




## Extending a sentiment lexicon with synonym–antonym datasets: SWNetTR++

Fatih SAĞLAM<sup>1,\*</sup>, Burkay GENÇ<sup>2</sup>, Hayri SEVER<sup>3</sup>

<sup>1</sup>Department of Computer Engineering, Hacettepe University, Ankara, Turkey

<sup>2</sup>Policy and Strategy Studies, Hacettepe University, Ankara, Turkey

<sup>3</sup>Department of Computer Engineering, Çankaya University, Ankara, Turkey

Received: 18.09.2018

Accepted/Published Online: 03.02.2019

Final Version: 15.05.2019

**Abstract:** In our previous studies on developing a general-purpose Turkish sentiment lexicon, we constructed SWNetTR-PLUS, a sentiment lexicon of 37K words. In this paper, we show how to use Turkish synonym and antonym word pairs to extend SWNetTR-PLUS by almost 33% to obtain SWNetTR++, a Turkish sentiment lexicon of 49K words. The extension was done by transferring the problem into the graph domain, where nodes are words, and edges are synonym–antonym relations between words, and propagating the existing tone and polarity scores to the newly added words using an algorithm we have developed. We tested the existing and new lexicons using a manually labeled Turkish news media corpus of 500 news texts. The results show that our method yielded a significantly more accurate lexicon than SWNetTR-PLUS, resulting in an accuracy increase from 72.2% to 80.4%. At this level, we have now maximized the accuracy rates of translation-based sentiment analysis approaches, which first translate a Turkish text to English and then do the analysis using English sentiment lexicons.

**Key words:** Turkish sentiment lexicon, sentiment analysis, sentiment lexicon, graph model, GDELT, SWNetTR++

### 1. Introduction

The overall discipline of analyzing textual data to understand and evaluate it is called natural language processing (NLP). Sentiment Analysis (SA) is a subdiscipline under NLP that is focused on assigning a sentiment score to a piece of text [1–3]. There are mainly two approaches in SA [4]: the first is lexicon-based and uses the sentiment scores of words and phrases within the text [5], whereas the second is based on machine learning techniques [6, 7].

With the lexicon-based approach, obviously the most important thing is the existence of a sentiment lexicon. A lexicon here is basically a dictionary matching each word or phrase in the language with a sentiment score. For languages like English, Spanish, and German, many studies were conducted for the production of comprehensive lexicons [8–10]. However, languages like Turkish have been missing proper lexicons.

Well-known sentiment lexicons for English, such as SentiWordNet [8] and SenticNet [11], are general-purpose and domain-independent. However, they cannot capture sentiment variations across different domains or cultures. For example, the term “big” has a positive meaning when used in reference to room size in the hotel domain, and a negative one when referring to battery size in the camera domain [12]. According to Dehkharghani et al., the solution to this is domain-dependent and language-dependent (or culture-dependent) lexicons. There are studies where general-purpose lexicons are adapted to specific domains based on data from

\*Correspondence: fatihsaglam@hacettepe.edu.tr

that domain [13–15]. For this purpose, the existence of a general-purpose Turkish sentiment lexicon makes it viable to repeat similar studies in Turkish.

Rather than starting with a set of seed words, we generated our initial lexicon in a deductive manner, starting with a large labeled corpus of news texts obtained from the GDELT (<https://www.gdeltproject.org>) database. Using this corpus, we constructed the SWNetTR-GDELT lexicon applying the methodology summarized in Section 3.2.2. We then merged this lexicon with the existing SWNetTR lexicon by Ucan [16] and obtained a lexicon with a size of 37K words [17].

However, the lexicon we produced was still immature compared to resources in major languages. Therefore, in this paper, we aim to introduce a novel approach to extend SWNetTR-PLUS and significantly increase its accuracy. Our approach can be summarized in three steps: We first constructed a large database of synonym and antonym word pairs from the existing lexicon and publicly available Turkish language sources. We then modeled this database as a graph, and finally we propagated tone and polarity values from existing words to newly added words. Upon testing this extended 49K word lexicon using manually labeled test corpus, we obtained significant accuracy increases.

This paper provides details of our lexicon construction methodology and is organized as follows: In Section 2, we outline the existing sentiment analysis and polarity detection studies. The data and methodology we use is discussed in Section 3. The developed lexicon is evaluated and compared to existing lexicons in Section 4. Lastly, the results and future work is outlined in Section 5.

## 2. Related work

Classical sentiment lexicon generation studies can be grouped under three approaches: manual construction, dictionary-based, and corpus-based [18].

The first approach requires massive human effort. Abdul-Mageed et al. [19] manually constructed a lexicon of 3982 adjectives from a manually annotated news text corpus for the Arabic language which is similar to Turkish in its morphological richness. An and Kim [20] applied crowdsourcing, folksonomy, and voting to derive a sentiment lexicon from a 517K-word Korean dictionary. They were helped by over 35,000 college students.

The second approach is based on extending a manually constructed set of seed words using a dictionary (thesaurus, WordNet, etc.). Zaško-Zielińska et al. [21] used Polish WordNet (plWordNet) to construct plWordNet-emo which is a Polish sentiment lexicon. Although their approach is dictionary-based, it includes heavy manual effort. Around 30,000 lexical units in plWordNet was manually annotated by 6 annotators. Kamps et al. [22] proposed a more sophisticated approach that uses a WordNet distance-based method to determine the sentiment orientation of a given adjective. The distance between words is the length of the shortest path in WordNet. Hassan and Radev [23] presented a Markov random walk model over a word relatedness graph to produce a sentiment estimate for a given word. It uses WordNet synonyms and hypernyms to build a word relatedness graph.

Finally, document-based approach usually starts with a set of manually labeled seed words and expands it using available corpora. Kaji and Kitsuregawa [24] proposed a corpus-based model which uses polar sentences extracted from the corpus to construct a sentiment lexicon for the Japanese language. They considered one billion web pages to extract 500K polar sentences, which were then combined with manually determined "cue-words". Deng et al. [25] proposed a method to adapt existing sentiment lexicons for domain-specific sentiment classification using an unannotated corpus and a dictionary. This method utilizes three language resources: a

developing corpus, a seed lexicon, and a dictionary. They applied this method to the domain consisting of tweets related to the stock market. The evaluation results showed that the expanded lexicons improved the sentiment classification performance significantly compared to the seed lexicons. Moreover, there exist more sophisticated approaches benefiting from labeled data such as a corpus of reviews with known polarities [6, 26–28].

