Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 31(2), ss. 467-481, Aralık 2016
Çukurova University Journal of the Faculty of Engineering and Architecture, 31(2), pp. 467-481, December 2016

An Experimental Investigation of Document Vector Computation Methods for Sentiment Analysis of Turkish and English Reviews

Furkan GÖZÜKARA[*1], Selma Ayşe ÖZEL[1]

[1]*Çukurova Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana*

Geliş tarihi: 13.06.2016 *Kabul tarihi: 23.11.2016*

Abstract

Sentiment analysis is the task of identifying overall attitude of the given text documents by using text analysis and natural language processing techniques. In this study, we present experimental results of sentiment analysis on movie and product reviews datasets that are in Turkish and English languages by using a Support Vector Machine (SVM) classifier. Moreover, we compare different document vector computation techniques and show their effects on the sentiment analysis. We empirically evaluate SVM types, kernel types, weighting schemes such as TF or TF*IDF, TF variances, IDF variances, tokenization methods, feature selection systems, text preprocessing techniques and vector normalizations. We have obtained 91.33% accuracy as the best on our collected Turkish product reviews dataset by using C-SVC SVM type with linear kernel, log normalization TF* probabilistic IDF weighting scheme, L2 vector normalization, Chi-square feature selection, and unigram word tokenization. A very detailed comparison of the document vector computation methods over Turkish and English datasets are also presented.

Keywords: Sentiment analysis, Classification, Data mining, Product reviews, Support vector machines

Türkçe ve İngilizce Yorumların Duygu Analizinde Doküman Vektörü Hesaplama Yöntemleri için Bir Deneysel İnceleme

Öz

Duygu analizi, verilen bir metin belgesinin genel yargısını, metin analizi ve doğal dil işleme teknikleri kullanarak belirleme işlemidir. Bu çalışmada, İngilizce ve Türkçe dillerinde yazılmış film ve ürün yorumlarının, Destek Vektör Makineleri (DVM) sınıflayıcısı kullanarak yapılan, duygu analizi deney sonuçları yer almaktadır. Bunun yanında, farklı doküman vektör hesaplama yöntemleri karşılaştırılmakta ve bu tekniklerin duygu analizi üzerindeki etkileri gösterilmektedir. DVM türleri, kernel çeşitleri, TF veya TF*IDF gibi ağırlıklandırma yöntemleri, TF türleri, IDF türleri, öznitelik oluşturma yöntemleri, öznitelik seçme sistemleri, metin önişleme teknikleri ve vektör normalizasyon teknikleri deneysel olarak analiz edilmektedir. Oluşturduğumuz Türkçe ürün yorumları veri kümesi üzerinde, doğrusal kernel ile kullanılan C-SVC DVM türü, log normalleştirme TF* olasılıklı IDF ağırlıklandırma yöntemi, L2 vektör normalizasyonu, Ki-kare öznitelik seçme ve tekli kelime öznitelikleri kullanılarak %91.33 doğruluk ile en iyi sonuç elde edilmiştir. Ayrıca doküman vektörü hesaplama yöntemlerinin Türkçe ve İngilizce veri kümeleri üzerindeki detaylı karşılaştırmaları da çalışmada yer almaktadır.

Anahtar Kelimeler: Duygu analizi, Sınıflandırma, Veri madenciliği, Ürün yorumları, Destek vektör makineleri

[*] Sorumlu yazar (Corresponding author): Furkan GÖZÜKARA, *furkangozukara@gmail.com*

1. INTRODUCTION

With the emerging new technologies, today, electronic platforms such as internet blogs, E-commerce websites, digital newspapers, Facebook, Twitter, online forums, etc. have become important for our lives. People not only read these platforms but also leave their opinions, and the analysis of these opinions which is also known as sentiment analysis has become a major interest recently in the natural language processing (NLP) and data mining fields.

Sentiment Analysis is a task that classifies the sentiments expressed in review documents as positive, negative, or neutral. For example, a user may be positive or negative about a movie or a product and writes his/her review accordingly. Automatically determining the polarity of a review is important since it helps users which product to buy or which movie to watch. Sentiment analysis is not only restricted to product or movie reviews, but it can also be applied to other kinds of text documents such as news to collect opinion about a particular entity or a topic. Like traditional text classification, Sentiment Analysis involves data pre-processing, feature selection, and classification steps. However, sentiment analysis of morphologically rich languages such as Turkish still has not got much interest in the literature [1].

In this study, we present results of extensive experimental tests conducted by using an SVM classifier to do sentiment analysis on Turkish product reviews. We use LIBSVM [2] framework for the SVM-based classification task. Furthermore, we present a comparison of our methods on well-known datasets that are used by previous works in the literature. The contributions of this study are summarized as follows:

- Detailed empirical evaluation of SVM types, kernel types, kernel parameters, and shrinking feature of SVMs are made.
- Numerous document vector computation techniques that are TF, IDF, TF*IDF, RF, TF*RF weighting schemes with state-of-the-art TF and IDF variances are experimented over several datasets.
- Unigram word + different character N-gram word tokenization approaches and weight vectors normalizations are evaluated.
- Information Gain (IG) and Chi-square (CHI2) feature selection methods are thoroughly evaluated.
- Preprocessed versions of all of the experimented datasets are made publicly available to provide an objective comparison with the results of this study and future works that will be done by other researchers. Moreover, a new meticulously prepared Turkish product reviews dataset is now available to researchers.
- In earlier studies, usually expensive to compose dictionaries or computationally complex linguistic methodologies are employed to obtain high success rates. However, in this study, we obtain high classification accuracy by solely using the correct combination of the NLP methods and text analysis techniques, and we list these techniques in this study to help researchers for their future studies.
- To our knowledge, this is the first study which evaluates numerous document vector computation methods for sentiment analysis of Turkish product reviews.

The rest of the paper is organized as follows: in the next section, related work for the sentiment analysis is discussed. In the third section, details about the datasets used in this study are presented. The fourth section describes the used classification system. Experimental setup of the study is explained in the fifth section. The sixth section discusses the experimental results which show the effectiveness of the used methods. Comparison of the document vector computation methods over the publicly available datasets is presented in section seven, and finally, the last section concludes the study.

2. RELATED WORK

Sentiment analysis tries to determine the overall attitude (e.g. positive, negative, neutral, and so on) of the given text [3]. There has been much research

in the opinion mining and sentiment analysis area recently. There are three recent major surveys made by Pang and Lee [4], Liu and Zhang [5] and Vinodhini and Chandrasekaran [6].

