International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

TurkiS: An Automatic Labeled Dataset Generator for Turkish Sentiment Analysis

Emrah Inan[1, *], Fatih Soygazi[2] and Vahab Mostafapour[3]

Submitted: 03.12.2018 Accepted : 03/05/2019

Abstract: A preliminary task of sentiment analysis aims to detect polarities of a text either positive or negative. To increase the overall performance of the polarity detection for supervised learning methods, it requires properly labeled training texts. Also, the quality of labeled texts is critical for correct polarity detections. In this study, we provide a training and test dataset generator for Turkish sentiment analysis in which supervised learning methods can be trained without any human labor. To achieve these requirements, we extract comments and polarity scores from a popular e-commerce website for electronic devices in Turkey. Also, we employ a lexicon-based polarity detection to filter automatically labeled comments which exploit a translated version of a well-known sentiment lexicon into Turkish. We compare well-known supervised learning methods trained by both unfiltered and filtered versions of this dataset. The experimental setup is conducted by the generated evaluation dataset for the e-commerce domain. The test set contains 30% randomly selected from the generated dataset. Experimental results show that all supervised learning methods including SVM with linear kernel model, Multinomial Naive Bayes, and Logistic Regression perform 8.2 percent better in the filtered version of the dataset than the unfiltered version. Moreover, the logistic regression gets the highest score when it uses count vectorizer as a feature extraction mechanism.

Keywords: *Automatic Training Set, SenticNet, Sentiment Analysis.*

1. Introduction

Sentiment analysis is mainly interested in analysing the opinions and attitudes of people over products, organizations, and events [1]. These tools are most common used in the market analysis, following trends and monitoring public attitudes such as customer complaints or satisfaction.

Sentiment analysis approaches mainly focus on lexicon-based and supervised learning methods. There are advantages and disadvantages in terms of accuracy and human labor for both methods. Supervised learning methods such as Support Vector Machine (SVM) [2] and Naive Bayes [3] are commonly used in sentiment analysis. Although these methods achieve remarkable results, they require manually labeled datasets. To reduce the cost of human labor, lexicon-based methods depend on well-developed dictionaries such as SenticNet and SentiWordNet. However, lexicon-based methods are not as accurate as supervised learning methods. In this study, our goal is to enhance the quality of generated dataset by using a lexicon-based method.

This study concentrates on developing an open source Turkish sentiment analysis dataset called TurkiS that involves two versions. The first version is an unfiltered dataset that includes directly automatic extraction of comments from e-commerce sites. The second one is filtered dataset and it comprises the proper comments including polarity labels which are proved by a lexicon-based polarity detection method. To extract user comments from e-commerce websites, we consider comments including rating values. After this process, we detect polarity scores of the words in given comments. Then we evaluate sentence polarities using word polarity scores of extracted comments.

We compare ten supervised learning methods in this study for both datasets. The F1 score of the lexicon-based method is 62.4% for e-commerce dataset. To execute a supervised learning based sentiment analysis, we have investigated several methods. We evaluate different classifiers on automatically generated test dataset with distinct feature extraction techniques including word counts and inverse term frequency with the bag of words representation. The highest F1-score of supervised learning based methods are 89.0% for SVM, logistic regression, and Multinomial Naive Bayes.

The rest of this paper is organized as follows: Section 2 gives an overview of related work. In Section 3, the approach of TurkiS is proposed for the e-commerce domain. The experiments are shown for machine learning and lexicon-based techniques on the prepared evaluation dataset in Section 4. We conclude our study and highlight the research questions in Section 5.

2. Related Work

Ravi and Ravi [4] present a comprehensive review of publications during 2002-2015 related to the sentiment analysis area. It provides a task categorization of sentiment analysis in terms of the *problem addressed, exploited dataset details, feature representation* and *selection method for applied architectures*. Moreover, they categorize polarity determination into *ontology, non-ontology, machine learning, lexicon,* and *hybrid approach*. Most of the task application approaches are classified under *the machine learning, lexicon* and *hybrid* categorization, while the application approaches of *lexicon* and *aspect creation* tasks are classified into *ontology-based* and *non-ontology-based* categorization.

