

Sentiment Analysis in Turkish

Gizem Gezici and Berrin Yanikoglu

Abstract In this chapter, we give an overview of sentiment analysis problem and present a system to estimate the sentiment of movie reviews in Turkish. Our approach combines supervised learning and lexicon-based approaches, making use of a recently constructed Turkish polarity lexicon called SentiTurkNet. For performance evaluation, we investigate the contribution of different feature sets, as well as the effect of lexicon size on the overall classification performance.

1 Introduction

Sentiment analysis aims to identify the *polarity* and strength of the opinions indicated in a given text, that together define its *semantic orientation*. The polarity can be indicated categorically as positive, objective or negative; or numerically, indicating the strength of the opinion in a canonical scale.

Automatic extraction of the sentiment can be very useful in analyzing what people think about specific issues or items, by analyzing large collections of textual data sources such as personal blogs, product review sites, and social media. Commercial interest to this problem has shown to be strong, with companies showing interest to public opinion about their products and financial companies offering advice on the general economic trend by following the sentiment in social media (Pang and Lee, 2008). In the remainder of this chapter, we use the terms “document”, “review” and “text” interchangeably, to refer to the text whose sentiment polarity or opinion strength is to be estimated.

Gizem Gezici
Sabanci University, Istanbul, TURKEY, e-mail: gizemgezici@sabanciuniv.edu

Berrin Yanikoglu
Sabanci University, Istanbul, TURKEY, e-mail: berrin@sabanciuniv.edu

Approaches: There exist two fundamental approaches for sentiment analysis in state-of-the-art: (i) linguistic or lexicon-based (Turney, 2002) and (ii) statistical or based on *supervised learning* (Pang et al., 2002). The first approach has the advantage of being simple, while the second approach is typically more successful since it learns from samples of documents in with known sentiment, in the given domain.

A *polarity lexicon* contains the sentiment polarity of words or phrases. SentiWordnet (Esuli and Sebastiani, 2006) and SenticNet (Poria et al., 2012) are two of the most commonly used domain-independent polarity lexicons, for sentiment analysis. The lexicon-based approach obtains the polarities of the words or phrases in a document from a polarity lexicon, towards the goal of determining the semantic orientation of the document (Turney, 2002). The approach may be as simple as estimating the document polarity from that of the average polarity of the constituent words; or more complex where different properties of the text can be exploited with the hope of obtaining the semantic orientation more accurately; for instance the number of subjective words in the document, or the *purity* of the constituent words may be considered (Taboada et al., 2011). The distinctive aspect of lexicon-based approaches is that they do not involve supervised learning.

In supervised learning approaches, the principle of *learning from data* is adopted. While different learning techniques vary on how they use the available labelled data, called the *training data*, the common approach represents each review in the training data by its features (e.g. length, average word polarity, etc.) and a model is learned to associate each feature vector with the desired output. The problem can be approached as a *classification* (e.g. a review is classified as positive or negative) or *regression* (e.g. number of stars given by a review) problem. Furthermore, classification can be binary (positive/negative) or ternary (positive/negative/neutral), where the problem gets more difficult as the number of considered classes increase.

The model that is learned using the training data is tested on a separate *test* data, in order to measure the generalization performance of the system. However, if there is no designated test set, the available data is split into two datasets as *train* and *validation*, such that the success of the model trained using the training portion is evaluated using the validation portion. For the evaluation, the estimated labels (class labels or regression values) are compared with the true labels assigned to the validation/test data.

In case the available data is not very large, instead of splitting the data as training and test, one can make use of a technique called *cross-validation*. Cross-validation is a model validation technique for assessing how well the results of a predictive model would be generalized to an independent dataset. In *k*-fold cross-validation, the whole data is divided into *k* equal-sized subsets, where *k* - 1 subsets are used for training set and one is used for validation. To reduce variability, this train-test cycle is repeated for multiple rounds; at

each round different partitions are used for training and the results obtained with the validation data are averaged at the end.

In the basic approach to sentiment analysis, the given text is seen as a bag-of-words (BoW); in other words, the document is represented as a set (bag) of words, discarding word location information (Pang et al., 2002). With this representation, the sentence “A is better than B” is the same as “B is better than A”; however in many sentences, the loss of word order does not cause such a drastic affect (e.g. in “excellent movie was”). Alternatives to the bag-of-words approach are also possible, where word polarities of sentences at significant locations (e.g. first and last sentences) are taken into consideration (e.g. (Zhao et al., 2008)).