Pang et al. [7] have conducted classification experiments on a movie reviews dataset by using standard Naive Bayes (NB), Maximum Entropy (ME), and SVM classification algorithms. N-gram based tokenization, Part-of-Speech (POS) [8] tagging, and frequency versus binary term weighting by using Bag-of-Words (BOW) methodology are compared. In the later study, Pang and Lee [9] have proposed a system to extract objectivity containing part of the reviews to decrease the size and complexity of the dataset so that the objective part is discarded as it is not useful when classifying sentences into their polarities. They have experimented with using standard SVM and NB algorithms to classify reviews into their polarities. The classification accuracy of their dataset is improved from 82.8% to 86.4% by using empirically determined parameters. Hu and Liu [10] have proposed a system for mining opinionated product features from customer reviews. Additionally, they have classified opinionated sentences as positive or negative by using NLP techniques and have achieved average 84.20% accuracy in sentence polarity classification.

Dave et al. [11] have proposed a system to classify product reviews as negative or positive. They have employed substitutions which require extra human intervention as data-preprocessing. For instance, the names of the products are replaced with *productname* so that different product names can be treated equally. A computationally expensive linguistic parser to process words sentence by sentence is also used. Linguistic features such as WordNet, colocation, stemming and negation, are tested however; it is observed that all these features decrease the success rate in all tests except stemming which increases the success rate only in a single test. Li and Liu [12] have proposed a clustering system which uses K-NN clustering algorithm with a combination of TF*IDF weighting scheme, several preprocessing techniques, feature selection and term scoring by the help of the WordNet for polarity classification problem of documents. Wilson et al. [13] have proposed a system to accomplish polarity classification in phrase-level by using subjectivity clues from a lexicon. They have used BoosTexter AdaBoost.HM classification algorithm with 28 features methodology and they have obtained best 65.7% accuracy in polarity classification while their baseline accuracy has been 61.7%. Their system first determines whether a sentence is neutral or polar then the latter sentences are classified as negative, positive, both or neutral. Wilson et al. [14] have expanded their previously proposed system to accomplish phrase-level polarity classification by using subjectivity clues from an expanded lexicon. They have obtained their best success rates by using BoosTexter AdaBoost.MH classification algorithm with 32 features methodology. Their best-obtained accuracy is 76.5% when classifying sentences as polar or neutral. When polar classifying subjective sentences, they have obtained best 83.2% accuracy on manually labeled subjective sentences and 65.9% accuracy on automatically labeled subjective sentences.

Blitzer et al. [15] have proposed a system for better Domain Adaptation to utilize in sentiment classification task. They have improved Structural Correspondence Learning (SCL) algorithm; constructed and publicly released a product comments dataset which consists of four different product categories. Prabowo and Thelwall [16] have proposed a hybrid classification system in which firstly a Rule-Based Classifier (RBC), secondly a Statistics Based Classifier (SBC), thirdly a General Inquirer Based Classifier (GIBC), and finally an SVM classifier are applied. If one classifier fails to classify the test data, the unclassified test data is transferred to the next classifier. However, their SBC system utilizes public search engines' querying mechanism to obtain Closeness Measure. Thus, the system is not applicable to big datasets.

Martineau and Finin [17] have proposed a modified version of the term frequency (TF) * inverse document frequency (IDF) that is named as Delta TF*IDF weighting system to improve the

accuracy of the classification. O'Keefe and Koprinska [18] have empirically evaluated the performance of different weighting metrics and feature selection methods combination by using SVM and NB classifiers. They also have proposed several new feature weighting and feature selection metrics as well. However, their study does not include the Turkish language. Paltoglou and Thelwall [19] have made a thorough empirical analysis on weighting scores used in polarity classification task. They have experimented on publicly available datasets and obtained very high accuracies when compared to original state-of-the-art works done on those datasets. On the English movie reviews dataset [9] they [19] have obtained 96.90% accuracy however in [9] 87.2% accuracy is observed; for product reviews dataset [15] 96.40% accuracy is computed by [19], on the other hands, from 82.75% to 87.90% accuracies are obtained by [15]. However, in our study, as it is observed in [15], we have failed to obtain such high success rates with the proposed metrics in [19].

Jang and Shin [20] have proposed chunked sentiment analysis system, which uses contextual shifters (negation shifters and flow shifters). Negation shifters change the semantic orientation of the words and flow shifters control the flow of the sentiment like "*however*" or "*but*". They have used a polarity dictionary and a subjectivity lexicon as well. Their proposed contextual shifters and chunking methodologies or polarity based weighting system have failed to improve the results statistically significant on the short movie reviews dataset. However, their proposed system achieves better results than baseline TF*IDF weighting scheme in the news corpora. Arora et al. [21] have proposed a system that utilizes Subgraph Mining and Genetic Programming based feature construction to improve the success of SVM-based sentiment classification task. They have obtained best 76.93% accuracy on the English movie reviews dataset [22]. As a base score with just using unigrams, they have obtained 75.66% accuracy. Yessenalina et al. [23] have proposed a system that automatically generates rationales (a portion of the document that is more relevant to its class) to reinforce polarity classification of the documents. Then, these automatically generated rationales are used to generate additional training data to improve the accuracy of the classification task.

Eroğul [24] have experimented on Turkish movie reviews by using techniques which utilize linguistic features of the Turkish language, such as POS tagging, word polarity or spell correction. Additionally, they have tested N-gram methodology and used Zemberek, which is an NLP tool for the Turkish language, in their experiments. They have failed to obtain any better results with their experiments when compared to their bag-of-words methodology based baseline. Kaya et al. [1] have carried out sentiment classification experiments on political news from Turkish news sites. NB, ME, SVM and character based N-gram Language Model classifiers by using Lingpipe DynamicLMClassifier are used in experiments. The best accuracies that are observed are around 75% with empirically determined parameters. Boynukalın [25] have proposed a framework for emotion analysis in the Turkish language. They have combined machine learning algorithms with NLP methodologies and tools for sentiment analysis. NB and SVM classification algorithms are tested with punctuations removal, stop words removal, proper names removal, spelling correction, stemming, negation, POS tagging, and word level N-grams. They have classified sentences of Turkish fairy tales dataset into five opinions (none, sad, anger, fear, and joy) and have obtained 74.5% accuracy by using Complement Naive Bayes (CNB) classifier and 69.42% accuracy by using an SVM classifier. Seker and Al-Naami [26] have experimented on a massive Turkish comments dataset. They have employed user assigned scores for evaluation. SVM, K-NN and C4.5 algorithms are used for the classification task. As a weighting scheme TF*IDF, and for feature selection IG are used. Each algorithm classifies each test sample separately, and the most voted class is chosen as class label. Their system obtains approximately 55% F-Measure. Aytekin [27] have proposed a system to classify comments that belong to