Machine learning techniques are commonly used in the task of polarity detection for sentiment analysis. Naive Bayes and Support Vector Machine (SVM) are the most used of them in sentiment

[1]*National Centre for Text Mining, School of Computer Science, The University of Manchester, United Kingdom ORCID ID: https://orcid.org/0000-0002-1229-6895*
[2]*Department of Computer Engineering., Adnan Menderes University, Aydın, Turkey ORCID ID: https://orcid.org/0000-0001-8426-2283*
[3]*Department of Computer Engineering., Ege University, İzmir, Turkey ORCID ID: https://orcid.org/0000-0003-4414-5706*
**Corresponding Author Email: emrahinan@gmail.com*

analysis tasks [3], [5] and [6]. Pang and Lee [3] propose one of the primary works in sentiment classifiers that compare the performance of machine learning algorithms with human-produced baselines on the movie reviews dataset and found that this method outperformed the results.

Catal and Nangir [2] present the investigation about the potential benefit of multiple classifier systems concept on Turkish sentiment classification problem. After checking out the performance of many of the classification algorithms in Weka on their datasets, they found Naive Bayes and SVM classifiers with the best results. They also increased their performance by combining multiple classifier systems.

Multinomial Naive Bayes is a specialized version of Naive Bayes that is designed more for text documents [3]. Whereas simple Naive Bayes would model a document as the presence and absence of particular words, Multinomial Naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with it [2].

Dehkharghani et al. have published SentiTurkNet [7] as a Turkish polarity lexicon that assigns positive, negative and neutrality polarities to each 15k synsets extracted from TurkishWordNet [8]. In another paper, they have translated SenticNet to the Turkish language by taking overall scores of 14k entries that are in word or phrase form, directly from SenticNet [9]. In the case of a specific domain, our method is mainly inspired by the works of Wu and Huang [10] that presented a sentiment domain adaptation with multiple sources system. They outperformed state-of-the-art sentiment adaptive systems by using sentiment graph. According to their work, sentiment classifiers that are trained from global polarity or from a domain are affected. Then preparing a dataset or lexicon in a specific domain will outperform the classifiers. In other words, training a sentiment classifier domain independently or in the movie or book domains will lose its accuracy to predict sentiments in the electronics domain. Because the same word has a different sentiment in different domains [10]. For example, the word 'basit' (means easy) in general speaking has positive polarity in the electronic domain but negative polarity in movie or book domain means low level in the Turkish language.

For Turkish sentiment analysis dataset preparation, Makinist et al. [11] present a dataset generation system from social media and global web. Their system is based on Apache ManifoldCF (MCF) (https://manifoldcf.apache.org/). MCF is used for web search and user comments. They are focused on spelling correction and distributed file systems that are so important in social web analysis. Generating users review data from www.hepsiburada.com for the electronic domain is similar to this work that is using Jsoup Java HTML Parser (https://jsoup.org/). Our work is different in the selected domain that is a key objective in the polarity of terms. Moreover, we are preparing a domain-specific lexicon based on SenticNet.

SenticNet is a preliminary tool used for tracking and predicting public trends. In addition, there are also domain-specific sentiment analysis tools interested in news and blogs [12]. SenticNet assigns polarity scores for the given words, and it is the fundamental task of the sentiment analysis [3]. Measuring polarities is even clear when the given text contains explicit opinion such as adjectives (good, bad) or verbs (love, hate). However, implicit opinions tackle in identifying polarity determination. Since these sort of opinions contain unpopular or rarely used terms which cannot be identified in SenticNet. Therefore, a method which can cover implicit opinions by enriching SenticNet is required. In this study, we develop a Word2Vec model to enrich SenticNet in a large scale environment.

Giatsoglou et al. [5] provide a generic methodology based multilingual sentiment detection system to extract subjective and useful information mainly for marketing and business. To train their model of polarity classifier, they used a hybrid model considering lexicon and word embeddings based vector representation of texts. In addition to lexicon-based features, they capture labeled documents including sentiment polarity values. To generate semantic and syntactic features Word2Vec model [13] is applied. Also, four data sets involving online user reviews in both Greek and English languages are used to evaluate the system [5]. In our work, we present a lexicon and labeled dataset in the Turkish language that is usable for training classifiers in the Electronic domain related to the language.

Zhang et al. [14] present a combined sentiment detecting system with a lexicon-based module and a binary classifier. To improve the low recall and F-measure of the lexicon-based module, an SVM binary classifier is employed for labeled tweets in lexicon based module. Training data for the supervised classifier is fed by outputs of an unsupervised module. A binary feature value vector (instead feature frequency vector) of features that includes basic features such as unigrams and advanced ones such as emoticons and hashtags to Twitter data is used [15].

