Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 34(3), ss. 95-107, Eylül 2019
Çukurova University Journal of the Faculty of Engineering and Architecture, 34(3), pp. 95-107, September 2019

Effects of Feature Extraction Techniques on Classification of Turkish Texts

Özge AKDOĞAN[1], Selma Ayşe ÖZEL[*2]

[1]*Gaziantep University, Nizip Vocational School, Gaziantep*
[2]*Çukurova University, Faculty of Engineering, Computer Engineering Department, Adana*

Geliş tarihi: 27.05.2019 *Kabul tarihi: 30.09.2019*

Abstract

Feature extraction is the most important preprocessing step of text classification task. Effects of preprocessing techniques on text mining for English have been extensively studied. However, studies for Turkish are limited and generally belong to a specific problem domain. In this study, we investigate the effects of feature extraction techniques on four different Turkish text classification problems including news classification, spam e-mail detection, sentiment analysis, and author detection to show the differences and similarities among the problems. We also propose a new feature selection method to reduce feature space. The experimental analysis has showed that, stopword removal improves classification performance. However, stemming does not make any positive effect on classification accuracy. The most successful term weighting methods are *tf* and *tf*idf*. The proposed feature selection method improves classification performance and has higher accuracy than the well-known methods.

Keywords: Text classification, Preprocessing methods, Feature extraction, Turkish texts

Nitelik Çıkarımı Yöntemlerinin Türkçe Metinlerin Sınıflandırılmasına Etkisi

Öz

Nitelik çıkarımı metin sınıflamanın en önemli önişleme adımıdır. Önişleme tekniklerinin İngilizce metin sınıflandırma üzerindeki etkisi çok çalışılmış bir konu olmasına rağmen, Türkçe için bu konuda yapılmış çalışmalar oldukça sınırlı ve belirli bir problem alanına bağlıdır. Bu çalışmada nitelik çıkarımının haber sınıflama, spam e-posta tespiti, duygu analizi ve yazar tanımayı içeren dört farklı Türkçe metin sınıflandırma problemi üzerindeki etkisi araştırılmış ve problemler arasındaki benzerlik ve farklılıklar gözlenmiştir. Ayrıca yeni bir nitelik seçimi yöntemi önerilmiştir. Deneysel analizler sonucunda durak kelimelerin çıkarılmasının sınıflandırma performansını artırdığı görülmüştür. Ancak kelime köklerinin alınmasının sınıflandırma doğruluğu üzerinde olumlu bir etkisi gözlenmemiştir. En başarılı terim ağırlıklandırma yöntemlerinin *tf* ve *tf*idf* olduğu görülmüştür. Önerilen nitelik seçimi yöntemi sınıflandırma performansını iyileştirmiş ve sıklıkla kullanılan yöntemlerden daha yüksek doğruluk değerine sahip olmuştur.

Anahtar Kelimeler: Metin sınıflandırma, Önişleme yöntemleri, Nitelik çıkarımı, Türkçe metinler

[*]Sorumlu yazar (Corresponding author): Selma Ayşe ÖZEL, *saozel@cu.edu.tr*

1. INTRODUCTION

The development of Information Technology has resulted in the rapid growth of e-mail usage, forming blogs, content sharing in social media, e-commerce activities on the internet, that cause to increase in the dimensions of the online data. As it is impossible to process large data manually, automatic techniques to gather and analyze this data have been necessary over time. Extracting information from these large-scale data by using some machine learning techniques is called as data mining. Data mining is described as "analysis of large observational datasets to summarize data those are understandable and useful to users with new methods and to find unexpected relationships between them" [1]. With the widespread use of the Internet, web, and mobile devices; the unstructured and unprocessed text data flow in different file formats like pdf, txt, doc, html etc. from different sources have accelerated therefore it has been difficult to control these data.

The increasing number of documents on the web revealed the need to classify documents into pre-determined classes and this task is known as a problem in text mining. Text mining is an active research area in data mining field such that, in text mining, pattern is mined from natural language texts rather than proper databases as in data mining [2]. Text classification, which is a sub-problem of text mining, uses attributes extracted from a document and takes a list of predefined categories; then determines the category of the document by using statistical and/or machine learning methods. To apply data mining techniques to text data, the unstructured text documents must be preprocessed at first. Therefore, data preprocessing is the first step for classification algorithms.