products and services. The proposed system assigns positive and negative polarities to the words in the Turkish language automatically from an English sentiment dictionary. The system uses NB algorithm for classification, utilizes the generated Turkish sentiment dictionary and obtains 72.71% accuracy. Demirtas and Pechenizkiy [28] have proposed a classification system that utilizes machine translation to improve classification accuracy by co-training. They have used NB algorithm for this purpose and obtained approximately 79% accuracy as the best on the English movie reviews dataset [22] and nearly 83% accuracy as the best on the Turkish movie reviews dataset by the proposed co-training classification system. Akba et al. [29] have evaluated different feature selection methods on a Turkish movie reviews dataset. They have employed different NLP tools and preprocessing techniques as well. They have obtained 83.9% F-Measure by using an SVM classifier when binary classifying (positive or negative) the dataset. Both CHI2 and IG based feature selection methods achieve the same F-Measure in polarity classification. They also have tested three categories (positive, negative and neutral) classification by using an SVM and an NB classifier and obtained best 63.3% F-Measure. SVM classifier performs better than NB classifier.

Yıldırım et al. [30] have experimented NLP layers on Turkish sentiment analysis. Their dataset has been collected from Twitter in the telecommunication domain, and they have processed the dataset both automatically and manually to have high confidence data. The effects of normalization, preprocessing, stemming, morphological analyses and POS tagging are tested and 73%-79% accuracy on the dataset with about 6% relative improvement by using empirically determined parameters are obtained. Dehkharghani et al. [31] have semi-automatically generated a Turkish polarity resource, SentiTurkNet, and experimented on Turkish movie reviews. They have used three classifiers that are Logistic Regression (LR), Feedforward Neural Networks, and SVM with sequential minimal optimization algorithm and obtained 91.11% accuracy as the best when classifying synsets into the three polarities. In this test, they have compared their classification results with manually classified polarities. In the final test, a Turkish movie reviews dataset is classified as positive, negative or objective and the highest accuracy achieved is 66.77% by using the LR classifier.

In our study, we evaluate all possible combinations of TF*IDF weighting schemes, several vector normalization methods, unique words versus N-gram tokenization, and CHI2 versus IG feature selection methods as document vector computation methods and compare their performance on several Turkish and English sentiment analysis datasets. We have obtained our best accuracies without depending on any dictionary-based methodology. Therefore, our methods can be applied to any dataset in any language without requiring complex algorithms and computations.

3. DATASETS USED IN THIS STUDY

In this study, we have developed our own dataset by crawling the top 50 popular E-commerce websites in Turkey. Then, comments on products are fetched by our focused crawler. Table 1 shows some statistics about the comments that are crawled by our system. From these comments, 1000 negative and 1000 positive comments are randomly picked according to the scores given by the users. If the user score is less than 50, the comment is classified as negative, and if it is greater than or equal to 50, the comment is classified as positive. Next, we have manually analyzed all of the comments and decided which ones truly express an opinion. Then totally unrelated and neutral comments are removed from the dataset. The final comment dataset is composed of 600 positive and 600 negative comments. We have to note that, we did not make any categorical selection. So these comments are from a wide and very different category of products such as electronic devices, adult products, or perfumes. There are also many comments in the dataset that contains both positive and negative opinions since only fully neutral and unrelated comments are removed.

Table 1. Statistics of all comments that are crawled by our focused crawler system

# of comments without any user score	# of comments having user score 49/100 or below	# of comments having user score 50/100 or above	total # of comments crawled
37.215	2.214	42.234	81.663

In our experimental evaluations we have also used popular publicly available datasets that are English movie review dataset [9] which has 1000 positive and 1000 negative comments; a greater sized English movie review dataset [22] which includes 5330 positive and 5331 negative comments; Turkish movie review dataset [28] that has 5326 positive and 5326 negative reviews; English product review dataset [15] that has reviews from books, DVD, electronics, and kitchen categories such that each category has 1000 positive and 1000 negative comments; and finally Turkish product review dataset [28] having comments for books, DVD, electronics, and kitchen categories and each category has 700 positive and 700 negative comments. Preprocessed versions of all datasets that are used in this study are published on the https://github.com/FurkanGozukara/Sentiment-Analysis to be used by other researchers in their future studies to make more accurate comparison with our results.

4. THE CLASSIFICATION SYSTEM

To conduct more rational experiments, we split the dataset into 10 disjoint sets and apply 10-folds cross-validation for evaluation. In 10-fold cross validation, data is split into non-overlapping, equal sized 10-sets and tests are conducted 10-times with order while leaving one set out as a test set and using the rest of the sets as training set [32]. Then the macro-average of the obtained scores from the tests is calculated, and it becomes the final score. To evaluate classification performance we use both accuracy and F-measure scores. Accuracy is the percentage of correctly classified instances in the test set. F-measure is the harmonic mean of precision and recall of classification of test instances, where precision is the exactness, and recall is the completeness of the classification algorithm.

Our system first applies pre-processing to the dataset. In the pre-processing phase, comments are converted into bag-of-words form, where the text is split into words by space character and each word represents a feature [33]. We did not apply any stemming or stop word removal since it is shown to be not effective in sentiment analysis for the Turkish language [34]. We have converted all of the characters into lower case, discarded single letter words, replaced punctuations with a space character, applied transformation of diacritics and then removed non-ASCII characters. To the best of our knowledge, transformation of diacritics and removal of the non-ASCII characters at the same time is not done before in the Turkish sentiment analysis area. In the transformation of diacritics process, characters are replaced with their non-diacritic version such as "ş" converted into "s". It is critical to note that removal of non-ASCII characters process should not be applied alone or before the transformation of diacritics process. If it is applied, it can cause unintended results such as "*alışveriş*" becomes "*alveri*". Additionally, we have conducted experiments with N-gram transformation. An N-gram is an N-character slice of a longer string which is applied continuously [35]. For example 2-gram representation of term "school" is *"sc", "ch", "ho", "oo", "ol"*. N-character slicing can be replaced with N-words slicing as well however we have not applied word level N-gram transformation.