In the literature, studies directly focusing on developing a Turkish sentiment lexicon are scarce. To provide a background as rich as possible, here we try to summarize all existing studies. Vural et al. [29] used the SentiStrength [30] tool to produce a framework for studying sentiment analysis on movie reviews. The lexicon of SentiStrength is constructed by manual translation to Turkish by editors. However, there is no information about the capacity and content of the lexicon. Türkmenoğlu and Tantığ [31] studied whether a lexicon-based or machine-learning-based approach is more successful for Turkish language. Similar to Vural et al. [29], they used SentiStrength and manually translated words into Turkish, obtaining 2547 words. Considering negations and manually adding more words, they extended the lexicon to 3657 words. Ucan [16], as part of his thesis, produced the Turkish Sentiment Dictionary via translation of language resources from English into Turkish. This lexicon has 27K Turkish words and phrases with assigned binary polarity scores. Aytekin [32] used a machine learning approach to determine the sentiment polarity of texts on Turkish Blog pages related to production and service sectors. He produced a 4744-word lexicon via translation from English into Turkish. Kaya and Conley [33] compared two lexicons, the first one has a size of 6800 words and obtained by translation from English, whereas the other is of size 536 words and constructed using an ad hoc approach from a Twitter dataset. Besides translation-based approaches, seed word list use is also quite frequently studied. In most studies, the seed word list is generated manually. Ekinci and Omurca [34] recently mentioned the importance of working with seed word lists and proposed a novel approach. They started with 62 pure polarity words and extended the seed list using the Beam search algorithm to 189 words via synonyms and antonyms.

SentiTurkNet, developed by Dehkharghani et al. [12], is the most developed Turkish sentiment lexicon in the literature. They followed a similar methodology and started with a seed set, extending it by 15,000 synsets in the Turkish WordNet. They assigned polarity strength values based on the confidence scores of a classifier algorithm. Although this work is extensible to other languages containing corresponding WordNets, this is also a limitation on its size. The Turkish WordNet is not in public domain, and its size of 15,000 synsets is well below the English SentiWordNet which contains 117,000 synsets. Moreover, a word may exist in different synsets with varying polarities in SentiTurkNet. The purpose of this is to figure out the synset to be used based on the context, using the Gloss field of the WordNet. However, Gizem and Yanikoglu [35] mentioned that SentiTurkNet contains words without the Gloss field, and due to the difficulty of determining which synset to use, it was preferred to use the average polarity of all possible cases. The main advantage of our lexicon over SentiTurkNet is that we do not base it on the Turkish WordNet; hence, we are not limited by its size and structure.

Works in the literature show that there does not exist a complete and comprehensive Turkish sentiment lexicon. Most studies are based on translation-dependent lexicons or have relatively narrow focuses. In this study, we propose a novel, fully automated methodology to obtain a general-purpose Turkish sentiment lexicon with competitively accurate polarity and tone scores.

### 3. Data and methodology

#### 3.1. Data

We have summarized the datasets used and produced through out this study in Figure 1. Brief information related to these datasets is provided below; more details and methodology presented in Section 3.2:

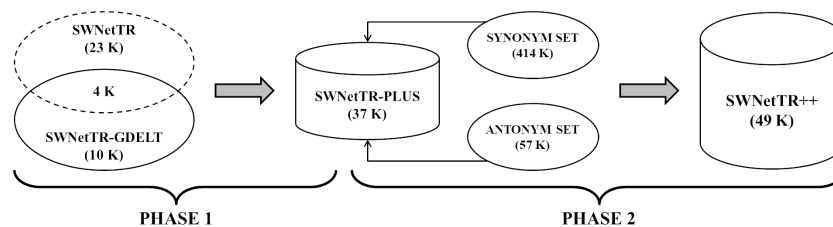


Figure 1. The datasets used and produced.

**SWNetTR** Developed by Ucan [16] within the context of his master of science thesis studies. It has a total of 27K words. It is based on the translation of SentiWordNet using different translation tools. We refer to this dataset in our study as SWNetTR.

**SWNetTR-GDELT** This dataset has been produced by our team using the GDELT (Global Database of Events, Language and Tone) Project (<https://www.gdeltproject.org/>) datasets, and contains about 14K words. The details of development are presented in Section 3.2.2.

**SWNetTR-PLUS** This is a union of SWNetTR and SWNetTR-GDELT lexicons, containing a total of 37K words. This work was presented in [17], but has some flaws such as missing negations and the overall size of the lexicon.

**SYNONYM and ANTONYM datasets** These two datasets have been collected by our team from different synonym and antonym sources on the net to extend SWNetTR-PLUS. The details of our approach are presented in Section 3.2.3.

**SWNetTR++** This is the final lexicon we have obtained as a result of our studies shown as Phase 2 in Figure 1. It is based on SWNetTR-PLUS and extensions through synonym and antonym datasets using a graph-based approach. Details of the methodology used are given in Section 3.2.

**MLTC (Manually labeled Turkish corpora)** There are very few annotated corpora in Turkish, and the existing ones are domain-specific (hotel, tourism, movie review, etc.) [36, 37]. MLTC is constructed manually by 3 native Turkish speakers to test the performance of SWNetTR++ in this study. It contains manually labeled, 500 Turkish news texts.