There are several preprocessing steps to be applied for text documents such as stemming, stopwords removal, term extraction, term weighting, and feature selection. Stemming is one of the commonly used preprocessing methods in the document classification. Stemming provides us to find the root form of any given term therefore to reduce the feature space. The words which are used in the language often and can be found in almost every text document are called as "stopwords". The stopword removal is another operation that is applied in the pre-processing phase of text mining to reduce feature space and noise.

A computer cannot understand or interpret the words used in the document. The words in the text are transformed into a form that the computer can understand, and term weighting methods are used to accomplish this. In term weighting, a term in the document is assigned a numerical weight, and by using the weights of all terms in the document, a numerical vector for the document is formed.

Feature selection is another important preprocessing step that is applied to make text classification. There are some algorithms to do feature selection on text mining. Main purpose of feature selection is to remove terms which are useless for classification to obtain more successful results. Also, removing irrelevant and useless features decreases the size of datasets and reduces time required to make classification.

In this study, our aim is to investigate the effects of the above mentioned preprocessing methods that are used to extract and form features for text classification on four different Turkish text document classification problems including news classification, spam e-mail detection, sentiment analysis, and author identification so that whether different problem domains require different preprocessing steps or not. Previous studies have compared the effects of these methods only on one problem domain, or only a few preprocessing methods have been analyzed. Therefore, we try to determine the best preprocessing steps are to be done for each text mining problem separately and show if there exists a similarity or difference among the domains. In this study we investigate the effects of using three different stemmers that are Zemberek, Affix Stripping, and Fixed Prefix stemming as well as not applying stemming, and try to determine the best stemming method in general; using stopword removal or not; comparing five different term weighting methods that are *tf*, *tp*, *normtf*, *logtf*, and *tf*idf*, and try to determine the best weighting method for Turkish

texts; and applying feature selection. Additionally, we propose a new feature selector, and compare its performance with the well-known methods that are information gain and chi-square.

The rest of the paper is organized as follows: in the next section related work for Turkish text classification is summarized. The third section includes the datasets used and the methods applied in this paper. The fourth section presents the experimental results and discussions, and finally the last section concludes our study.

2. RELATED WORKS

In this section, we summarize the Turkish text classification studies which investigate the effects of preprocessing methods on different text classification domains.

One of the earliest studies on Turkish text mining is [3] which investigates the effects of using n-grams for classifying Turkish texts. In [3], three different classification problems that are classification of author, type of the text, and gender of the author are studied on the dataset that is collected from Turkish newspapers. For each classification problem, 2 different n-gram models are used namely; bi-gram and tri-gram. When comparing the results, feature selection increases the classification success of the three classification problems. While the bi-gram is the best model for the author, the two n-gram models for types and gender give the same classification results. Naïve Bayes (NB) is the best classifier for the author, the best classifier is Support Vector Machines (SVM) for types and gender [3].

Yıldız et al. [4] proposed a new feature vector computation method in which the class weights are used instead of the weight of the words in the texts, and the sum of these class weights is normalized. The dataset used in this study is collected from daily newspapers and has classes as economy, magazines, health, politics, and sports. Zemberek is used for finding roots. It is observed that the highest accuracy is observed by using the NB classifier.

Çataltepe et al. [5] have analyzed the performance of classifiers when only the consonants in the word stems, and the longest or shortest roots found by a stemmer are used. Two datasets namely, Milliyet and Vikipedi having 5 different categories are used for experimentation. Stopwords are removed from documents. Zemberek is applied for stemming. It is found that when a large number of documents needs to be classified in a short time, only the consonants are taken from the words to form features.

Güran et al. [6] have analyzed the effects of using n-grams on Turkish text classification. The collected news dataset has 6 categories namely; auto, politics, medicine, magazine, economics, and sport. Each document is represented by using unigram words, bigrams words, and trigrams words separated with a pipe. *tf***idf* weighting, and PC-KIMMO stemmer are applied. According to the experimental results, the most successful feature extraction method for all classifiers is the unigram words.

Torunoğlu et al. [7] analyzed the effect of preprocessing methods on classification of Turkish news texts. Two term weighting methods that are binary weighting (*tp*) and term frequency (*tf*) weighting are used. For the classification, NB, Naïve Bayes Multinomial (NBM), SVM, and K-Nearest Neighbor (KNN) methods are applied. Zemberek and Fixed Prefix 3, 5, 7 stemming algorithms are used in the experiments. Stopword filtering is also applied. They have observed that Zemberek is the best stemmer for SVM algorithm. NBM classifier has better performance with stopword removal and fixed prefix 5 stemmer.