After preprocessing is done, features are assigned numerical weights and document vectors are formed since SVMs works with numerical values. This process is very crucial part of text classification task according to results of our empirical analysis when SVMs are used. How well the features are weighted directly affects the success of the system. We have applied a combination of different weighting metrics, and they are presented in Section 5.

After data is preprocessed and document vectors are formed, it is properly formatted for LibSVM.

LibSVM accepts sparse data, and that is a big advantage in text classification where there can be thousands of features missing in each document. After data preparation is done, the C parameter is estimated by using the training set. The C parameter defines how much we want to avoid misclassifying the training dataset. There is a practical guide to usage of SVMs published by Hsu et al. [36] and we refer to their study for information about SVM parameters. For determining the best C parameter, we use "grid-search" technique. For not overfitting the training model, we employ 10-fold cross-validation while using the training set. To find out best performing C parameter, we test the following C parameters and choose the best performing one: 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384. Our system is developed in C# programming language by using .NET 4.5 framework and LibSVMSharp wrapper.

5. EXPERIMENTAL SETUP

System development tests are done only on the dataset which is collected by us in this study. The publicly published datasets of the previous literature studies are tested with only certain promising configurations due to lack of resources. Thus, they may perform better with further parameter optimization. We provide these datasets as well on the Github repository and present their results in Section 7.

We have experimented with the following SVM parameters:

- Shrinking: True or False
- SVM Type: C-SVC or nu-SVC
- Kernel Type: Linear, Polynomial, Radial Basis Function, Sigmoid
- Kernel Parameters: d: Degree, g: gamma, r: coef0, c: cost, n: nu, p: epsilon

More information about these parameters can be obtained from the official document of LibSVM framework. In addition to these parameters, we have tested L1, L2, L3, L4 and L5 normalizations of *SVMNodes* provided by LibSVMSharp wrapper.

The formula of Lp normalization is presented in equation 1 where x_i is the ith element of the vector x, n is the number of the elements in the vector and p is integer from 1 to 5:

$$Lp=\|x\|_p=(\sum_{i=1}^{n}|x_i|^p)^{1/p} \quad (1)$$

For document vector computation, we have tested the following metrics:

- Weighting Scheme: TF, IDF, TF*IDF, Relevance Frequency (RF), TF*RF
- Term Frequency (TF) Types: Binary, Raw Frequency, Log Normalization, Double Normalization, and BM25
- Inverse Document Frequency (IDF) Types: Unary, IDF, IDF Smooth, IDF Max, Probabilistic IDF, Delta IDF, BM25 IDF, and Delta BM25 IDF
- Bag-of-words methods: 1-Word Gram, N-character Grams, 1-Word Gram + N-character Grams

An extensive study of above metrics is presented by Paltoglou and Thelwall [19] where we have taken the formulations of the BM25 TF and Delta BM25 IDF schemes. Delta BM25 IDF scheme is a new metric proposed by Paltoglou and Thelwall [19].

Variants of TF weights [19] are given in the below equations where $f_{t,d}$ is the term frequency of term t in document d, k_1 and b are constants of BM25, dl is document length and avg_dl is the average document length of all documents:

$$\text{Binary TF}=\begin{cases}0 & \text{if term t is not in document d}\\ 1 & \text{otherwise}\end{cases} \quad (2)$$

$$\text{Raw Frequency} = f_{t,d} = \text{frequency of term t in document d} \quad (3)$$

$$\text{Log Normalization}=1+\log(f_{t,d}) \quad (4)$$

$$\text{Double Normalization}=0.5+0.5\times\frac{f_{t,d}}{\max_{\{t'\in d\}}f_{t',d}} \quad (5)$$

$$\text{BM25}=\frac{(k_1+1)\times f_{t,d}}{k_1\times\left((1-b)+b\times\frac{dl}{avg_dl}\right)} \quad (6)$$

Variants of IDF [19] weights are shown below where N is number of documents, N_t is number of documents that contain term t, N_P is number of positive class documents, N_{pt} is number of positive class documents that contain term t, N_N is number of negative class documents, and N_{nt} is number of negative class documents that contain term t.

$$\text{Unary IDF}=1 \tag{7}$$

$$\text{IDF}=\log\frac{N}{N_t} \tag{8}$$

$$\text{IDF Smooth}=\log\left(1+\frac{N}{N_t}\right) \tag{9}$$

$$\text{IDF Max}=\log\left(1+\frac{\max_{\{t'\in d\}}N_{t'}}{N_t}\right) \tag{10}$$

$$\text{Probabilistic IDF}=\begin{cases}0 & \text{if } N=N_t\\ \log\frac{N-N_t}{N_t} & \text{otherwise}\end{cases} \tag{11}$$

$$\text{Delta IDF}=\log\left(\frac{N_P\times N_{nt}}{N_N\times N_{pt}}\right) \tag{12}$$

$$\text{Delta BM25 IDF}=\log\left(\frac{(N_P-N_{pt}+0.5)\times N_{nt}+0.5}{(N_N-N_{nt}+0.5)\times N_{pt}+0.5}\right) \tag{13}$$

When implementing Delta IDF to our system, we had to make a modification. Proposed Delta IDF method causes error when N_{nt} or N_{pt} is zero. Therefore, if $N_{nt}=0$, we set the IDF as $log(N_{pt})$ and if $N_{pt}=0$, we set IDF as $log(N_{nt})$. For BM25 we have chosen $k_1=1.2$ and $b=0.95$.

RF is a new metric proposed by Lan et al. [37] and claimed to be more successful than other known techniques for text categorization task. The formula of RF is shown below in equation 14:

$$\text{RF IDF}=\log\left(2+\frac{N_{pt}}{\max(1,N_{nt})}\right) \tag{14}$$

For TF weighting scheme; all TF formulations are tested; for TF*IDF weighting scheme all combinations of TF and IDF formulations are experimented; for RF scheme, RF is implemented as in equation 14 and all combinations of TF*RF are tested. All above algorithms are implemented in C#.

Feature selection techniques are also extensively tested on our dataset. In vectoral representation of text documents, each feature is a dimension thus, feature selection is also known as dimension reduction. For selecting features, top N scored features by a metric such as IG and CHI2 are selected, and the rest of the features are discarded. Feature selection increases the classificaiton performance by reducing the complexity of the problem and can improve the success rate by eliminating noise features. According to our experimental analysis, there is a tradeoff between the performance and the success rate when feature selection is applied, and there isn't a clear pattern for success improvement. For feature selection, we have employed standard definitions of IG and CHI2 algorithms. We refer to the study of Lan et al. [37] for more information about IG and CHI2 algorithms.