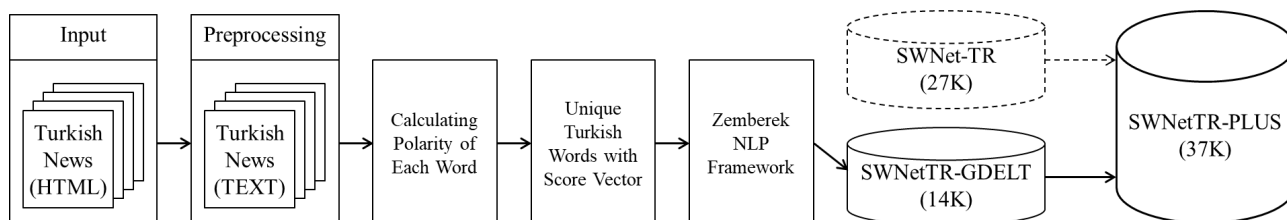
### 3.2. Methodology

#### 3.2.1. Polarity and tone concepts

In this work we consider two measures for the sentiment value of a document or word: the first one is polarity and the second one is tone. The polarity is either +1, representing a positive sentiment, or -1, representing a negative sentiment. The tone, on the other hand, can change continuously between these two extremes. Hence, the tone provides more information about the strength of the sentiment, whereas the polarity provides a sharp summary. Throughout this study, our main aim will be to compute the most accurate tone and polarity values for the words in the Turkish language.

### 3.2.2. Developing SWNetTR-PLUS lexicon with GDELT dataset

In our previous work presented in [17] and outlined in Figure 1 as Phase 1 with further details provided in Figure 2, we developed SWNetTR-PLUS, mostly relying on GDELT Project's datasets. Here, we briefly summarize our old studies to pave the way for the current efforts.



**Figure 2.** Main steps of SWNetTR-PLUS development process.

GDELT is a large database of world news media, providing open access to metadata on news texts from the world to researchers. Among the metadata they provide, one can find the type of the event, actors acting on the event, location of the event, theme of the event, and tone and polarity of the text. GDELT applies content analysis on each news text using 18 different advanced language tools (<https://blog.gdeltproject.org/introducing-the-global-content-analysis-measures-gcam/>), SentiWordNet [8] and WordNet Affect [38] being among them. Whereas the polarity provided by GDELT is a binary value representing either positive (+1) or negative (-1) sentiment value, tone is a real number between +10 and -10. Therefore, the tone value not only hints at the polarity, but also provides a relative sense of the strength of that polarity. Their polarity and tone assignment is based on a translation approach. They first translate the Turkish text to English and then run their sentiment analysis algorithms in English.

We have used GDELT's knowledge graph dataset to acquire a hundred thousand news articles from their sources, half of them were labeled as having positive polarity and the other half as having negative polarity by GDELT. We preprocessed all texts to strip HTML tags, punctuation marks, numerical expressions, and stop words off the body of the text. We then used Zemberek (<http://code.google.com/p/zemberek>), the Turkish NLP engine, to obtain stems of words. At this step, we obtained 11 million words, of which 14,023 were unique. We constructed word vectors to compute polarity and tone metrics for each word, finally resulting in SWNetTR-GDELT, a lexicon with the size of 14K words. Then, we merged SWNetTR and SWNetTR-GDELT to obtain a lexicon with the size of around 37K words, which we named SWNetTR-PLUS.

We evaluated both SWNetTR and SWNetTR-PLUS on the MLTC dataset, and realized that whereas the accuracy of SWNetTR is 60.6%, SWNetTR-PLUS achieves 72.2%. The still low accuracy of SWNetTR-PLUS led us to deepen our research and extend the lexicon even further in this study.

### 3.2.3. SYNONYM and ANTONYM datasets

To further improve the SWNetTR-PLUS lexicon, we turned to a novel approach of extending it via synonyms and antonyms of the existing words. To this end, we used four different web sources, "www.dilbilgisi.net", "www.es-anlam.com", "www.supersozluk.com", and YTU Kemik Dil Grubu, to obtain possible synonyms and antonyms of the existing words in our lexicon. Each resource provided a different amount of synonyms and antonyms due to the way their databases were constructed (manual vs automatic). All merged together, we had 414,275 unique pairs of synonyms and 57,148 unique pairs of antonyms. Whereas sources such as www.dilbilgisi.net

provide manually curated words, others such as [www.supersozluk.com](http://www.supersozluk.com) provide automatically generated word lists. This posed two important challenges that we will explain in detail in the following sections: irrational synonym–antonym relations and missing polarity scores.

Once the synonym and antonym pairs from different sources were merged, we realized that we had a number of irrational relations between word pairs. Let us give a simple example to explain this problem. Consider three words A, B, and C. Assume that A and B are synonyms, also A and C. Naturally, we expect B and C to be synonyms as well. However, there may exist an antonym relation in the database between B and C. This situation creates a paradoxical state, and hinders our algorithms which we will later describe. One may assume that this rarely occurs and may be resolved manually. However, due to the nature of construction of some of our sources, this situation is observed at more than an acceptable level and requires an automated solution.

Moreover, the newly discovered words needed to have a polarity score before we could add them to the lexicon. Unfortunately, these words do not exist in our corpus. Hence, we had to find a way to assign them polarities.

To solve both problems, we decided to construct a graph model based on the synonym and antonym pairs. This way, we would be able to visualize and analyze the polarity scores and synonym-antonym relations easily through a single interface. Moreover, manipulating the graph would be much more easier than working on independent pairs of words. Therefore, we created the lexicon-graph based on the synonym–antonym relations we collected from the web sources. As a result, we obtained a large graph with 79,326 nodes and 471,258 undirected edges. Note that each edge is labeled as either a synonym or an antonym edge, but not both. Nodes are labeled with their polarity scores, if one exists.

The resulting undirected graph had 22,087 connected components and the largest connected component contained a massive number of 48,861 words. The reason for the largest component being so massive is having many weak connections between words. These weak connections are mostly created by the automated algorithms generating the sources we use and they are also the main source of the irrational relations between words. Therefore, we needed a way to distinguish strong relations from weak ones so that we can eliminate the really weak edges between nodes, terminating irrational relations and dismantling the largest component as much as possible.

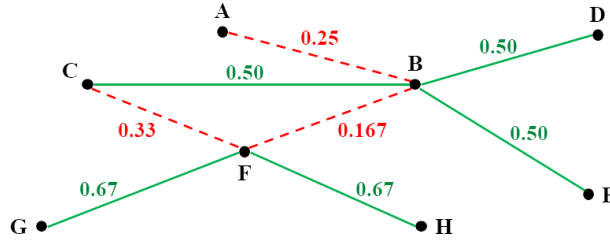
### 3.2.4. Tie strength (TS)

To be able to distinguish between strong and weak pairs of words, we decided to develop a metric to measure the strength of an edge between two words. This metric, which we call tie strength, measures an edge and assigns a value between 0 and 1 to it, where 1 represents the highest possible connection strength.