Some of the supervised learning methods require a polarity lexicon in order to extract features of the given text (e.g. average word polarity, length, and the number of negative words etc.) that are later used in the learning algorithm, in addition to the training corpus; in some other methods, the lexicon is not needed. In the Latent Dirichlet Allocation (LDA) approach which is one of the most successful supervised learning approaches, the probability distributions of topic and word occurrences in the different categories (e.g. positive and negative reviews) are learned by using a training corpus; and the classification of a new text is done according to its likelihood of coming from these different distributions (Bespalov et al., 2011) (Bespalov et al., 2012).

Data Type: There are two main types of data for which automatic sentiment analysis is of interest. In *reviews*, the text is generally longer and people express their opinions on different aspects of the product (e.g. a movie, a hotel, a cell phone). In contrast, *tweets*¹ express opinions in a very limited space, on a variety of topics (products to politics).

In the hotel domain, the TripAdvisor dataset is well-known, consisting of reviews that are crawled from the website of TripAdvisor² which is a famous travel website. In the movie domain, there is a database consisting of reviews from IMDB website³. For product reviews, researchers often use online product reviews from various websites such as Amazon⁴. Tweets in various topics can be collected using keyword searches and are generally difficult to analyze for sentiment, mainly for their short length and spelling errors or abbreviations. For comparing different approaches on the *same* tweet data set, a yearly evaluation campaign is organized (Rosenthal et al., 2014).

Domain-Dependence: Words may have different meanings in different domains: for instance, the word “small” has a negative meaning in the hotel domain whereas it is in general positive in the cellphone domain. Since

¹ <http://twitter.com>

² <http://www.tripadvisor.com>

³ <http://www.imdb.com>

⁴ <http://www.amazon.com>

domain-independent lexicons such as SentiWordNet (Esuli and Sebastiani, 2006) and SenticNet (Poria et al., 2012) do not contain homonyms (a word that has diverse meanings in different contexts), they may mislead the sentiment analysis system. Hence, one may need a domain-specific lexicon which can be constructed by using a corpus of labeled reviews in a specific domain.

The rest of the chapter is as follows. Section 2 will briefly discuss the related work, in Section 3, difficulties encountered in Turkish sentiment analysis will be explained elaborately. In Section 4, we will describe a Turkish sentiment analysis framework and give experimental results on a movie dataset.

2 Related Work

Research in sentiment analysis has been active for the last 15 plus years, with increasing academic and commercial interest to the field. An elaborate survey of the previous works for sentiment analysis has been discussed in (Pang and Lee, 2008) while we only summarize research about the fundamental issues and sentiment analysis for Turkish, here.

In their seminal work (Pang et al., 2002), Pang et al. evaluate several features with three different machine learning methods, on a dataset collected from movie reviews crawled from the well-known internet movie database, IMDB³. The Support Vector Machine (SVM) classifier taking as input the occurrence counts of unigrams and bigrams has been shown to give the best performance (82.9% on 1400 movie reviews).

In the earlier works, the document is typically viewed as a bag-of-words and its sentiment orientation is estimated from features (e.g. the words and frequencies of words in the text) extracted from this “bag” (Hatzivassiloglou and McKeown, 1997)(Pang et al., 2002)(Pang and Lee, 2004)(Mao and Lebanon, 2006). Since the relative location of words in the document is lost in the bag-of-words approach, researchers sought to find different methods and focused on analysis of phrases and sentences for a complex analysis. Wilson et al. (2004) showed the data in a tree structure and generated features for displaying the relations in the tree with the help of boosting and rule-based methods. Gezici et al. (2012) analyzed sentence-level features in order to bridge the large gap between word and review-level sentiment analysis.

In searching for features at different levels, it has been discovered that one of the most important review properties which is highly relevant to sentiment analysis is *subjectivity*. It has been found that identifying subjective parts within the text first, may help to estimate the overall sentiment more accurately. In an early study, Hatzivassiloglou and Wiebe investigates the impacts of adjective orientation and gradability on sentence subjectivity (Wiebe, 2000). The aim of the approach is to understand whether a given sentence is subjective or not, by looking at the adjectives in that sentence.

A broad subjectivity analysis can be found in (Wiebe et al., 2004) which presents a comprehensive survey of subjectivity recognition using various features and clues. One of the first datasets generated for the classification of subjectivity consists of 5000 subjective movie review snippets and 5000 movie plot sentences which are assumed to be objective (Pang and Lee, 2004). Using this dataset, Pang and Lee built a two-layer algorithm for classification, where the first layer differentiated subjective sentences from objective ones; and classified subjective sentences as positive or negative. The two-layer classification process increased the overall result by 4% from 82.9% (Pang et al., 2002) to 86.9% (Pang and Lee, 2004).