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have experimented with every combination of shrinking, SVM types, kernel types, and the parameters of kernels. According to our empirical analysis, shrinking does not affect the results significantly and selecting the default mode as "true" performs the best. For SVM type; C-SVC, for normalization; L2 normalization perform the best. Among the kernel types, the best performing kernel type is Linear kernel. It is fast when compared to the other kernels, and it produces better success rates. The possible explanation of this is since text categorization task already contains so many dimensions, mapping them into higher dimension by using the other kernels decreases the success rate and increases the complexity of the problem. According to our experiments, parameter tuning of the other kernels is not practical due to the massive number of features when thousands of text documents are being classified. Thus, obtaining higher success rates via parameter tuning is harder in the other kernels when compared to the Linear kernel in sentiment analysis field.

After the first experiments have been completed, we have decided to continue experimenting with only Linear kernel, C-SVC as SVM type and none or L2 normalization. At the second stage of the experiments, we have tested every combination of the TF*IDF weighting metrics that presented in Section 5. We present the best performer configuration for each one of the TF and IDF metrics according to accuracy and F-Measure values in Table 2. The first row in the table shows the best configuration which uses TF*IDF weighting scheme, where TF is implemented as Log Normalization and IDF is computed by Probabilistic IDF formula, with L2 normalization. The worst performer weighting scheme, which is given in the last row of the table, uses raw frequency as TF, and Delta BM25 IDF as IDF without any normalization.

Next, we have experimented N-gram transformations in two different settings with a combination of all of the weighting metrics presented in Section 5. In the first mode, the documents are transformed into N-gram literals and in the second mode, the documents are transformed into unigram word + N-gram literals representation. At the end of the experiments, we observed that none of the experimented N-gram configurations are more successful than the best configuration presented in Table 2 which do not involve any N-grams. Thus, we propose that in Turkish sentiment analysis of e-commerce product reviews, using N-grams does not lead to any improvement. Furthermore, when N is chosen as 4 or bigger, the dimension of the problem notably increases since the number of features is increased as shown in Table 3. Thus, the complexity and the processing times significantly increase. Additionally, when a lower N value is chosen, the number of distinctive features between documents decreases, therefore, SVM classifier takes much more time to converge when training the model.

Table 2. Best results of each TF and IDF weighting metrics on our collected dataset

TF	IDF	Normalization	Accuracy	F-measure
Log Normalization	Probablistic IDF	L2	**91.08**	**91.22**
Log Normalization	BM25	L2	90.91	91.02
BM25	Probabilistic IDF	L2	90.83	90.97
BM25	IDF Smooth	L2	90.75	90.83
Log Normalization	IDF	L2	90.75	90.82
Double Normalization	Probabilistic IDF	L2	90.75	90.89
Raw Frequency	IDF Smooth	L2	90.58	90.64
Binary	IDF Smooth	L2	89.91	90.03
Binary	Delta IDF	L2	89.75	89.82
BM25	IDF Smooth	None	89.50	89.50
BM25	RF	None	88.25	88.78
Log Normalization	IDF Max	L2	87.66	87.61
Raw Frequency	RF	None	86.83	87.16
BM25	Delta BM25 IDF	None	86.5	86.23
Raw Frequency	Delta BM25 IDF	None	84.66	83.86

Table 3. Number of features in the entire dataset when N-gram method is applied

N-Gram	Words
1	11,058
2	1,165
3	8,532
4	32,612
5	75,422
6	119,284
7	154,698

We have made extensive spelling correction tests as well. To employ efficient spelling correction, a very fast dictionary system which uses Symmetric Delete spelling correction algorithm is implemented. The dictionary framework is publicly published by Wolf Garbe [38]. The framework provides top N suggestions based on maximum edit distance parameter. Our tests involve all combinations of the following options:

- *Dictionary datasets* [39]: Hunspell, Dict_Aff
- *Dictionary with suffix*: True, False
- *Maximum edit distance*: 0, 1, 2, 3, 4
- *Add top N suggestions to the document*: 0, 1, 2, 3, 4, 5
- *Discard word if not exists in the dictionary*: True, False

However, among the 240 (2*2*5*6*2=240) different configurations shown above, none of the configurations have improved the accuracy of the best configuration. Thus, we propose that in Turkish product comments sentiment analysis, spelling correction does not improve the accuracy even fine tuning of the parameters are made.

Our next experiments are about testing the effects of feature selection. We have experimented all combination of the following settings with the best configuration which is shown in the first row of Table 2. IG and CHI2 methods are used to select top N percent features where

- N is between 0.1%-10% with 0.1% increment
- N is between 10%-99% with 1% increment

The results of the feature selection experiments are presented in Figure 1, Figure 2, Figure 3 and Figure 4 where in Figure 1 and Figure 2, the *X* axis is the top *N%* feature selection ratio and the *Y* axis is the obtained accuracy, and in Figure 3 and Figure 4 the *X* axis is top *N%* feature selection ratio and the *Y* axis is the ratio of documents which become empty after feature selection.

When the best configuration in Table 2 is used without any feature selection, our dataset is classified with 91.08% accuracy. When top 0.1%-10% features are selected with 0.1% increment by using the IG and CHI2 feature selection methods, the best-obtained accuracy of IG is 88.50% where N is 2.6%, and the best-obtained accuracy of CHI2 is 89.25% where N is 8.7% as shown in Figure 1. Figure 2 displays classification accuracy of IG or CHI2 based feature selection when top 10%-99% features are selected with 1% increment. The best-obtained accuracy of IG is 91.25% where N is 99%, and best-obtained accuracy of CHI2 is 91.33% where N is 81%. As shown in Figure 1 and Figure 2, if we select very small number of features, classification accuracy decreases, however if enough sized feature subsets are chosen classification accuracy increase with respect to no feature selection case. Performances of IG and CHI2 feature selection methods are very similar; however CHI2 has slightly better feature selection performance.