Uysal and Günal [8] have studied the influence of the preprocessing tasks on text classification for two different domains and languages that are Turkish and English. They have used four preprocessing methods namely; tokenization, stopword removal, lowercase conversion, and stemming. Zemberek and Fixed Prefix stemming algorithms are used for Turkish, and Porter's stemming algorithm is applied for English. Two different text classification domains that are spam e-mail detection, and news classification in two

different languages namely Turkish and English are studied. For feature selection chi-square (CHI2) method is used. To classify datasets, SVM classifier is applied and Micro-F1 score is computed. Thorough experimental analysis elucidated that significant improvement may be ensured through appropriate combinations of preprocessing tasks on the basis of domain and language while the accuracy may also be reduced by inappropriate combinations. Certain preprocessing steps such as feature extraction and selection in text classification are as significant as the classification step. Despite specific preprocessing tasks employed to ensure an improvement in the classification success with regards to accuracy and dimension reduction irrespective of domain and language, we can talk about no distinctive combination of preprocessing tasks that generate effective classification results for every domain and language examined.

A comprehensive research about text classification is made by Amasyalı et al. [9] who have used tf, tf*idf, binary, log, normalize1 and normalize2 term weighting methods for six different datasets for emotion classification, sentiment analysis, author identification from articles, gender identification, news classification, and author identification from poems. 14 different text representation methods that are formed by using different term weighting methods, word stems, word types, n-grams, functional words, suffixes, concept generalizations, punctuation marks, word counts, sentence counts, inverted sentence counts, letter counts, affix counts, average number of words and letters in sentences, affix counts in words etc. are compared. Zemberek is used to find the roots of the words. According to the experimental results, n-grams based text representation is more successful than other methods. N-gram with binary, log, and normalize1 feature weighting gives better results than other methods.

Açıkalın and Beyazıt [10] have conducted a study to investigate the importance of preprocessing in the classification of Turkish texts. They have used paper abstracts in journals and conferences as a dataset. Latent Dirichlet Allocation is applied to do text preprocessing. Zemberek and fixed prefix with length 5 is used as the stemming methods. Naïve Bayes, Support Vector Machines, and Random Forest are applied as classifiers. As a result, they observed that classifier performance increases with both stemming methods.

Another study on term weighting and feature extraction for emotion analysis has been conducted by Parlar and Özel [11]. A new feature selector is proposed and tp, tf, tf*idf term weighting methods are applied. Naïve Bayes Multinomial is used as classifier. As a result of the experiments, they observed that the tp and tf term weighting methods are more successful than the tf*idf. When the feature selection methods are applied, the classification accuracy for tf*idf increases significantly.

When we compare the previous studies with our study, the most important difference is that we compare effects of preprocessing methods on four different text classification problem domains; however the previous studies make this comparison only for one problem domain, or compare only a few methods on small number of different problem domains. Several stemmers that are Zemberek, Fixed Prefix, Affix Striping, PC-KIMMO, Findstem, A-F, and L-F algorithms have been used in the previous studies. However, majority of the researchers usually preferred Zemberek and Fixed Prefix in the previous studies. In our study, we use Zemberek, Affix Stripping and Fixed Prefix 3, 5, 7 for stemming, and try to compare their performances on four different text classification problems.

As stopwords are often repeated frequently in text, they do not affect classification accuracy and for this reason stopwords are removed from the text in most of the previous studies. In this study, we also show the effect of removing stopwords or not for text classification for different problem domains. In majority of the previous studies, tf and tf*idf methods have been used as term weighting; and only in a few study logtf, normalize1, normalize2 and tp weighting have been used together and compared. In this study, we also use tf, tp, logtf, normtf and tf*idf methods and compare them for different text classification problems.

In the previous studies, NB, NBM, SVM, C4.5, KNN, and Random Forest have been used as classifiers. In this study, we use NBM as the classifier because it is one of the most successful classifiers for text classification problems in the literature. As our aim is only to make comparison of preprocessing methods, we think that using one successful classifier is enough.

We also propose a simple feature selection method which is based on standard deviation of frequencies of features. According to the literature, the researchers often use information gain, and chi square measures for feature selection. Therefore in this study, we compare our proposed method with information gain and chi square to show the effect of feature selection on different text classification problems.