Most sentiment analysis research in literature is done for English and most sentiment analysis resources (e.g., polarity lexicons, parsers) are established for English as well. Research on sentiment analysis of non-English texts has picked interest in recent years. For instance, Ghorbel and Jacot (2011) formulated a method to classify French movie reviews by using supervised learning and linguistic features that are extracted with part-of-speech tagging and chunking, using the semantic orientation information of words from SentiWordNet (Esuli and Sebastiani, 2006). French words in the reviews are translated to English so as to obtain their semantic orientation from SentiWordNet.

Sentiment analysis of texts in Turkish has attracted research interest in recent years and there is still a lot to do in the field. Erogul's Master thesis work (2009) is one of the first studies in Turkish sentiment analysis. In this work, Turkish movie reviews, crawled from the well-known website of BeyazPerde ⁵, are classified using an SVM classifier. The study uses n-grams as features and studies the effect of part-of-speech tagging, spell-checking and stemming on the overall result. An 85% of accuracy is achieved on the binary sentiment classification of Turkish movie reviews.

More recently, Vural et al. (2013) proposed a lexicon-based sentiment analysis system using SentiStrength, which is a lexicon-based sentiment analysis library developed by (Thelwall et al., 2010). The library generates a positive or a negative sentiment score for each word in a given text. Authors evaluated their unsupervised Turkish sentiment analysis framework on the same dataset that was already used in (Erogul, 2009) and report an accuracy of 76% for positive/negative classification.

Türkmenoğlu and Tantuğ (2014) also introduce a lexicon-based framework which is similar to the systems described in (Thelwall et al., 2010) and Vural et al. (2013), with additional handling of simple negation and multi-word expressions. This system is reported to achieve 79.0% and 75.2% accuracies on the movie and Twitter datasets, respectively.

Apart from the binary classification of texts in Turkish, a deeper analysis for *emotion analysis* has also attracted attention from researchers. In her master thesis, Boynukalin (2012) presented an emotion classification frame-

⁵ <http://www.beyazperde.com>

work for Turkish. For experimental results, a new dataset for Turkish emotion analysis was created and the effects of newly added features which are compatible with the morphological characteristics of Turkish language were investigated. Within such a framework, the performances of several classifiers were compared on the newly established dataset for Turkish emotion analysis. In another study on emotion analysis of Turkish texts, a methodology indicating the feasibility of a fuzzy-logic representation of Turkish emotion-related words was presented. It was indicated that there is a strong connection between emotions referred to word roots and sentences in Turkish (Cakmak et al., 2012).

Analysis of Turkish political news and tweets has also attracted researchers' interest. Kaya et al. (2012) investigated the performances of supervised machine learning algorithms of Naive Bayes, maximum entropy, SVM and the character based n-gram language models. It was observed that maximum entropy and the n-gram outperformed the SVM and Naive Bayes and achieved 76-77% accuracy with different features. Subsequently, Kaya (2013) described an improved version of their previous system by implementing transfer learning into the existing framework. This system accomplished more than a 25% improvement over the previous system and with all of the three machine learning approaches (Naive Bayes, SVM and maximum entropy), accuracy values over 90% were obtained for the sentiment classification of Turkish political columns.

3 Main Difficulties From the Sentiment Analysis Perspective

The motivation behind building a sentiment analysis framework specific to Turkish, rather than utilizing an already established system for English and translating it into Turkish, is due to certain differences between these two languages. These main differences can be divided and summarized in three categories as follows.

1. *Agglutinative Morphology*: Turkish is an agglutinative language: it is possible to generate new and arbitrarily long words by adding suffixes to a root word. These derivational and inflectional suffixes may change the Part-of-Speech (POS) tagging and semantic orientation of the word (e.g. "beğenme" (to like or don't like), "beğendim" (I liked it), "beğenmedim" (I didn't like it)).

The practical limitation of the agglutinative morphology in speech, handwriting, or sentiment analysis problems is that it makes it infeasible to build a (polarity) lexicon that would need to contain all variants of Turkish words. Hence, sentiment analysis systems for agglutinative languages

like Turkish face some extra challenges compared to those for which a reasonable size lexicon (e.g. 30,000 as for English) is sufficient for many applications.

2. *Negation*: In Turkish, there are many ways a word may be negated in a way that its sentiment polarity will change: with the affixes *me/ma* or *siz/siz* (as in “olmadı” (didn’t happen), “başarısız” (unsuccessful)); or using a separate word such as “değil” or “yok” (as in “güzel değil” (not beautiful) or “konusu yok” (didn’t have a topic)).
3. *Turkish Alphabet*: Turkish has several characters that do not exist in the English alphabet: “ç”, “ğ”, “ı”, “ö”, “ş”, “ü”. In informal writing, people tend to substitute these Turkish letters for the closest ASCII characters (e.g. “ç” is written as “c”), which complicates the mapping of a string to the words. Thus, one needs a preprocessing step before sentiment analysis known as de-ASCIIfication (i.e., converting the ASCII English characters to their Turkish equivalents, to find the words and obtain their polarities from the lexicon).