Figure 3 and Figure 4 display percentage of training and test documents which do not have any features when feature selection is applied. The values are average of 10-fold cross-validation tests. When IG based feature selection is made, at the very least 72% of the top features need to be selected to utilize all of the documents in the training set, and when CS based feature selection is made, 73% of the features need to be selected. According to Figure 4 when IG based feature selection is made, at the very least 5.8% of the top features need to be selected to classify all of the documents in the test set, and when CHI2 based feature selection is made, 3.8% of the features need to be selected. Number of predictive features or training features that are left to be used when either training the model or classifying the test data, is an important issue when applying feature selection. We ignore training and test documents that become empty after feature selection is made. Another method to apply is to assign empty test documents to the class having the highest number of instances and this second choice is used by WEKA [40] data mining tool which is commonly used in literature by the researchers for classification experiments. However this method may be misleading when the class sizes are not balanced.

7. COMPARISON OF PREVIOUS WORKS AND THIS STUDY

Finally, we have tested our most successful weighting configurations with the ones which are found as successful in the literature. Therefore we have compared five configurations that are given in Table 4 over the publicly available datasets which are English product reviews [15] (Books,

DVD, Electronics, Kitchen, All_Merged), Turkish movie reviews [28], Turkish product reviews [28] (Books, DVD, Electronics, Kitchen, All_Merged), English movie reviews [9], and sentences of English movie reviews [22]. In Table 4, Config_1 and Config_5 are our proposed most successful document vector computation methods; Config_2 and Config_4 are the best performer methods that are proposed by [19], and Config_3 is the method that is found as successful by [37]. The first four methods in Table 4 that are Config_1, Config_2, Config_3, and Config_4 do not apply any feature selection. Therefore we can directly compare their results. However, Config_5 is the feature selection applied version of Config_1, therefore comparing Config_5 with only Config_1 will be meaningful.

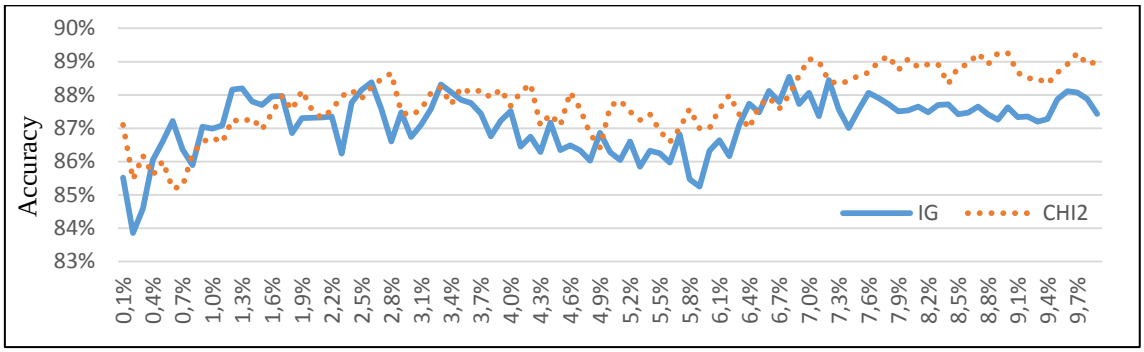


Figure 1. Classification accuracy when top N% (0.1% - 10%) features are selected

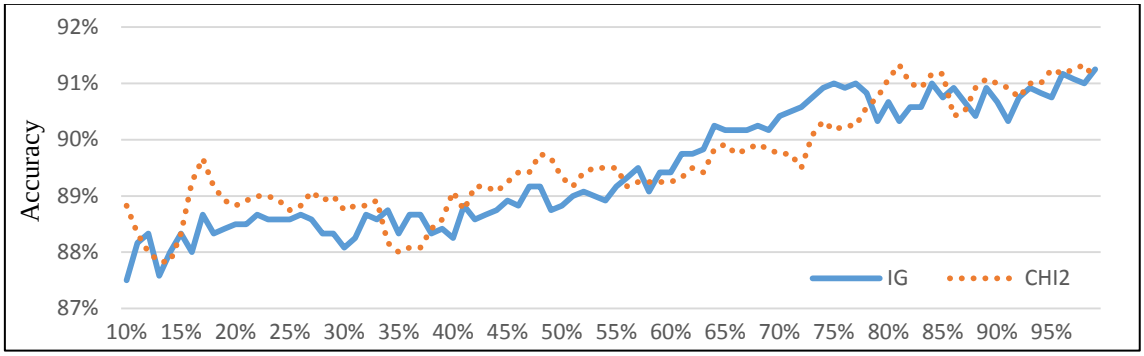


Figure 2. Classification accuracy when top N% (10% - 97%) features are selected

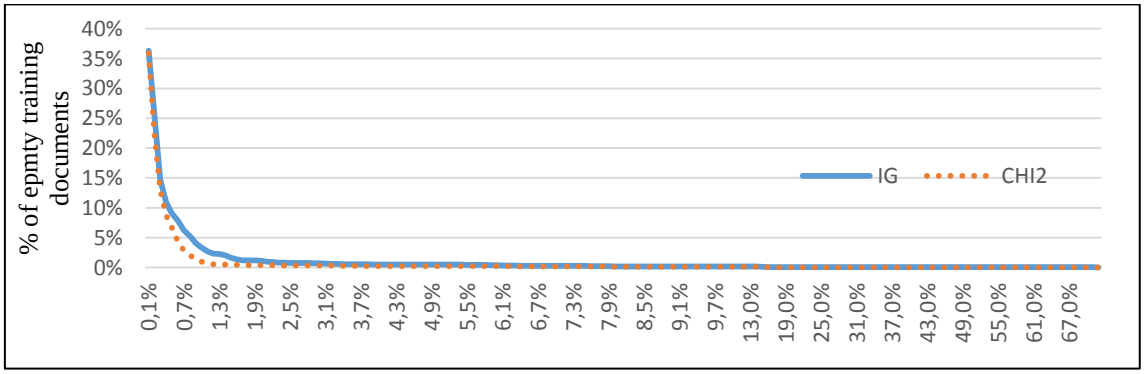


Figure 3. Percentage of training documents that become empty when top N% feature selection is applied