Let us assume that A and B are two words connected by an edge in the lexicon graph. If the edge is a synonym edge, then the tie strength between A and B is computed as in Eq. (1), else, as shown in Eq. (2). Here,  $A^{Syn}$  represents the set of all adjacent nodes of A, including A itself, connected to A with a synonym edge, whereas  $A^{Ant}$  is the set of all adjacent nodes of A, connected to A with an antonym edge.  $B^{Syn}$  and  $B^{Ant}$  are defined similarly.  $TS_A^{Syn}$  represents the strength of the synonym tie between nodes A and B from node A's side, whereas  $TS_B^{Syn}$  represents the strength of the same tie from B's side. Note that these two are not necessarily equal. Finally,  $TS_{A-B}^{Syn}$  is the computed synonym tie strength for the edge A-B.  $TS_A^{Ant}$ ,  $TS_B^{Ant}$ , and  $TS_{A-B}^{Ant}$  are defined similarly for antonym edges.

$$TS_A^{Syn} = \frac{|A^{Syn} \cap B^{Syn}|}{|A^{Syn}|} \quad TS_B^{Syn} = \frac{|A^{Syn} \cap B^{Syn}|}{|B^{Syn}|} \quad TS_{A-B}^{Syn} = TS_A^{Syn} \times TS_B^{Syn} \quad (1)$$

$$TS_A^{Ant} = \frac{|A^{Syn} \cap B^{Ant}|}{|A^{Syn}|} \quad TS_B^{Ant} = \frac{|A^{Ant} \cap B^{Syn}|}{|B^{Syn}|} \quad TS_{A-B}^{Ant} = TS_A^{Ant} \times TS_B^{Ant} \quad (2)$$



**Figure 3.** An example of the word relation graph with Tie Strength.

Figure 3 shows a synthetic example of the word relation graph with tie strength attributes. Consider the B-C edge. Since this edge is a synonym edge, we need to apply Eq. (1) to compute its tie strength. We first need to compute the synonym and antonym sets of both B and C:  $B^{Syn} = \{B, C, D, E\}$ ,  $B^{Ant} = \{A, F\}$ ,  $C^{Syn} = \{B, C\}$ , and  $C^{Ant} = \{F\}$ . Next, we compute  $TS_B^{Syn}$  which is equal to 0.5, and  $TS_C^{Syn}$  which is equal to 1.0. Hence,  $TS_{B-C}^{Syn}$  becomes 0.5.

One can see from Eq. (1) that to maximize  $TS_A^{Syn}$  when it becomes 1, we need to have  $A^{Syn} \subseteq B^{Syn}$ . Similarly, to maximize  $TS_B^{Syn}$ , we need to have  $B^{Syn} \subseteq A^{Syn}$ . Hence, to maximize  $TS_{A-B}^{Syn}$ , we need to have both  $A^{Syn} \subseteq B^{Syn}$  and  $B^{Syn} \subseteq A^{Syn}$ , which leads to  $A^{Syn} = B^{Syn}$ . In other words, as the overlap between the synonym sets of A and B increase, the value of the tie strength metric also increases. The situation for the antonym case is more complicated. To have a high antonym tie strength between two words, we require the synonyms of one word to be elements of the antonym set of the other word.

### 3.2.5. Pruning weak edges and filtering outlier nodes

Once the tie strength of each edge is computed, we prune synonym edges below a certain tie strength threshold. This allows us to simplify the graph model and avoid irrational relations. Through enumeration and expert opinion, we have determined that a tie strength value of 0.2 provides the optimum pruning without destroying useful ties. As an example, we provide Figure 4, which shows the synonym neighborhood of the word “abartı”. The direct translation of “abartı” to English is “exaggeration”. Here, we apply a 0.2 threshold and draw edges below this threshold with thin dashed lines, whereas edges above this threshold are drawn as heavy solid lines. First of all, we observe that most of the dashed edges, indeed, connect to words with relatively weak synonymous meaning. For example, “yalan” which means “lie” has a tie strength of 0.0093, “fanatiklik” which means “fanaticism” has a tie strength of 0.1948, and “saplantı” which means “obsession” has a tie strength of 0.0635. On the other hand, “abartma” which means “to exaggerate” has a tie strength of 0.2037, “mübalaga” which is the Arabic word for “abartı” has a tie strength of 0.4074, “izam etme” which can be translated as “to exaggerate” has a tie strength of 0.2593. Overall, we can say that our metric quite reasonably distinguishes between strongly and weakly associated words.

We also observe that wrong uses of a word are also properly refined by this approach. For example,