We established a sentiment analysis framework for Turkish by taking into account the above-mentioned issues, as described in Section 4. Then in Section 5, we give experimental results of this system on the Turkish movie dataset.

4 Practical Sentiment Analysis for Turkish

In this section, we present a basic sentiment analysis system for Turkish, starting with the basic approach and showing how subsequent steps of Turkish NLP and increasing the lexicon size improve the basic results.

We first describe resources that are used to compute the machine learning features of our system in Section 4.1; then the system is described in Section 4.2. We then report experimental results in Section 5, using movie reviews in Turkish. Our evaluation procedure is composed of two main parts: first, we report the effectiveness of different sets of features; next, we investigate the influence of lexicon size on detecting the overall sentiment of the reviews in the same dataset.

4.1 Resources

4.1.1 Polarity Lexicon

Our polarity lexicon is the first comprehensive Turkish lexicon established by Dehkharghani et al. (2014) by using several resources both in English as well as in Turkish *SentiTurkNet*. In building this lexicon, authors did not translate SentiWordNet to Turkish as done in (Türkmenoğlu and Tantuğ, 2014), rather they established the lexicon using NLP techniques and available resources such as Turkish WordNet (Bilgin et al., 2004), English WordNet (Miller, 1995), SentiWordNet (Esuli and Sebastiani, 2006) and SenticNet (Cambria et al., 2010).

SentiTurkNet consists of 15,000 synsets with their Part-of-Speech Taggings (a (adjective), n (noun), v (verb) and b (adverb)), and three associated polarity values (positive, negative and neutral/objective). The polarity scores stand for the measurement of negativity, objectivity and positivity, and they sum up to 1. Some sample entries from SentiTurkNet are provided in Table 1.

Note that a given word may have different synsets with different sentiment polarity; hence, one needs to find the correct synset to obtain the correct sentiment polarity. If this is not feasible, then sentiment polarity values across different synsets corresponding to the given word may be averaged. We took the latter approach in this work.

Table 1 Sample Entries from SentiTurkNet

Synset	POST	Negative	Objective	Positive
mükemmel, kusursuz (wonderful)	a	0	0	1
kötü (bad)	a	0.946	0.018	0.036
çekici, güzel (beautiful)	a	0	0	1
şaka, latife (joke)	n	0.06	0.397	0.543
gülmek (to laugh)	v	0.095	0.095	0.81
fiilen, gerçekten (really)	b	0.06	0.872	0.068

4.1.2 Seed Words

Seed words are highly sentiment-bearing words (e.g. “excellent”, “horrible”) that are expected to be strong indicators of the review sentiment. Seed word sets have been commonly used for sentiment analysis (e.g. (Hu and Liu, 2004) (Qiu et al., 2011)) .

Another advantage of highly highly sentiment-bearing words is that their sentiment polarity does not change much in different domains. For instance, *muhteşem* (*excellent*) is a positive word and *berbat* (*awful*) is a negative word, independent of the context. Hence they may be helpful in domain-independent tasks.

Yet another important quality of seed words is that they are often not used in negated form, simplifying the analysis of the sentiment they carry. For instance while one may use “not very good”, where the sentiment polarity of the word “good” is reversed, it is not common to use highly sentiment-bearing words in negated form (as in: “it was not excellent”).

We use a positive seed word list of 69 words and a negative seed word list of 126 words in the work described here. A sample of 10 positive and 10 negative seed words from this list are shown in Table 2.

Table 2 Sample Seed Words

Positive Words	Type	Negative Words	Type
muhteşem (magnificent)	a	fiyasko (failure)	n
güzel (beautiful)	a	berbat (awful)	a
eğlenceli (enjoyable)	a	hayalkırıklığı (disappointment)	n
harika (awesome)	a	sıradan (average)	a
şahane (fantastic)	a	sıkıcı (boring)	a
etkileyici (fascinating)	a	olumsuz (negative)	a
başyapıt (masterpiece)	n	vasat (mediocre)	a
kaliteli (good quality)	a	terrible (crappy)	a
kusursuz (perfect)	a	beğenmedim (I did not like)	v
inanılmaz (incredible)	a	değmez (not worth it)	v

4.1.3 *Booster Word List*

Booster words are adverbs that accentuate the sentiment polarity of the words that follow; for instance the words “very” or “really”, as in “it was a really good movie”. The boosting effect has been already investigated for Turkish (Türkmenoğlu and Tantı, 2014).