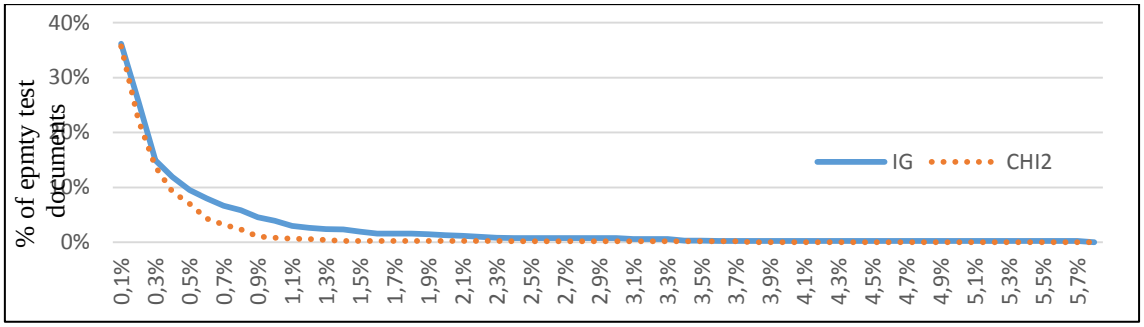


Figure 4. Percentage of test documents that become empty when top N% feature selection is applied

Table 4. List of the tested configurations on the other datasets

	TF	**IDF**	**Normalization**	**Feature Selection**
Config_1:	Log Normalization	Probabilistic IDF	L2	None
Config_2 [19]**:**	BM25	Delta BM25 IDF	None	None
Config_3 [37]**:**	BM25	RF	None	None
Config_4 [19]**:**	BM25	Delta IDF	None	None
Config_5:	Log Normalization	Probabilistic IDF	L2	CHI2 with Top 1% Features

Table 5. Experimental results for English and Turkish movie reviews datasets

	English Movie Reviews [9]		**English Movie Reviews** [22]		**Turkish Movie Reviews** [28]	
	Accuracy	**F-Measure**	**Accuracy**	**F-Measure**	**Accuracy**	**F-Measure**
Config_1	**86.70**	**86.67**	74.45	74.14	**88.21**	**88.15**
Config_2	81.05	81.28	73.69	73.03	85.72	85.27
Config_3	84.55	84.98	73.47	74.12	86.13	86.43
Config_4	85.85	85.97	**75.68**	**75.75**	87.46	87.40
Config_5	82.35	82.52	69.02	69.02	87.41	87.52

Table 5 presents results when the five methods are applied to English and Turkish movie review datasets that are proposed by [9], [22] and [28]. The best accuracy and F-Measure values for each dataset are written in boldface. As shown in the table, for two out of three movie review datasets, our proposed Config_1 method performs better than Config_2, Config_3, and Config_4. Only for one English movie review dataset [22], Config_4 has the best performance; and our proposed Config_1 is the second best. Applying feature selection to our method does not improve results.

In Table 6 and
Table 7, experimental results for the English product reviews [15] and Turkish product reviews [28] datasets are presented. When the results in Table 6 and
Table 7 are analyzed, it is easily observed that our proposed method Config_1 performs better than other methods. Only for Turkish product reviews dataset, when all categories that are books, DVD, electronics, and kitchen are merged, applying feature selection to our method (i.e., Config_5) improves the performance of Config_1, and as it is seen from
Table 7, the difference between Config_1 and Config_5 in terms of accuracy and F-Measure is very small. Applying feature selection to other methods that are Config_2, Config_3, and Config_4 may also improve their classification accuracies and this experiment may be performed as future work.

When all categories are merged for both product review datasets, all five methods perform slightly better. This improvement may be due to the fact

that when we merge all categories, the number of training instances increases and this allows us to learn better classification model.

In our experiments, we observed that there is a significant difference between our obtained results and the results computed by [19] for Delta IDF (i.e., Config_4) and Delta BM25 IDF (i.e., Config_2) methods even on the same dataset. For example, Config_2 obtains 96.90% accuracy on the English Movie Reviews [9] dataset and 96.40% accuracy on the English Product Reviews [15] dataset in [19], however we obtain 81.05% and 81.25% accuracy respectively. Additionally, RF weighting scheme (i.e., Config_3), which is shown as superior to the traditional TF*IDF method by [37], has never surpassed the success rate of the best TF*IDF configuration in any of our experiments. We believe that the reason for obtaining different results is due how feature extraction is made, which data preprocessing method is applied, or how empty documents are handled. These differences in the results also show the importance of making formatted datasets publicly available as we do in this study because all researchers will use exactly the same settings, thus, the results will be much more objectively comparable.

Table 6. Experimental results for English product reviews dataset [15]

	Books		**DVD**		**Electronics**		**Kitchen**		**All Categories**	
	Acc.	**F-M.**	**Acc.**	**F-M.**	**Acc.**	**F-M.**	**Acc.**	**F-M.**	**Acc.**	**F-M.**
Config_1	**84.95**	**85.07**	**86.45**	**86.23**	**87.95**	**87.83**	**90.85**	**90.83**	**88.33**	**88.30**
Config_2	81.25	79.29	81.10	78.82	86.25	85.37	88.95	88.67	84.12	82.89
Config_3	80.50	81.72	81.05	81.96	85.20	85.63	87.45	87.94	84.82	85.29
Config_4	83.20	83.16	84.70	84.52	86.45	86.17	89.20	89.24	87.13	87.04
Config_5	80.15	82.25	81.35	81.28	84.64	84.63	85.79	85.85	83.13	83.17

Table 7. Experimental results for Turkish product reviews dataset [28]

	Books		**DVD**		**Electronics**		**Kitchen**		**All Categories**	
	Acc.	**F-M.**	**Acc.**	**F-M.**	**Acc.**	**F-M.**	**Acc.**	**F-M.**	**Acc.**	**F-M.**
Config_1	**81.92**	**81.85**	**75.35**	**75.73**	**80.92**	**80.50**	**77.07**	**76.11**	83.58	83.46
Config_2	78.50	76.66	67.57	63.21	77.28	77.23	73.64	73.60	81.15	80.45
Config_3	79.50	80.16	74.28	75.15	76.71	77.57	73.35	74.24	82.11	82.40
Config_4	77.07	76.57	72.64	71.99	77.14	76.96	76.14	75.74	82.85	82.57
Config_5	80.62	80.63	75.05	74.98	77.59	75.77	75.61	74.07	**83.97**	**83.91**

8. CONCLUSION

In this study, we have presented through experimental results of sentiment analysis on different datasets and provided a detailed comparison of state-of-the-art weighting metrics. Our results clearly indicate that some of the previously proposed metrics need further experiments for verification. Furthermore, we propose that, with the proper parameter tuning, just using the statistical weighting schemes are as good as expensive linguistic methodologies. We observe that using log normalization TF * probabilistic IDF weighting scheme with L2 vector normalization has the best classification accuracy in both Turkish and English datasets. Additionally we provide preprocessed versions of all datasets that we have experimented in this study on a GitHub repository. We believe that this contribution is significantly important since it provides easiness and objective comparability for the researchers. As a future study, we plan to do research for a better weighting scheme to employ in sentiment analysis task and perform more comprehensive experiments on the different datasets.