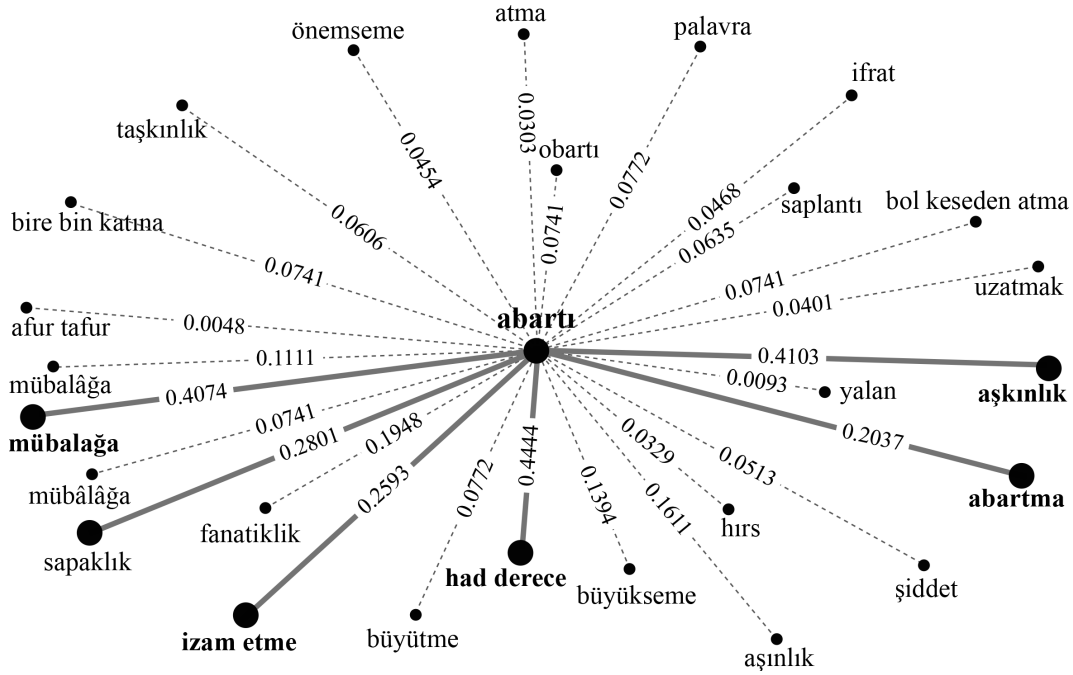


Figure 4. Synonyms subgraph of the word "abartı".

"mübalâğa", "mübâlâğa", and "mübalâğa" all represent the same word, where "mübalâğa" is the correct spelling. We can see that the wrong spellings are eliminated since they are below the threshold.

Moreover, by eliminating weak edges, we got rid of all irrational relations since these were due to the weak synonymous relations. By eliminating these weak relations, we destroyed all irrational cycles within the graph.

Note that we do not prune the antonym edges. There are mainly two reasons for that. First, we have much less antonym edges than synonym edges in the graph; second, the tie strength of antonym edges seem to be smaller than the synonym edges due to the way they are computed. Pruning antonym edges would mean removing most of them from the graph and significantly affect the results. In Figure 5, we show the final neighborhood of word "abartı". In this figure, the triangles denote the antonyms and the circles denote the synonyms of the word after pruning the weak edges.

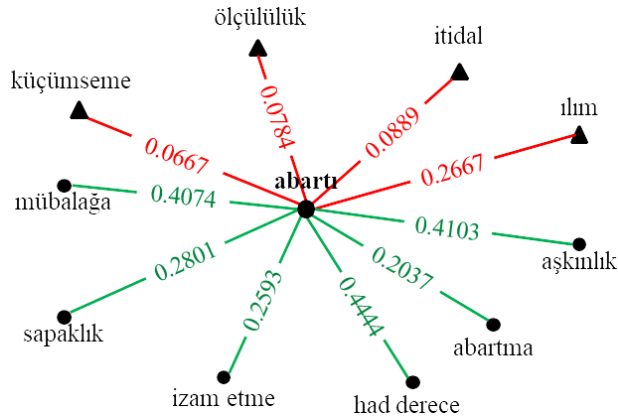


Figure 5. Subgraph of "abartı" word.



Once pruning is completed, we have some connected components of one or more nodes in the graph, which do not contain any words with assigned tone values. Since there is no means of assigning a tone score to these components, we called them outliers and eliminated them from the graph model, leaving only components that contains at least one word with an assigned tone score.

After filtering the useless components, there were 63,979 nodes and 210,024 edges left in the graph model. There were also 27,677 connected components and the size of the giant component was 19,118. The number of the connected components increased and the size of the giant component decreased, as expected. At this step, the graph model is stabilized, and we can pass to the next step where we propagate the tone information through the graph.

### 3.2.6. Tone propagation

Let us first remind that all articles in the GDELT library are assigned a tone score and a polarity score. Whereas the polarity is either  $-1$  for negative or  $+1$  for positive, tone is a real value between  $-10$  and  $+10$ . We scaled the GDELT tone scores down to the  $[-1, +1]$  range in our study to make them more consistent with the polarity values. From GDELT's tone and polarity scores, we have developed our first lexicon SWNetTR-PLUS, assigning each word in the lexicon both a polarity and a tone score. Therefore, all the words in the graph model, which come from SWNetTR-PLUS, already have both scores. However, we have many new words added to the graph and all of them are missing tone and polarity scores; hence, we next try to compute a tone score for each such word via "tone propagation".

To be able to compute the tone and polarity scores of newly added words without the need to have matching texts in the learning corpus, we decided to propagate the tone scores of each node to its neighbors at certain proportions. This way, each node with a tone score affects its neighbors' tone scores. Obviously, a single round of update does not ensure that all nodes in the graph gets a tone score. Therefore, we need to repeat this procedure a certain number of times until all nodes in the graph receive a tone score and each further update does not change these scores in a significant amount.

The following formula describes the update procedure of each node in a formal way:

$$T'_u = \alpha \times T_u + \beta \times \sum_{v \in N_u} \left[ S_{u-v} \frac{T_v \cdot TS_{u-v}}{\sum_{v \in N_u} TS_{u-v}} \right]. \quad (3)$$

In Eq. (3), there are two components: the first component with coefficient  $\alpha$  represents the current tone score of word  $u$ , the second component with coefficient  $\beta$  represents the contribution of  $u$ 's neighbors to  $u$ 's tone score. The first component is trivial as  $T_u$  represents the current tone of word  $u$ . In the second component,  $S_{u-v}$  is the synonym-antonym coefficient and given as:

$$S_{(u,v)} = \begin{cases} +1 & \text{if } u \text{ is a synonym of } v \\ -1 & \text{if } u \text{ is an antonym of } v, \end{cases}$$

whereas  $N_u$  represents the set of neighbors of  $u$  which has an assigned tone score, and  $T_v$  is the tone of neighbor node  $v$ .