In this study, we try to determine the best preprocessing methods for four different text classification tasks and investigate the answer of the research question "should we apply the same preprocessing methods for all text classification tasks? or should we use problem specific preprocessing methods?".

3. MATERIAL AND METHOD

3.1. Material

The datasets used in this study consist of 1150 News, 3000 Tweets, Turkish Email, and 25 Authors which have different lengths and characteristics, belonging to the four different text classification problems. 1150 News dataset [12] contains 1150 newspaper articles written in Turkish from 5 different classes namely; economy, magazine, health, politics and sport. 3000 Tweets dataset [13] is used for sentiment analysis, and it contains 3 classes such that positive class has 756 documents, negative class has 1287 documents, and neutral class has 957 documents. In Turkish Email dataset [14] there are 2 different classes which are called as normal e-mail, and spam e-mail. Each class has 400 documents in it. 25 Authors dataset is formed from the 2500 columnist dataset [15] which consists of 50 classes and 50 documents for each class totally have 2500 documents. The dataset contains articles of various columnists. In this study, we have chosen 25 classes from 2500 columnist dataset randomly. Detailed information about the datasets used in this study are presented in Table 1 where number of words shows the total number of words in the datasets, and number of features gives the number of unique words extracted from the datasets.

Table 1. Properties of the datasets

Dataset	# of classes	# of instances	# of words	# of features
1150 News	5	1,150	187,638	44,983
3000 Tweet	3	3,000	33,624	10,780
Turkish Email	2	800	180,793	46,279
25 Author	25	1,250	486,360	94,370

3.2. Methods

In the below subsections we summarize the methods used for preprocessing that includes stemming algorithms, term weighting methods, and feature selection methods.

3.2.1. Stemming Algorithms

In text mining and information retrieval, the process of finding root of a word is named as "stemming". There are lots of stemming algorithms each of which is designed for different languages. Stemming algorithm is language dependent because it must use appropriate language rules to find roots of terms. In this study we use Zemberek [16], Affix Striping [17], and Fixed Prefix [18] algorithms that are developed for Turkish language. Zemberek [16] is an open source platform independent NLP framework. Affix stripping [17] is a different morphological analyzer applied to Turkish. Affix stripping is proposed for doing the analysis of Turkish words without using any lexicon [17]. The fixed prefix stemming method [18] is actually a pseudo stemming technique such that it takes just the first n characters of the word as its stem. If the word has less than n characters, in that case it takes the

whole word as its stem [18]. In this study we use 3, 5, 7 characters as the lengths of the word stems.

3.2.2. Term Weighting Methods

Term weighting is an important preprocessing step in text classification, and in this step, we assign weights to terms with respect to their importance in the documents. By using term weighting methods, terms in each document are represented with numeric weight values, and then the document is converted to a numeric vector as in equation 1 to be processed by the statistical or machine learning based classification algorithms. Otherwise, classification algorithms can not process unstructured textual documents for text mining applications.

$$d_i=[w_{i1}, w_{i2}, ...,w_{in}] \quad (1)$$

where d_i is the numeric vector representation for document i, w_{ij} is the weight of the j^{th} term for document i for $1 \leq j \leq n$, and n is the number of unique terms extracted from the whole document collection.

To apply term weighting methods, at first, all unique terms in the document collection are extracted. If we have N documents in the document collection, and we extract n unique terms from the collection, we compute a N×n document-term matrix (i.e., W), where each row represents a document, and each column represent a term in the collection as in equation 2. Any matrix entry w_{ij} is the weight of term j for document i which is computed by using the term weighting methods.

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{Nn} \end{bmatrix} \quad (2)$$

In this study five different term weighting methods that are term frequency (tf), normalized term frequency (normtf), log normalized term frequency (logtf,), term frequency * inverse document frequency (tf*idf), and term presence (tp) methods are used to compute document-term matrix and their performances are compared for the selected problem domains. These weighting methods are summarized as follows [11,19]:

Term frequency (tf_{ij}) is the observed frequency of term j in the document i. It is calculated separately for each term in the document as in equation 3.

$$w_{ij}=tf_{ij}=\text{frequency of term j in document I} \quad (3)$$

Normalized term frequency ($normtf_{ij}$) of term j for document i is obtained by normalizing the term frequency of the term for the document by dividing with the document length. Document length is the total number of terms that the document has, and the normalized term frequency is computed as in equation 4.