We have a very small list of 4 commonly used booster words which are shown in Table 3. Strengthening is done by shifting the polarity value of the corresponding adjective towards its sentiment pole, i.e., positive or negative. We chose a value of 0.4 for shifting.

Table 3 Booster Words

Word	Type
en (the most)	b
gerçekten (really)	b
çok (very)	b
bayağı (too many/much)	b

4.2 Methodology

Our approach combines supervised learning and lexicon-based approaches. In the basic approach, we simply compute the average polarity of the words (adjectives, verbs, and nouns) in the review and train a classifier (Naive Bayes or SVM) to separate positive reviews from negative ones according to this feature. Then we measure the effectiveness of more complex processing techniques or features: handling negation; considering the effects of booster words; and using more features derived from seed words. In all of these approaches, the document is viewed as a bag-of-words approach with features in increasing complexity.

4.2.1 Preprocessing

Before feature extraction, several steps are necessary as preprocessing steps, in order to obtain the corresponding polarity values of the word in a review. These polarity values form the basis of the features used in this work.

As an initial step, we tokenize the given text into words and then we use the Zemberek Tool (Akın and Akın, 2007) for de-ASCIIification in identifying the words in a document accurately. An accurate identification of words is needed to obtain correct polarity values from the lexicon.

While obtaining the polarity value for each word in the document, we follow the following procedure: We first search the word itself in the lexicon (SentiTurkNet) with the POSTag information. If the word is not found, then we identify the root with Zemberek and search for the root in the lexicon. If we still cannot find it and the POSTag of the word is verb, then we search the root of the word by adding the infinitive suffixes (mek/mak) to the end of it. If none of these help to find the polarity values for the word, this means that the word does not exist in the lexicon, therefore its polarity values are set to 0.

The process is illustrated in Table 4 for the sample word “hoslanmadim (I did not like it)” which is an inflected verb with some substituted letters.

Table 4 Sample Preprocessing

Input	Preprocessing Step	Processed Form	Lexicon Search
hoslanmadim	De-ASCIIification	hoşlanmadım	Not Successful
hoşlanmadım	Obtaining the root	hoşlan	Not Successful
hoşlan	Adding infinitive suffixes	hoşlan+(mak)	Successful

A word in the lexicon may have multiple synset entries. In order to get the correct polarity values, it is important to find the correct synset, or as a lesser alternative, to compute an average of the polarity values of all corresponding synsets. We take this approach in this work for simplicity. The polarity values

for different synsets of the word “hoşlanmak” are shown in Table 5, together with the average polarity values.

Table 5 Obtaining Polarities from SentiTurkNet

Word	POSTag	Negative	Objective	Positive
hoşlanmak	verb	0.060	0.002	0.938
hoşlanmak	verb	0.125	0.125	0.750
hoşlanmak	verb	0.093	0.064	0.844

Here

4.2.2 Basic approach

In the basic approach, we only use the average polarity of the constituent words to estimate the document polarity. The overall average sentiment polarity is computed by averaging the polarity of all potentially sentiment-bearing words in the document (adjectives, verbs and nouns), while adverbs affect the overall polarity indirectly if they are in the booster list shown in Table 3. The average polarity of a given text is calculated as follows:

$$F_1 = \frac{1}{N} \sum_{w_i} pol(w_i) \quad (1)$$

where w_i are the corresponding words in the document, N is the total number of sentiment-bearing words and $pol(w_i)$ is calculated from the polarity values obtained from SentiTurkNet. The *average polarity* of a word w , denoted by $pol(w)$, is calculated as:

$$pol(w) = (pol^+ - pol^-)/2 \quad (2)$$

where pol^+ and pol^- represent the positive and negative polarity values assigned to the word w in the polarity lexicon respectively. For simplicity, we do not take into account of the neutral/objective polarity value of the word in this work. For example, the polarity triplet of the word “kabus” (nightmare) is $\langle 0.535, 0.408, 0.057 \rangle$ and $pol^- = 0.535$, $pol^+ = 0.0570$; hence $pol(\text{“kabus”})$ is calculated to be -0.239 since “kabus” has only one synset in SentiTurkNet.

An alternative to using the average polarity is to use the *dominant polarity* of a word (Demiröz et al., 2012); however we preferred the average as it takes into consideration both polarity values.

In this basic approach, the classifier is trained with this single feature F_1 .

4.2.3 Handling negation

In comparison to English, negation handling is quite complicated for Turkish. For instance, in English the word *not* is used for negation purposes, while in Turkish negation can happen in several different forms (see Section 3).