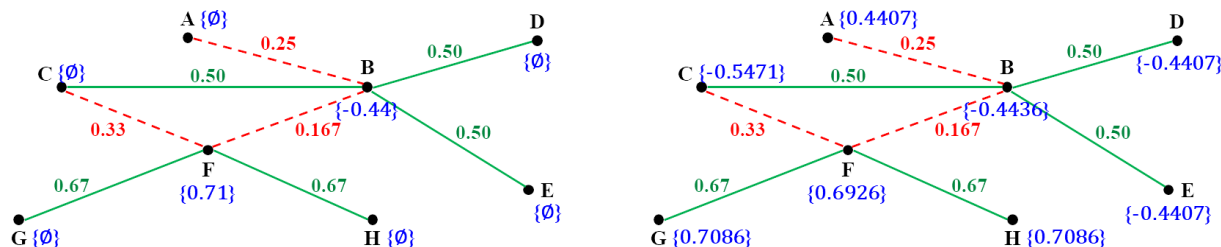
In this update procedure, we can face four different scenarios, which are summarized in Table 1. In the first scenario, neither  $u$  nor any of its neighbors have any tone values assigned, i.e.  $N_u$  is an empty set. In this case, there is no information we can use to set the tone of  $u$ . Hence, we set both *alpha* and *beta* as 0, which

leads to  $T_u$  becoming 0 as well. In the second scenario,  $u$  does not have a current tone score, but it has at least one such neighbor. Then,  $beta$  becomes 1, which means the tone score of  $u$  is completely determined by the weighted average of its neighbors' tone scores. Third case is when  $u$  has a tone score, but none of its neighbors do. In this case, we set  $alpha$  as 1 to keep the current tone score of  $u$  intact. The final case is when both  $u$  and some or all of its neighbors have assigned tone scores. In this case, we set  $alpha$  to 0.95 and  $beta$  to 0.05. The choices of the values of  $alpha$  and  $beta$  here are subjective. A higher  $alpha$  results in each word sticking to its first assigned tone score more firmly, whereas a higher  $beta$  allows faster termination and a smoother tone distribution among words. Actually, increasing  $beta$  above 0.10 results in a fast convergence of all tone scores to an average fixed value throughout the whole graph.

**Table 1.** Possible scenarios of node  $u$  and its neighbors.

Cases	$u$ has tone	$N_u$ is empty	Coefficients
Case 1	False	False	$\alpha = 0 \quad \beta = 0$
Case 2	False	True	$\alpha = 0 \quad \beta = 1$
Case 3	True	False	$\alpha = 1 \quad \beta = 0$
Case 4	True	True	$\alpha = 0.95 \quad \beta = 0.05$

Figure 6 provides an example where the graph on the left turns into the graph on the right after three iterations. In this example, solid green edges denote synonyms (C-B, B-D, B-E, F-G, F-H) and dashed red edges denote antonyms (A-B, C-F, F-B). Initially, only B and F have tone scores. Three iterations later, all words in the component have their tone scores updated.



**Figure 6.** An example of tone propagation in a connected component.

At each iteration, the graph evolves and the tone scores of words change. However, if we do very few iterations, the propagated information may fail to reach some words. Moreover, the tone score of many words would then fail to reach a stable value. On the other hand, if we do too many iterations, then tone scores of all words start converging to an average of the whole graph. In our tests, we found that a good approximation for an ideal number of iterations per component is given by the diameter of that component. This ensures that each word is updated at least once and does not cause a significant convergence problem.

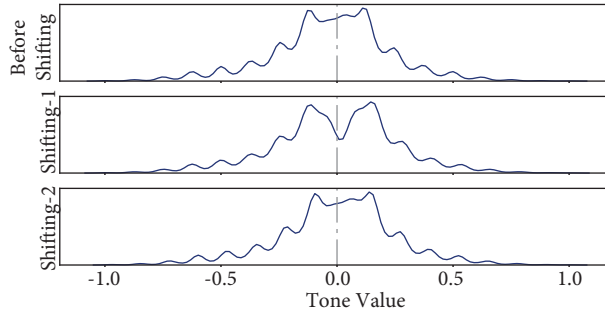
### 3.2.7. Negative bias problem and tone shifting

At this stage, we left 49,241 nodes in the graph, each corresponding to a word and its assigned tone score. We, then, directly determined the polarity from the sign of the tone score. Table 2 gives a summary of the tone score distribution. Although, the number of positive and negative polarities within the model is relatively balanced, the mean tone values for negative and positive polarities differ significantly. This means that the

**Table 2.** Polarity and tone summaries before tone shifting.

	Positive	Negative	Total
#Words	23,603	25,638	49,241
Rate	48%	52%	
Mean of tone values	0.17	-0.21	

negative words in the lexicon are stronger than the positive words. We think that this is due to the content and the language used to present this content in the news. Unfortunately, this leads to an unexpected problem: we tend to classify texts with a negative bias.

**Figure 7.** Tone shifting of lexicon.

To be able to fix this, we decided to shift the tone scores of all words to create an equilibrium. We tested two alternative approaches, both of which we name as tone shifting:

- **Tone Shifting - 1:** In this approach, we incremented the tone scores of all positive words such that, at the end, the absolute values of the mean tone score for positive and negative words became equal. In this approach, the tone scores of negative words did not change. Also, there was no change in word polarities.
- **Tone Shifting - 2:** In the second approach, we incremented the tone scores of all words such that, at the end, the mean tone score for the whole lexicon became 0. In this approach, all words' tone scores were affected and some barely negative words became positive words.

Figure 7 shows the density plots of tone distributions before and after the shifts. The effects of both shifts are clearly visible. In the first type of shifting, we created a gap to the right side of 0.0 by moving everything right of 0.0 further to the right, creating a change in the shape of the density plot. In the second type of shift, we moved everything equally to the right, preserving the shape of the distribution but changing polarities of some words.

As a result, we obtained two different lexicons via these approaches and tested both of them on the MLTC dataset. We talk about all the evaluation results in the next section.

#### 4. Experimental evaluation

In Section 3.1, we have mentioned the MLTC dataset that contains 500 news texts manually labeled as positive or negative by 3 native Turkish speakers. From now on, we will refer to this dataset as MLTC-500. On 353 of these 500 texts, all three evaluators agreed on the polarity of the text. On the remaining 147 texts, one of the

evaluators disagreed with the others. We constructed an alternative corpus of these 353 texts and called them MLTC-353. We conducted our tests on both MLTC-500 and MLTC-353. We conducted two types of tests:

- Polarity-based: In this type of test, we used the assigned polarities of words to compute the polarity of the text in the corpus.
- Tone-based: In this type of test, we used the assigned tone scores of words to compute the polarity of the text in the corpus.