9. ACKNOWLEDGEMENT

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) scholarship 2211-C.

10. REFERENCES

1. Kaya, M., Fidan, G., Toroslu, I.H., 2012. 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, 01:174-180.

2. Chang, C.C., Lin, C.J. 2011. LIBSVM: A Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology (TIST), 2:3, p. 27.

3. Melville, P., Gryc, W., Lawrence, R. D., 2009. Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification, In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1275-1284.

4. Pang, B., Lee, L., 2008. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, 2:1-2, p. 1-135.

5. Liu, B., Zhang, L., 2012. A Survey of Opinion Mining and Sentiment Analysis, In Mining Text Data, 415-463.

6. Vinodhini, G., Chandrasekaran, R., 2012. Sentiment Analysis and Opinion Mining: A Survey, International Journal of Advanced Research in Computer Science and Software Engineering, 2: 6, p. 282-292.

7. Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques, In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 10:79-86.

8. Brown, R. W., 1957. Linguistic Determinism and the Part of Speech, The Journal of Abnormal and Social Psychology, 55:1-5.

9. Pang, B., Lee, L., 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 271-278.

10. Hu, M., Liu, B., 2004. Mining and Summarizing Customer Reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168-177.

11. Dave, K., Lawrence, S., Pennock, D. M., 2003. Mining the Peanut Gallery, Opinion Extraction and Semantic Classification of Product Reviews, Proceedings of the 12th International Conference on World Wide Web, 519-528.

12. Li, G., Liu, F., 2012. Application of a Clustering Method on Sentiment Analysis, Journal of Information Science, 38:2, p. 127-139.

13. Wilson, T., Wiebe, J., Hoffmann, P., 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 347-354.

14. Wilson, T., Wiebe, J., Hoffmann, P., 2009. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis, Computational Linguistics, 35:3, p. 399-433.

15. Blitzer, J., Dredze, M., Pereira, F., 2007. Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 440-447.

16. Prabowo, R., Thelwall, M., 2009. Sentiment Analysis: A Combined Approach, Journal of Informetrics, 3:2, p. 143-157.

17. Martineau, J., Finin, T., 2009. Delta Tfidf: An Improved Feature Space for Sentiment Analysis, Proceedings of the Third International Icwsm Conference, 258-261.

18. O'keefe, T., Koprinska, I., 2009. Feature Selection and Weighting Methods in Sentiment Analysis, Proceedings of the 14th Australasian Document Computing Symposium, Sydney, 67-74.

19. Paltoglou, G., Thelwall, M., 2010. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis, In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 1386-1395.

20. Jang, H., Shin, H., 2010. Language-Specific Sentiment Analysis in Morphologically Rich Languages, Proceedings of the 23rd

International Conference on Computational Linguistics: Posters, 498-506.

21. Arora, S., Mayfield, E., Penstein-Rosé, C., Nyberg, E., 2010. Sentiment Classification Using Automatically Extracted Subgraph Features, In Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, 131-139.

22. Pang, B., Lee, L., 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization With Respect To Rating Scales, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 3:1, p. 115-124.

23. Yessenalina, A., Choi, Y., Cardie, C., 2010. Automatically Generating Annotator Rationales to Improve Sentiment Classification, In Proceedings of the ACL 2010 Conference Short Papers, 336-341.

24. Eroğul, U., 2009. Sentiment Analysis in Turkish, M.S. Thesis. The Graduate School of Natural and Applied Sciences of Middle East Technical University.

25. Boynukalin, Z., 2012. Emotion Analysis of Turkish Texts by Using Machine Learning Methods, M.S. Thesis. The Graduate School of Natural and Applied Sciences of Middle East Technical University.

26. Seker, S.E., Al-Naami, K., 2013. Sentimental Analysis on Turkish Blogs via Ensemble Classifier, In Proc. The 2013 International Conference on Data Mining, 10-16.

27. Aytekin, Ç., 2013. An Opinion Mining Task in Turkish Language: A Model for Assigning Opinions in Turkish Blogs to the Polarities, Journalism and Mass Communication, 3:3, p. 179-198.

28. Demirtas, E., Pechenizkiy, M., 2013. Cross-Lingual Polarity Detection with Machine Translation, Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, p. 9.

29. Akba, F., Uçan, A., Sezer, E., Sever, H., 2014. Assessment of Feature Selection Metrics for Sentiment Analyses: Turkish Movie Reviews, 8th European Conference on Data Mining 2014, 180-184.

30. Yıldırım, E., Çetin, F.S., Eryiğit, G., Temel, T., 2015. The Impact of NLP on Turkish Sentiment Analysis, Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 7:1.

31. Dehkharghani, R., Saygin, Y., Yanikoglu, B., Oflazer, K., 2015. Sentiturknet: A Turkish Polarity Lexicon for Sentiment Analysis, Language Resources and Evaluation, 1-19.

32. Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, International Joint Conference on Artificial Intelligence, 14:2, p. 1137-1143.

33. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M., 2010. Short Text Classification in Twitter to Improve Information Filtering, In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 841-842.

34. Parlar, T., 2016. Feature Selection for Sentiment Analysis in Turkish Texts, Ph.D. Dissertation, Çukurova University, Institute of Natural and Applied Sciences, the Faculty of Engineering and Architecture Electrical & Electronics Engineering.

35. Cavnar, W.B., Trenkle, J.M., 1994. N-Gram-Based Text Categorization, In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 161-175.

36. Hsu, C.W., Chang, C.C., Lin, C.J., 2010. A Practical Guide to Support Vector Classification, Department of Computer Science National, Taiwan University, Taipei 106, Taiwan, 1-16.

37. Lan, M., Tan, C.L., Su, J., Lu, Y., 2009. Supervised and Traditional Term Weighting Methods for Automatic Text Categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31:4, p. 721-735.

38. https://Github.Com/Wolfgarbe/Symspell. Accessed, 10.06.2016.

39. https://Code.Google.Com/Archive/P/Tr-Spell/. Accessed, 10.06.2016.

40. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA Data Mining Software: An Update, ACM SIGKDD Explorations Newsletter, 11:1, p. 10-18.