In our approach, we take into consideration the negating word “değil” as well as the absence/presence suffixes (*sız/siz (without)*, *lı/li (with)*) since SentiTurkNet does not include the polarity of these derived forms. These suffixes modify the part-of-speech tagging of words and represent a form of negation. We also include the negation suffixes of “me/ma” in our system which may have a negation effect on verbs, for a comprehensive negation analysis in Turkish. Our negation analysis does not contain the word of *yok* because it requires the negation analysis at review-level instead of word-level, differently from the previously considered negation cases.

For each negated word, we negate its polarity $pol(w_i)$ as defined in Eq. 2, to obtain the negated polarity. Once we re-compute negated word polarities, the average review polarity is also re-calculated to give feature F_2 .

Sample negated words are displayed in Table 6, for a proper understanding.

Table 6 Sample Negated Words

Word	Type
umut-suz (hope-less)	a
başarı-sız (un-successful)	a
beğen-me-dim (did not like it)	v
sev-me-dim (did not love it)	v

4.2.4 Booster Effect

As mentioned in Section 4.1.3, the booster words are adverbs that strengthen the meaning of adjectives that they come before.

Considering the booster effect, we compute the average review polarity by considering the effects of booster words shown in Table 3, to obtain the feature F_3 .

4.2.5 Seed Words

We have chosen a positive seed word list of 34 words and a negative seed word list of 93 words, as discussed in Section 4.1.2. The seed word approach aims to give a sentiment polarity estimate that is less error-prone in comparison to using a large polarity lexicon that may contain errors. The corresponding features are the positive and negative seed words count in a review.

$$F_5 = \sum_{w_i} PositiveSeed(w_i)$$

where w_i are the sentiment-bearing words that are adjectives, adverbs, verbs and nouns in the document and $PositiveSeed(w_i)$ returns 1 if the word w_i is a positive seed word and zero otherwise.

4.2.6 Sample Analysis

Table 7 shows the feature values for 3 separate sentences. The first example contains a booster; therefore the average polarity of the following adjective is shifted by 0.4 towards the positive pole and the average polarity with booster handling (F_3) is found to be -0.37. In the second example there is a negation suffix (-me) and the average polarity with negation handling (F_3) is found to be -0.37. The third example has neither has a negation nor a booster; however the sentiment-bearing word (hata-error) in this sentence has negative polarity, thereby overall polarity is negative and the same for all of the three features.

Table 7 Sample Sentences

Sample Input	Words	Pol	F_1	F_2	F_3
Cok guzel-di (It was very beautiful)	çok (booster), güzel (a)	0.5	0.5	0.5	0.9
Hiç sev-me-dim (I did not like at all)	sev (v), -me (negation)	0.37	0.37	-0.37	-0.37
Hata-larla dolu (full of errors)	hata (n)	-0.47	-0.47	-0.47	-0.47

4.2.7 Classifier Training

We randomly split the available data into train and test sets containing a balanced number of positive and negative reviews in each. Then, the system is trained using a Naive Bayes or SVM classifier using only the training set and tested on the test set.

Our system is implemented in Java, in the Eclipse environment, which generates intermediate files that are given to WEKA (Hall et al., 2009) to obtain classification results. WEKA is a commonly used machine learning toolbox that provides many supervised as well unsupervised algorithms.

5 *Experimental Evaluation*

We evaluated the considered approach and features using the Turkish movie dataset described in Section 5.1.

The sentiment analysis problem is approached as a binary classification problem and evaluated using the misclassification error on the test data. The success rates of different features and the effect of the polarity lexicon size are analyzed in Sections 5.2 and 5.2 respectively.

5.1 *Database*

The Turkish movie reviews dataset is introduced by Demirtas and Pechenizkiy (2013). The dataset is composed of 5331 positive and 5330 negative movie reviews in Turkish and it is collected from a well-known movie site called Beyazperde. Beyazperde is a platform in which users write reviews about movies and give star ratings of 0 to 5 scale ⁵.

The star ratings of reviews (1 to 5 stars) are used as ground-truth labels for evaluation. Since we only address the binary classification problem, 4 or 5-star reviews are taken as positive reviews; while 1 or 2-star reviews are considered negative. Reviews with 3-stars are excluded from the study, as often done in binary classification evaluations. Some sample positive and negative movie reviews from the database are shown in Table 8 and 9, respectively.