Table 3 outlines the results for the MLTC-500 and MLTC-353. From the MLTC-500 results on Table 3, we can see that the default GDELTA polarity assignments have a 78.8% match rate with the manual labeling. We remind that GDELTA uses a translation-based sentiment analysis approach. They first translate the Turkish text into English, and then do the sentiment analysis on the English text using an English lexicon. Our aim is to provide means of doing sentiment analysis directly on Turkish texts with the same or better success ratios.

**Table 3.** Performance testing results on MLTC-500 and MLTC-353.

DATASET		MLTC-500		MLTC-353	
		Polarity-based	Tone-based	Polarity-based	Tone-based
GDELTA		78.8%		84.99%	
SWNetTR		60.6%	64.2%	62.89%	67.99%
SWNetTR-PLUS		72.2%	75.2%	75.07%	78.75%
SWNetTR++	Before tone shifting	75.0%	73.8%	78.75%	77.05%
SWNetTR++	<b>Tone shifting - 1</b>	75.0%	<b>80.4%</b>	78.75%	<b>84.42%</b>
SWNetTR++	Tone shifting - 2	73.6%	77.8%	79.32%	83.85%

SWNetTR lexicon can achieve a 60.6% accuracy for polarity-based testing and 64.2% accuracy for tone-based testing. Both values are well below the GDELTA scores and demonstrate the poor state of the lexicon before our studies. SWNetTR-PLUS, which was our previous contribution, achieved 72.2% and 75.2% accuracy rates. These are significant improvements; however, they are still below the GDELTA ratios. Our contributions in this paper is incrementing the ratios to 75.0% and 80.4% when we apply the first type of tone shifting to the lexicon. Although improved, polarity-based sentiment analysis scores are below GDELTA's benchmarks, whereas tone-based scores are now beyond the GDELTA levels. This shows that although there are still opportunities for improvement, we have reached a major checkpoint in producing a lexicon for Turkish sentiment analysis in par with a comparable translation-based approach.

When we want to have a deeper look into what is incorrectly assigned a polarity, we observe discrepancies at both word and document levels. For example, "acması" ("pathetic" in English) is almost always a word with negative polarity. However, in the lexicon, it is assigned a positive tone score. However, we have very rare such occurrence of inaccuracies at the word level. At the document level, one example is a news text where an attack at a statue in the Kars province is presented. The first part of the text contains a short explanation of the event, followed by a lengthy praise of the sculptor. Although the human evaluators labeled this text as negative due to its main theme of "attack on art", the automated process labeled it as positive due to the highly positive sentiments provided about the sculptor. These issues are currently heavily studied under aspect-level sentiment analysis, and we aim to include them in our future studies.

**Table 4.** Development process of Turkish sentiment lexicons.

Phase	Lexicon Name	Size	Performance	Test Type
0	SWNetTR	27K	64.2%	tone-based
1	SWNetTR-PLUS	37K	75.2%	tone-based
2	SWNetTR++	49K	80.4% 84.42%	MLTC-500, tone-based MLTC-353, tone-based

When the MLTC-353 results in Table 3 are analyzed, once again, our contributions in this paper are significant with some room for further improvements. It is clear that automated sentiment analysis works much better whenever humans agree on the sentiment. Almost all lexicons gain about 4%–6% success ratios when tested on texts with more obvious sentiments.

Overall, our results are summarized in Table 4. We have now increased the size of the lexicon to about 49K words and the success ratio to 80.4% on MLTC-500 and 84.42% on MLTC-353.

## 5. Conclusion and future works

In this paper, we have focused on extending the general-purpose Turkish sentiment lexicon called SWNetTR-PLUS, and to achieve a successful analysis rate of at least at par with translation-based approaches. Before this study, the best lexicon known to us was SWNetTR-PLUS, which is based on SWNetTR. We have, now, extended SWNetTR-PLUS to SWNetTR++ by incrementing the word count to 49K from 37K. This new lexicon achieves much higher success rates compared to SWNetTR and SWNetTR-PLUS on MLTC-500, a corpus manually created by us for this study.

The novel sides of our approach was to use synonym–antonym datasets to increase the size of the lexicon and the application of a tone propagation algorithm to compute the estimated tone scores of newly added words to the lexicon.

Although there are significant improvements in performance scores with this new lexicon, we determine also a number of potential opportunities for further improvement:

- The lexicon can be further extended with direct negations, such as “sevmek” vs. “sevmemek”. Right now, we do not automatically produce negation pairs and the lexicon is missing these potential additions.
- The complex agglutinative word structure of Turkish makes sentiment analysis quite challenging. By applying a smart dissection-extension algorithm to each existing word, we can multiply the size of the lexicon.
- We believe analysis of n-grams may prove to be very valuable in determining the sentiment score of a document.
- Further testing with larger manually labeled corpora may reveal more opportunities for improvement. Unfortunately, such corpora is scarce in Turkish language.

We think that we have developed and outlined a novel approach to lexicon construction which is language-independent. We hope to refine our approach further and to see if it applies to lexicons in other languages.

We will make the entire lexicon publicly available in the near future and it will be available for download at (<https://github.com/swnettr>).