$$w_{ij}=normtf_{ij}=\frac{tf_{ij}}{\sum_j tf_{ij}} \quad (4)$$

Log normalized term frequency ($Logtf_{ij}$) is obtained by taking the logarithm of the term frequency in base 10 as in equation 5.

$$w_{ij}=Logtf_{ij}=\log(tf_{ij}) \quad (5)$$

Term presence (tp_{ij}) is the binary weighting method in which if the term frequency for term *j* is greater than zero for document *i*, it is equal to 1, otherwise it is equal to 0 (see equation 6).

$$w_{ij}=tp_{ij}=\begin{cases} 0, & \text{if } tf_{ij}=0 \\ 1, & \text{if } tf_{ij}>0 \end{cases} \quad (6)$$

Term frequency and inverse document frequency ($tf_{ij}*idf_j$) is one of the most popular weighting methods, and it is calculated by multiplying the term frequency of term j in document i with the inverse document frequency of the term. In general, we assume that a term is important for a document if it frequently occurs in that document. Therefore, tf shows the importance of the term for the document. On the other hand, idf of a term is computed by considering the whole document collection as given in equation 7, and it emphasizes the specificity of the term, meaning that if a term occurs only in a few documents in the collection it is an important term. Inversely, if

a term occurs in all documents, it is highly probable that it is a stopword, and this word has idf value which is equal to 0.

$$idf_j=\log\frac{|D|}{df_j} \quad (7)$$

In equation 7, $|D|$ is the number of documents in the dataset, and df_j is the number of documents that contain term j. Then, *tf*idf* weight for a term j in document i is computed as in equation 8. In this method, we use raw term frequency values of terms in the documents as *tf* value.

$$w_{ij}=tf_{ij}\times idf_j \quad (8)$$

3.2.3. Feature Selection Methods

There are various algorithms to do feature selection on text mining. Purpose of feature selection is to remove useless terms for classification, therefore to increase accuracy. On the other hands, by removing irrelevant and useless features we decrease the size of datasets and also improve runtime performance of classifiers. In this study we propose a new feature selection method that is based on standard deviation of term frequencies, we also compare its performance with the well-known feature selection methods [19] that are information gain and chi square.

Proposed Feature Selection Method: Standard deviation is a frequently used method in the field of mathematics and statistics. It is used to calculate how the data are distributed according to the arithmetic mean. The standard deviation uses the mean of the distribution as a reference point and measures the distribution of the values by computing the average distances between each value and the mean of the values. It is calculated as shown in equation 9.

$$S=\sqrt{\frac{\sum_i^n (x_i-\bar{x})^2}{n}} \quad (9)$$

where S is the standard deviation for the variable x, $\bar{x}$ is the mean value of the variable x, x_i is the i^{th} value of the variable x, and n is the number of instances in the dataset.

In this study, our proposed feature selection process can be summarized as follows: for each feature extracted from the dataset, total frequencies of the terms (i.e., features) for each class in the dataset are computed. If the dataset has m classes, then for each class, frequency of each term is computed separately. Therefore, m frequency values for a term j are computed, as there are m classes. Then the standard deviation of these class-based frequencies is calculated by using equation 9 for each unique term extracted from the whole dataset. After that, terms are sorted in descending order with respect to their standard deviations, and terms in the top ranked 0.1, 0.5, 1, 5, 10, 15, 20 percent standard deviation or having standard deviation greater than 0 are selected; and used for the classification process.

In our proposed method we assume that, if a feature has high standard deviation with respect to class labels, it is highly probable that this feature is class-specific, therefore it should be useful for the classification process. We compare the performance of our proposed method with the well-known methods that are information gain and chi square feature selection methods that are summarized in the below subsections.