3. Method

TurkiS begins with extraction phase of the e-commerce website for preparing the unfiltered dataset for supervised learning methods as denoted in Fig. 1. Based on the ratings, we split comments into positive and negative texts and we obtain an equal number of positive and negative comments. Meanwhile, we also translate SenticNet into Turkish for the e-commerce domain. Then we use the translated SenticNet into a lexicon-based polarity detection method to generate the filtered dataset.

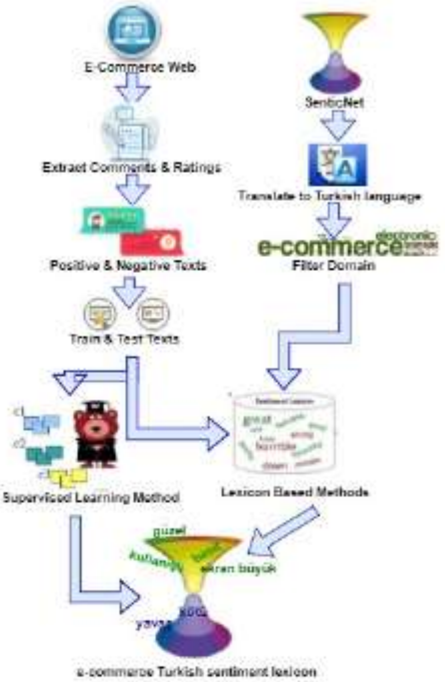

Fig 1. General structure of TurkiS

3.1. Unfiltered Dataset Preparation

For sentiment analysis, a dataset about electronics domain is used to apply our method in this study. This dataset is gathered from the comments of a popular e-commerce website (https://www.hepsiburada.com/) in Turkey. The comments are classified by the ratings from 1-5. 1 and 2 points correspond to negative comments while 4 and 5 points mean express positive thoughts. The comments getting 3 points are not considered because of assuming them as neutral. The number of comments is shown in Table 1 according to the positive or negative classifications for the cellphone category of the chosen website.

Table 1. Rating values and number of comments for unfiltered dataset

Rating	# Comment
1	863
2	479
4	479
5	863

As illustrated in Table 1, there is no correction during the construction of the unfiltered dataset. All comments and their polarity labels are directly extracted by the given website. A lexicon-based method is applied to enhance the quality of the second version of this extracted dataset as explained in the following subsection in detail.

3.2. Filtered Dataset Preparation

Initially, we translate a proper sentiment analysis lexicon to use it in a lexicon-based method. Then we generate a filtered version of the automatic labeled raw dataset with this method.

3.2.1. Lexicon Translation

There are some widely used sentiment analysis lexicons such as SentiWordNet [16] and SenticNet [17]. These studies are based on English and a vast majority of studies exploit English text for sentiment analysis of various datasets.

In SenticNet, words are considered with their implicit meanings of commonly accepted concepts. Instead of caring about primarily on syntactic issues, SenticNet assigns polarity values those words between -1 and 1. Minus sign corresponds to negative polarities of commonsense knowledge about a word while positive sign sets an optimistic value to the words.

In this study, we use an English-to-Turkish dictionary named Tureng (http://tureng.com/) including the words and their categories for translation of the knowledge base with 50000 entries in SenticNet 4 (http://sentic.net/senticnet-4.0.zip). We translate SenticNet with the help of Tureng and obtain a tuple of lines as:

<English Word> -> <Turkish Translation> -> <Category> -> <Polarity>

During the translation process, if an English word includes a translation in Teknik (Technical in English) category, these translations are just selected and other translations are omitted for that word. Otherwise, the first three translations of the word are written into translated polarity file. Translation document involves a dense number of technical words with this approach. Therefore, most of the matched words in the e-commerce website fits the cellphone category technically.

3.2.2. Lexicon Based Dataset Validation

Our main objective in this subsection is to validate the polarities of the raw generated dataset and produce a new version of this dataset including polarity corrected comments. To get more accurate results, Algorithm 1 handles each word in comments to predict a weighted sentence polarity value by summing up the polarity scores of the words.