## References

- [1] Ronen F. Techniques and applications for sentiment analysis. *Communications of the ACM* 2013; 56 (4): 82-89.
- [2] Taboada M, Brooke J, Tofloski M, Voll K, Stede M. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 2011; 37 (2): 267-307.
- [3] Liu B. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 2012; 5 (1): 1-167.
- [4] Thelwall M, Buckley K, Paltoglou G, Cai D, Kappas A. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 2010; 61 (12): 2544-2558.
- [5] Turney PD. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: *ACL-02 40th Annual Meeting on Association for Computational Linguistics Conference*; Philadelphia, PA, USA; 2002. pp. 417-424.
- [6] Pang B, Lee L, Vaithyanathan S. Thumbs up?: Sentiment classification using machine learning techniques. In: *ACL-02 Empirical Methods in Natural Language Processing Conference*; Pennsylvania, PA, USA; 2002. pp. 79-86.
- [7] Nasser A, Dinçer K, Sever H. Investigation of the feature selection problem for sentiment analysis in Arabic language. *Research in Computing Science* 2016; 110: 41-54.
- [8] Baccianella S, Esuli A, Sebastiani F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: *The Seventh International Conference on Language Resources and Evaluation*; Valletta, Malta; 2010. pp. 2200-2204.
- [9] Perez-Rosas V, Banea C, Mihalcea R. Learning sentiment lexicons in Spanish. In: *The Eight International Conference on Language Resources and Evaluation*; Istanbul, Turkey; 2012. pp. 73-77.
- [10] Clemenatide S, Klenner M. Evaluation and extension of a polarity lexicon for German. In: *The First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*; Lisbon, Portugal; 2010. pp. 7-13.
- [11] Cambria E, Speer R, Havasi C, Hussain A. Senticnet: a publicly available semantic resource for opinion mining. In: *AAAI Fall Symposium: Commonsense Knowledge*; Arlington, VA, USA; 2010. pp. 14-18.
- [12] Dehkharghani R, Saygin Y, Yanikoglu B, Oflazer K. SentiTurkNet: a Turkish polarity lexicon for sentiment analysis. *Language Resources and Evaluation* 2016; 50 (3): 667-685.
- [13] Choi Y, Cardie C. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In: *Empirical Methods in Natural Language Processing Conference*; Singapore; 2009. pp. 590-598.
- [14] Lu Y, Castellanos M, Dayal U, Zhai C. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: *The Twentieth International WWW Conference*; Hyderabad, India; 2011. pp. 347-356.
- [15] Demiroz G, Yanikoglu B, Tapucu D, Saygin Y. Learning domain-specific polarity lexicons. In: *IEEE 12th International Conference on Data Mining Workshops*; Brussels, Belgium; 2012. pp. 674-679.
- [16] Ucan A. Automatic sentiment dictionary translation and using in sentiment analysis. MSc, Hacettepe University, Ankara, Turkey, 2014.
- [17] 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*; Agadir, Morocco; 2016. pp. 1-5.
- [18] Zhao J, Liu K, Xu L. Sentiment analysis: mining opinions, sentiments, and emotions. *Computational Linguistics* 2016; 42 (3): 595-598.
- [19] Abdul-Mageed M, Diab MT, Korayem M. Subjectivity and sentiment analysis of modern standard Arabic. In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies Conference*; Portland, OR, USA; 2011. pp. 587-591.
- [20] An J, Kim HW. Building a Korean sentiment lexicon using collective intelligence. *Journal of Intelligence and Information Systems* 2015; 21 (2): 49-67.

- [21] Zaško-Zielińska M, Piasecki M, Szpakowicz S. A large wordnet-based sentiment lexicon for Polish. In: Recent Advances in Natural Language Processing Conference; Hissar, Bulgaria; 2015. pp. 721-730.
- [22] Kamps J, Marx M, Mokken RJ, De Rijke M. Using WordNet to measure semantic orientations of adjectives. In: 4th International Conference on Language Resources and Evaluation; Lisbon, Portugal; 2004. pp. 1115-1118.
- [23] Hassan A, Radev D. Identifying text polarity using random walks. In: ACL-10 48th Annual Meeting on Association for Computational Linguistics Conference; Uppsala, Sweden; 2010. pp. 395-403.
- [24] Kaji N, Kitsuregawa M. Building lexicon for sentiment analysis from massive collection of HTML documents. In: The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; Prague, Czech Republic; 2007. pp. 1075-1083.
- [25] Deng S, Sinha AP, Zhao H. Adapting sentiment lexicons to domain-specific social media texts. *Decision Support Systems* 2017; 94: 65-76.
- [26] Ye Q, Zhang Z, Law R. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications* 2009; 36 (3): 6527-6535.
- [27] Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R. Sentiment analysis of twitter data. In: Workshop on Language in Social Media; Portland, OR, USA; 2011. pp. 30-38.
- [28] Gezici G, Yanikoglu B, Tapucu D, Saygin Y. New features for sentiment analysis: Do sentences matter? In: 1st International Workshop on Sentiment Discovery from Affective Data; Bristol, UK; 2012. pp. 5-15.
- [29] Vural AG, Cambazoglu BB, Senkul P, Tokgoz ZO. A framework for sentiment analysis in Turkish: Application to polarity detection of movie reviews in Turkish. In: 27th International Symposium on Computer and Information Sciences; Paris, France; 2012. pp. 437-445.
- [30] Thelwall M. The Heart and soul of the web? Sentiment strength detection in the social web with SentiStrength. In: Holyst J. (editor). *Cyberemotions: Collective Emotions in Cyberspace*. Cham, Switzerland: Springer Publishing, 2017, pp. 119-134.
- [31] Türkmenoğlu C, Tantuğ AC. Sentiment analysis in Turkish media. In: International Conference on Machine Learning; Beijing, China; 2014. pp. 1-11.
- [32] Aytekin Ç. An opinion mining task in Turkish language: a model for assigning opinions in Turkish blogs to the polarities. *Journalism and Mass Communication* 2013; 3 (3): pp. 179-198.
- [33] Kaya M, Conley S. Comparison of sentiment lexicon development techniques for event prediction. *Social Network Analysis and Mining* 2016; 6 (1): 7.
- [34] Ekinçi E, Omurca Sİ. A new approach for a domain-independent Turkish sentiment seed lexicon compilation. *International Arab Journal of Information Technology* 2019; 5: 1-11.
- [35] Gezici G, Yanıkoğlu B. Sentiment analysis in Turkish. In: Oflazer K, Saraçlar M (editor). *Turkish Natural Language Processing*. Cham, Switzerland: Springer Publishing, 2018, pp. 255-271.
- [36] Omurca Sİ, Ekinçi E, Türkmen H. An annotated corpus for Turkish sentiment analysis at sentence level. In: International Artificial Intelligence and Data Processing Symposium; Malatya, Turkey; 2017. pp. 1-5.
- [37] Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S et al. SemEval-2016 task 5: aspect based sentiment analysis. In: 10th International Workshop on Semantic Evaluation; San Diego, CA, USA; 2016. pp. 19-30.
- [38] Strapparava C, Valitutti A. Wordnet affect: an affective extension of wordnet. In: 4th International Conference on Language Resources and Evaluation; Lisbon, Portugal; 2004. pp. 1083-1086.