18] Akın, A. A., & Akın, M. D. Zemberek, an open source NLP framework for Turkic languages. ***Structure***, 10(2007), 1-5, (2007).
[19] Günyaktı, R. İ., & Bursa, N. Covid-19 Pandemisinde Sağlık Çalışanları ve Öğretmenlere Yönelik Algının Duygu Analizi ile Twitter Verileri Üzerinden İncelenmesi. ***Selçuk İletişim***, 15(1), 264-285, (2022).
[20] Kandıran, E., Gümüş, B., & Özer, M. A. Covid-19 pandemi sürecinde uzaktan eğitimin twitter yansımalarının duygu analizi. ***Int J Social Sci and Educ Res***, 8(3), 228-242, (2022).
[21] Yüksel, A. S., & Tan, F. G. Metin madenciliği teknikleri ile sosyal ağlarda bilgi

keşfi. ***Mühendislik Bilimleri ve Tasarım Dergisi***, 6(2), 324-333, (2018).

[22] Köksal, A. Özgür. Twitter dataset and evaluation of transformers for Turkish sentiment analysis. ***In 2021 29th Signal Processing and Communications Applications Conference (SIU)***, *Istanbul, Turkey*, (2021).

[23] Güran, M. Uysal, Ö. Doğrusöz. Destek vektör makineleri parametre optimizasyonunun duygu analizi üzerindeki etkisi. ***Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi***, 16(48), 86-93, (2014).

[24] Sağlam, F., Sever, H., & Genç, B. Developing Turkish sentiment lexicon for sentiment analysis using online news media. ***In IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)***, *Morocco*, (2016).

[25] Altınel, A. B., Buzlu, K., & İpek, K. Performance Analysis of Different Sentiment Polarity Dictionaries on Turkish Sentiment Detection. ***In 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)***, *Biarritz, France*, (2022)

[26] Boser, B. E., Guyon, I. M., & Vapnik, V. N. A training algorithm for optimal margin classifiers. ***In Proceedings of the fifth annual workshop on Computational learning theory,*** *New York,* (1992).

[27] Berrar, D. Bayes' theorem and naive Bayes classifier. ***Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics***, 403, 412, (2018).

[28] Jain, K. (2021). What is Naive Bayes? Published in Analytics Vidhya. [cited 2023 March 6].

[29] Breiman, L. Random Forests, ***Mach Learn***, 1,5-32, (2001).

[30] Hochreiter, S., & Schmidhuber, J. Long short-term memory. ***Neural comput***, 9(8), 1735-1780, (1997).

[31] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. Recurrent neural network based language model. ***In Interspeech***, (2),3, 1045-1048, (2010).

[32] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. ***arXiv preprint arXiv***:1207.0580, (2012).

[33] Belkin, M., Niyogi, P., & Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. ***Jour of machine learning research***, 7(11), (2006).

[34] Belkin, M., Matveeva, I., & Niyogi, P. Regularization and semi-supervised learning on large graphs. ***In Proceedings of Learning Theory: 17th Annual Conference on Learning Theory, COLT***, *Banff, Canada*, (2004).

[35] Kara, A. Global solar irradiance time series prediction using long short-term memory network. ***Gazi Üniversitesi Fen Bilimleri Dergisi, Part C: Tasarım ve Teknoloji***, 4(7), 882-892, (2019).