Initially, we preprocess the comment lines and return a usable corpus. Firstly, we remove punctuation marks, then we segment each comment into words and transform into lemmatized form by Turkish NLP library, Zemberek [18]. Secondly, we deasciify words with actual word in Turkish like *kotu -> kötü (bad); cok -> çok (many); degil -> değil (not)*. We especially pay attention to the adjectives (bad), intensifiers (many) and negation words (not). Because these type of words affect the polarity of a sentence more than domain-specific words like cellphone or product name. These kinds of words are also seen frequently in the comments and affect the overall performance. Hence, applying a replacement process for the specified words increase the accuracy of prediction.

Input: D_c : Comment document
W_{lwl} : Lemmatized word list
D_s : SenticNet polarity document
P_{lwl} : Turkish translated SenticNet word polarities in the comment sentence
P_{ts} : Total comment sentence polarity
wp : Number of words with polarity in the comment sentence
P_{ws} : Weighted comment sentence polarity

1 $W_{lwl} \leftarrow lemmatizeComments(D_c)$
2 $P_{lwl} \leftarrow calculateWordPolarities(W_{lwl}, D_s)$
3 $P_{ts} \leftarrow calculateWeightedWordPolarities(P_{lwl}) + calculateWeightedWordPolaritieswithNegation(P_{lwl})$
4 $P_{ws} \leftarrow P_{ts}/wp$
5 $updateConfusionMatrix(P_{ws})$

Algorithm 1. Lexicon-based sentiment algorithm for a comment sentence

Lemmatization analyzes words morphologically, removes inflectional endings and returns the searchable part of the word in a dictionary. Step 1 comprises lemmatization function using Zemberek NLP tool as denoted in Algorithm 1. If the word is a lemmatized word, that word is compared with the translated SenticNet. When there is a match between the translated SenticNet document and lemmatized word, that word is flagged as a word with polarity. Then the category of the word is tried to match with "Teknik" (Technical) category.

In Step 2, if a translated word has "Teknik" meanings, it affects overall polarity less. Emphasizing this assumption, the polarity of each word in the translation of a comment is calculated by adding the actual word polarity divided to the total number of words in the comment (*wc*) if the category is "Teknik" shown in Eq. 1.

$$P_{dw} = P_{dw} + \frac{P_w(Teknik)}{wc} \quad (1)$$

In this equation, P_{dw} indicates the total domain specific polarity of the word w where the domain is “Teknik” in our scenario for Eq.1. $P_w(Teknik)$ denotes the polarity of a word where the translation of the word is in “Teknik” category and wc specifies the total number of words in a sentence. Otherwise (the translated word does not involve a “Teknik” meaning), word polarity is added to the total word polarity directly supposing this word is not a domain-specific word as expressed in Eq. 2.

$$P_{dw} = P_{dw} + P_w \quad (2)$$

Here, P_w specifies the SenticNet translated polarity of each word if it is not classified under a defined category. After attaining all matched words for that comment and added to the specific word polarity, weighted word polarity (P_{ww}) is calculated by dividing total word polarity by the number of translations of a word with polarity in SenticNet (*wpc*). *wpc* can be much more than *wc* where each word can have many translations and polarity values in Turkish translated SenticNet.

$$P_{ww} = \frac{\mathrm{P_{dw}}}{wpc} \quad (3)$$

In Turkish, *yok* and *değil* words give an inverted meaning to the sentence as negation words. *"Telefon kılıfının herhangi bir sorunu yok" (There is not any problem with cellphone case)* sentence actually has a positive meaning. But after making just lexicon-based syntactical process, this sentence is resolved wrongly.

For negation words, reweighted sentence polarity is changed by inverting the sign of each weighted word polarity except technical words. Because in the syntactical process, weighted word polarity was used with a wrong polarity sign. Therefore, weighted word polarities are inverted by the morphological process according to the semantics of the sentence and the indicated Turkish sentence is classified correctly. For negation word check, weighted word polarities of only nontechnical words are noticed. For example, "telefon" (cellphone) and "kılıf" (case) words do not affect the polarity of the sentence directly.

Morphemes *+me* and *+ma* negate the meaning of the verbs in Turkish. The sentence including a negated verb mostly determines a negative meaning accordingly. For example, *"Telefonun şarjı uzun süre gitmiyor." (Cellphone battery doesn't stay charged for long)* the sentence has a negative meaning. To improve the negative polarity of this sentence, technical words are considered to be inverted. Technical words for the example sentence are *"telefon" (cellphone)* and *"şarj" (battery)*. Therefore, opposite to the process for negation word, the weighted polarity of technical words in sentences including negated verbs are inverted and summed to get reweighted sentence polarity.