Table 8 Positive Movie Reviews

Review	Gloss
“gerçek bir başyapıt”	“it’s a true masterpiece”
“gelmiş geçmiş en iyi 10 filminden biri”	“it’s one of the top-10 movies ever”
“tek kelimeyle kusursuz”	“ine one word: perfect”

Table 9 Negative Movie Reviews

Review	Gloss
“benim için sadece büyük bir hayalkırıklığı”	“for me it’s just a big disappointment”
“hiç beğenmedim bu filmi”	“I didn’t like this movie at all”
“berbat bir film”	“it’s a terrible movie”

5.2 Results

The reported results are obtained with a Naive Bayes or SVM classifiers, using each of the approaches with increasing complexity.

The results of approaching the problem with the basic or more sophisticated features are given in Table 10; while Table 11 shows the effects of having a larger sized lexicon.

Parameter Optimization

Before the actual training with the SVM classifier, we performed parameter optimization for the SVM classifier (the Naive Bayes classifier does not have parameters to optimize).

For the optimization, we performed a 5-fold cross-validation on the training data and found the best parameter values of 10.0 and 10.0 for the cost and gamma parameters. We then re-trained the system with all of the training data. We used the LibSVM package which is implemented in WEKA (Hall et al., 2009) for parameter optimization, training and testing stages.

Feature Effectiveness

While the basic approach only uses one feature (F_1) to classify a given document, the approach that extends the basic approach with negation handling uses both F_1 and F_2 . Note that this approach could use only F_2 (as it represents a better estimate of the average word polarity), however we have chosen the proposed approach to allow for the possibility that the earlier feature(s) may contain some information that does not exist in the latter one. Similarly, the third approach that considers booster word effect uses three features ($F_1 - F_3$), while the last one that considers seed words occurrence, uses all of the features. As can be seen in this table, the basic approach obtains 67.49% with the Naive Bayes classifier and 67.61% with the SVM classifier; while best results are obtained using all of the features, achieving 74.28% with the Naive Bayes and 75.52% with the SVM classifiers, respectively.

Somewhat surprisingly, negation and booster effect handling do not improve classification accuracy significantly, while considering seed word occurrences does.

Lexicon Effect

The second part of our evaluation investigates the effect of the lexicon size on obtaining the overall sentiment of a given review. Increasing the lexicon size

Table 10 Classification Accuracy with Different Features

Feature Subset	Accuracy (NB)	Accuracy (SVM)
F_1 (Basic)	67.49%	67.61%
$F_1 + F_2$ (... + negation handling)	68.34%	68.92%
$F_1 - F_3$ (... + booster handling)	69.18%	69.78%
$F_1 - F_5$ (... + seed words)	74.28%	75.52%

generally improves the classification performance, since with a small lexicon, the system knows about the semantic orientations of more words.

To generate the lexicons of various sizes, we started with the polarity values of the seed words, obtained from the SentiTurkNet (Dehkharghani et al., 2014), then the rest of the new lexicon was filled by randomly choosing the necessary number of synsets from the lexicon. To obtain more robust results, we randomly chose the rest of the words in the new lexicon five times and obtained the results and computed an average over these. This process was repeated until the lexicon size reached that of SentiTurkNet which contains 15.000 synsets. Also, we only used our basic feature, F_1 to investigate the effect of lexicon size.

Results are displayed in Table 11. As can be seen, a larger lexicon always brings better or at least the same classification performance as the added words convey more information to the system. We see that the system achieves a little improvement with the lexicon size of 1000 over 100 for both of the classifiers. This is probably stemmed from the fact that neither the lexicon size of 100, nor the 1000 are sufficient numbers to achieve a good classification performance. Nonetheless, we observe almost 2% improvement on average for the classifiers when we increase the lexicon size from 1000 to 5000. Subsequently, when we use the whole lexicon, increasing the size from 5000 to 15000, the system achieves more than 8% accuracy increase in classification performance, meaning that we triple the lexicon size from 5000 to 15000, we double the percentage of improvement in classification accuracy, which is 5% on average. Lastly, as expected, we obtain the best results using all of SentiTurkNet as the lexicon, showing over 16% absolute increase in classification accuracy over the smallest lexicon size.

Table 11 The Effects of Lexicon Size on the Classification Performance

Lexicon Size	Accuracy (NB)	Accuracy (SVM)
100	51.27%	51.29%
1000	51.85%	51.88%
5000	52.07%	53.28%
15000 (All)	67.49%	67.61%