The polarity calculation for the words of the comments involving negation words or morphemes is performed as shown in Eq. 4.

$$P_{wwn} = -P_{ww} \qquad (4)$$

Finding weighted word polarities help us to compute total sentence polarity by adding all weighted word polarities in a comment (Step 3). Total sentence polarity value is equal to the addition of each word polarity either having regular or negated word polarities (Eq. 5).

$$\mathrm{P}_{ts} = \sum \mathrm{P}_{\mathrm{ww}} + \sum \mathrm{P}_{wwn} \qquad (5)$$

After doing this process for each word at the end of the comment sentence, weighted sentence polarity (P_{ws}) is evaluated by dividing total sentence polarity to the number of words with polarity value in that comment (Step 4). P_{ws} is the value to analyze the sentiment of a review (Eq. 6).

$$\mathrm{P}_{ws} = \frac{\mathrm{P}_{ts}}{wp} \qquad (6)$$

In Eq. 6, *wp* represents the number of words with polarity in a comment by skipping non-polarity words. For example, articles and prepositions mostly do not affect the polarity of a comment. Therefore, these kind of words are not considered as they are not involved in the translated SenticNet while calculating P_{ws}.

Weighted sentence polarity is used to examine whether the prediction gives a correct result or not (Step 5). After calculating the total polarity for each comment, we compare the output of lexicon-based method with already given polarity value of each comment in the unfiltered dataset. If both polarity values are equal, we add them to the filtered dataset. As shown in Table 2, the number of comments having rating 1 has the biggest decline in the filtered dataset. Also, the comments including rating 5 almost 25 percent smaller than the same comments in the unfiltered dataset.

Table 2. Rating values and number of comments for filtered dataset

Rating	# Comment
1	429
2	217
4	351
5	657

The total number of both positive and negative comments in unfiltered dataset is 2684. On the other hand, the total number of both positive and negative comments in filtered dataset reduced to 1654.

4. Experimental Results

In this study, the experiments are conducted by the generated evaluation dataset for the e-commerce domain. To show the improvement of the filtered dataset generated by TurkiS, we evaluate supervised learning methods in both unfiltered and filtered datasets.

We employ scikit-learn tool (http://scikit-learn.org/stable/index.html) for well-known supervised learning methods such as Support Vector Classification (SVC), Multinomial Naive Bayes and Logistic Regression. We perform all algorithms in terms of either TF-IDF or Count vectorizer methods, so we can convert arbitrary textual data into numerical features usable for the selected algorithms. Both vectorizer methods involve in Bag of Words (BOW) representation.

BOW representation facilitates tokenizing, counting and normalizing for extracting numerical features from raw texts in scikit-learn. With the help of token separators like white-spaces and punctuation, raw texts are split into tokens and scikit-learn gives an integer id for each possible token. Then, the occurrences of tokens are counted in each raw text.

After all, important tokens get higher weights in the last normalizing step. CountVectorizer is a common usage in BOW representation and considers both tokenization and occurrence counting for the feature extraction. Second vectorizer method considers term frequency times inverse document frequency (TF-IDF) which is a well-known method in the information retrieval. Some token might have very present like *"the"*, *"a"* or *"is"* but these tokens give insufficient meaning to the text. Hence, TF-IDF is used here to identify rare but meaningful tokens in the raw text.

Table 3. Experimental results of unfiltered dataset

Method	Vectorizer	Precision	Recall	F1
SVC (rbf kernel)	TF-IDF	0.25	0.5	0.33
SVC (linear kernel)	TF-IDF	0.89	0.89	0.89
Multinomial Naive Bayes	TF-IDF	0.89	0.89	0.89
Logistic Regression	TF-IDF	0.89	0.89	0.89
SVC (rbf kernel)	Count	0.75	0.57	0.47
SVC (linear kernel)	Count	0.87	0.87	0.87
LinearSVC	Count	0.87	0.87	0.87
Multinomial Naive Bayes	Count	0.88	0.88	0.88
Logistic Regression	Count	0.89	0.89	0.89

Table 3 indicates methods and their precision, recall and F1 measurements. Precision is the ratio of correctly positive predictions to the total positive predictions. Recall is the ratio of correctly positive predictions to the all positive actual values. Precision measures the exactness of polarity detector, whereas Recall measures the completeness or sensitivity of this detector. F1 measure is the harmonic mean of Precision and Recall, that describes the overall performance of Turkish datasets.

To compute evaluation score for both datasets, we randomly select 70% of each dataset for the training set and the remaining part as the test set. We compare the performance of the same supervised learning methods for both datasets.

Table 4 illustrates the overall scores for the filtered dataset and F1 scores of all methods are better than the experimental results of the same methods trained by the unfiltered datasets. These scores clearly indicate that the validation step of the lexicon-based method improved the quality of automatically generated dataset. In both cases, logistic regression gets the highest score when it uses count vectorizer as a feature extraction mechanism.

Table 4. Experimental results of filtered dataset

Method	Vectorizer	Precision	Recall	F1
SVC (rbf kernel)	TF-IDF	0.37	0.61	0.46
SVC (linear kernel)	TF-IDF	0.96	0.96	0.96
Multinomial Naive Bayes	TF-IDF	0.95	0.95	0.95
Logistic Regression	TF-IDF	0.97	0.97	0.97
SVC (rbf kernel)	Count	0.37	0.61	0.46
SVC (linear kernel)	Count	0.98	0.98	0.98
LinearSVC	Count	0.98	0.98	0.98
Multinomial Naive Bayes	Count	0.96	0.96	0.96
Logistic Regression	Count	0.98	0.98	0.98

5. Conclusion

This study interests in generating automatic labeled datasets for supervised learning methods in Turkish sentiment analysis. Firstly, training dataset is automatically extracted from an e-commerce website. Later, we filter this raw dataset using the lexicon-based method. We compare both datasets for the same supervised methods. Experimental results represent that the results of each method using filtered dataset overcome the same methods if they are trained using the unfiltered dataset.

As a future work, we will enhance the training dataset of e-commerce domain with other websites in order to exploit deep learning methods. In addition, we will foster supervised methods with continuous word representations rather than using BOW model. Further, we will examine other domains to optimize our methodology.

References

[1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1-167, May 2012.

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

[3] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, Association for Computational Linguistics, 2002, pp. 79-86.

[4] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14-46, Nov. 2015.

[5] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214-224, Mar. 2017.

[6] S. V. Wawre and S. N. Deshmukh, "Sentiment classification using machine learning techniques," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 819-821, Apr. 2016.

[7] R. Dehkharghani, B. Yanikoglu, Y. Saygin, and K. Oflazer, "Sentiment analysis in Turkish at different granularity levels," *Natural Language Engineering*, vol. 23, no. 4, pp. 535-559, Jul. 2017.

[8] O. Bilgin, O. Cetinoglu, and K. Oflazer, "Building a Wordnet for Turkish," *Romanian Journal of Information Science and Technology*, vol. 7, no. 1-2, pp. 163-172, 2004.

[9] R. Dehkharghani, B. Yanikoglu, Y. Saygin, and K. Oflazer, "Sentiment analysis in Turkish: Towards a complete framework," *Natural Language Engineering*, vol. 1, no. 1, 2015.

[10] F. Wu and Y. Huang, "Sentiment domain adaptation with multiple sources," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 301-310, 2016.

[11] S. Makinist, I. R. Hallac, B. A. Karakus, and G. Aydin, "Preparation of improved Turkish dataset for sentiment analysis in social media," in *Proc. CMES2017*, arXiv preprint arXiv:1801.09975.

[12] N. Godbole, M. Srinivasaiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs," in *Proc. of the International Conference on Weblogs and Social Media (ICWSM)*, vol. 7, no. 21, pp. 219-222, 2007.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", in *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, arXiv preprint arXiv:1301.3781, Jan. 2013.

[14] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, "Combining lexicon-based and learning-based methods for Twitter sentiment analysis," HP Laboratories, Tech. Rep. HPL-2011-89, Jun. 2011.

[15] M. S. Hajmohammadi, R. Ibrahim, and Z. A. Othman, "Opinion mining and sentiment analysis: a survey," *International Journal of Computers & Technology*, vol. 2, no. 3, pp.171-178, Jun. 2012.

[16] A. Esuli, and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proc. of the 5th Conference on Language Resources and Evaluation (LREC'06)*, 2006, pp. 417-422.

[17] E. Cambria, S. Poria, R. Bajpai, and B. W. Schuller, "Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives," in *Proc. of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016)*, Osaka, Japan, 2016, pp. 2666-2677.

[18] A. A. Akin and M. D. Akin, "Zemberek, an open source nlp framework for Turkic languages," *Structure*, vol. 10, pp. 1-5, 2007.