References

- Ahmet Afsin Akin and Mehmet Dünder Akin. Zemberek, An Open Source NLP Framework for Turkic Languages. *Structure*, 10, 2007.
- Dmitriy Bespalov, Bing Bai, Yanjun Qi, and Ali Shokoufandeh. Sentiment Classification Based on Supervised Latent N-gram Analysis. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 375–382. ACM, 2011.
- Dmitriy Bespalov, Yanjun Qi, Bing Bai, and Ali Shokoufandeh. Sentiment Classification with Supervised Sequence Embedding. In *Machine Learning and Knowledge Discovery in Databases*, pages 159–174. Springer, 2012.
- Orhan Bilgin, Özlem Çetinoğlu, and Kemal Oflazer. Building a WordNet for Turkish. *Romanian Journal of Information Science and Technology*, 7 (1-2):163–172, 2004.
- Zeynep Boynukalin. Emotion Analysis of Turkish Texts by Using Machine Learning Methods. *Middle East Technical University*, 2012.
- Ozan Cakmak, Abe Kazemzadeh, Serdar Yildirim, and Shri Narayanan. Using Interval Type-2 Fuzzy Logic to Analyze Turkish Emotion Words. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–4, 2012.
- Erik Cambria, Robert Speer, Catherine Havasi, and Amir Hussain. SenticNet: A Publicly Available Semantic Resource for Opinion Mining. In *AAAI Fall Symposium: Commonsense Knowledge*, volume 10, page 02, 2010.
- Rahim Dehkharghani, Yücel Saygin, Berrin Yanikoglu, and Kemal Oflazer. SentiTurkNet: A Turkish Polarity Lexicon for Sentiment Analysis. Technical report, Sabanci University, FENS, 2014.
- Gülşen Demiröz, Berrin Yanikoglu, Dilek Tapucu, and Yücel Saygin. Learning domain-specific polarity lexicons. In *12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, Brussels, Belgium, December 10, 2012*, pages 674–679, 2012.
- Erkin Demirtas and Mykola Pechenizkiy. Cross-Lingual Polarity Detection with Machine Translation. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 9. ACM, 2013.
- Umut Eroglu. Sentiment analysis in Turkish. Master’s thesis, Middle East Technical University, Turkey, 2009.
- Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006.
- Gizem Gezici, Berrin Yanikoglu, Dilek Tapucu, and Yücel Saygin. New Features for Sentiment Analysis: Do Sentences Matter? In *SDAD 2012 The 1st International Workshop on Sentiment Discovery from Affective Data*, page 5, 2012.

- Hatem Ghorbel and David Jacot. Sentiment Analysis of French Movie Reviews. In *Advances in Distributed Agent-Based Retrieval Tools*, pages 97–108. Springer, 2011.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- Vasileios Hatzivassiloglou and Kathleen R McKeown. Predicting the Semantic Orientation of Adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics, 1997.
- Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177. ACM, 2004.
- Mesut Kaya. *Sentiment Analysis of Turkish Political Columns with Transfer Learning*. PhD thesis, Middle East Technical University, 2013.
- Mesut Kaya, Güven Fidan, and Ismail H. Toroslu. Sentiment Analysis of Turkish Political News. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 174–180. IEEE Computer Society, 2012.
- Yi Mao and Guy Lebanon. Isotonic Conditional Random Fields and Local Sentiment Flow. In *Advances in Neural Information Processing Systems*, pages 961–968, 2006.
- George A Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.
- Bo Pang and Lillian Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.
- Bo Pang, Lillian Lee, and Shivakumar 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*, pages 79–86. Association for Computational Linguistics, 2002.
- Soujanya Poria, Alexander Gelbukh, Erik Cambria, Dipankar Das, and Sivaji Bandyopadhyay. Enriching SenticNet Polarity Scores Through Semi-Supervised Fuzzy Clustering. In *12th International Conference on Data Mining Workshops*, pages 709–716. IEEE, 2012.
- Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27, 2011.
- Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80,

- Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL <http://www.aclweb.org/anthology/S14-2009>.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307, 2011.
- Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.
- Cumali Türkmenoğlu and Ahmet Cüneyd Tantuğ. Sentiment Analysis in Turkish Media. Technical report, Istanbul Technical University, 2014.
- Peter D Turney. Thumbs up or Thumbs down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424. Association for Computational Linguistics, 2002.
- A. Gural Vural, B. Barla Cambazoglu, Pinar Senkul, and Z. Ozge Tokgoz. A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish. In *Computer and Information Sciences III*, pages 437–445. Springer, 2013.
- Janyce Wiebe. Learning Subjective Adjectives From Corpora. In *AAAI/IAAI*, pages 735–740, 2000.
- Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning Subjective Language. *Computational Linguistics*, 30(3):277–308, 2004.
- Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just How Mad Are You? Finding Strong and Weak Opinion Clauses. In *AAAI*, volume 4, pages 761–769, 2004.
- Jun Zhao, Kang Liu, and Gen Wang. Adding Redundant Features for CRFs-based Sentence Sentiment Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 117–126. Association for Computational Linguistics, 2008.