Information Gain Feature Selection Method: Information gain (IG) is widely used to identify the distinguishing features in the data set [19]. IG takes values between 0 and 1 and; if its value is close to 1, it means the feature is significant. The entropy must be calculated before the information gain is computed. Entropy shows the uncertainty and the possibility of unexpected occurrences. Entropy of a dataset D is computed as presented in equation 10 [19].

$$Entropy(D)=-\sum_{i=1}^{m} p_i \log_2 p_i \quad (10)$$

where p_i is the probability of any class i in the dataset, and m is the number of classes. If we assume that an attribute A of the dataset D has v distinct values $\{a_1, a_2, \ldots, a_v\}$, then we can divide the tuples in D into partitions by using the values of attribute A. If A is a discrete-valued attribute, we can directly split D into v partitions or subsets,

$\{D_1, D_2, \ldots, D_v\}$, where D_j contains those tuples in D in which attribute A has value a_j. Information requirement for attribute A can be computed according to equation 11.

$$Info_A(D)=\sum_{j=1}^{v} \frac{|D_j|}{|D|} \text{ x } Info(D_j) \quad (11)$$

where $Info(D_j)$ is the entropy of the j^{th} partition of the dataset D. Information gain of attribute A is defined as the difference between the original information requirement and the new requirement as in equation 12.

$$Gain(A)=Entropy(D)-Info_A(D) \quad (12)$$

In the IG feature selection method, information gain of all extracted attributes is computed, then the top ranked n attributes having the highest information gain, or attributes having information gain that are greater than a pre-specified threshold are selected to do classification [19].

Chi-Square Feature Selection Method: Chi-square (CHI2) statistic is used to compare results for two (or more) independent groups or categorical responses [19]. Let a discrete valued attribute A has v distinct values, namely $a_1, a_2, \ldots, a_v$ and class C has m distinct values, namely $c_1, c_2, \ldots, c_m$. The data tuples having value a_i for attribute A, and class label c_j for all i and j values are shown as a contingency table, with the v columns and the m rows. For each and every possible (a_i, c_j) combination, we have a cell in the contingency table. The chi square statistic for this contingency table is computed as in equation 13.

$$\sum_{i=1}^{v} \sum_{j=1}^{m} \frac{(o_{ij}-e_{ij})^2}{e_{ij}} \quad (13)$$

where o_{ij} is the observed frequency of the (a_i, c_j) pair, and e_{ij} is the expected frequency of (a_i, c_j) pair, which can be computed as in equation 14.

$$e_{ij}=\frac{count(A=a_i) \text{ x } count(C=c_j)}{n} \quad (14)$$

where n is the number of data tuples in D, count $(A=a_i)$ is the number of tuples having value a_i for attribute A, and count $(C=c_j)$ is the number of tuples having value c_j for class label. The sum in equation 13 is calculated over all of the (v x m) cells.

The attributes having high chi square values are selected for classification, as the actual count of these attributes are different from the expected values [10].

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

We use four different Turkish datasets to investigate the effects of preprocessing on the classification of Turkish texts. Firstly, we apply 5 different term weighting methods to the datasets to form document-term matrices. Then, stopwords are removed from documents, it is observed that there is little or no effect on classification accuracy when stopwords are removed, therefore stopwords are removed from the datasets to reduce dataset sizes for further experiments. After that Zemberek, Affix Stripping, Fixed Prefix 3, 5, 7 stemming algorithms are applied to the datasets to observe the effects of stemming and choose the best stemmer.

After the best stemmer is determined, effects of feature selection are investigated. We apply information gain, chi square, and our proposed standard deviation-based methods to all datasets. We select terms having information gain and chi square score values that are greater than zero, and we use the raw forms of the selected terms. To evaluate performance of our proposed feature selection method, we select features that have standard deviations greater than zero, and also we sort the features with respect to their standard deviations and select the top ranked 0.1, 0.5, 1, 5, 10, 15, and 20 percent features to make classification. As classifier we apply Naïve Bayes Multinomial (NBM) which is used for text classification [20]. We use Perl programming

language for implementing the weighting methods and our feature selector; Preto [21] for applying stopwords removal and stemming; and Weka [22] data mining tool for NBM classifier, IG and CHI2 feature selectors. In the below subsections results of each experiment are presented and discussed.

4.1. Effects of Term Weighting Methods

Five different term weighting methods that are tf, logtf, tp, normtf, and tf*idf are used to form document vectors. For testing, 10 folds cross-validation is applied. Classification results in terms of F-measure values of this experiment are shown in Table 2 where the best values are written in bold-face for each dataset.

Table 2. F-Measure values of the NBM classifier for different term weighting methods

	Term Weighting Methods				
Datasets	***tf***	***logtf***	***tp***	***normtf***	***tf*idf***
1150 News	**93.2**	64.3	92.1	92.9	92.9
3000 Tweet	**52.4**	27.2	52.2	51.3	48.7
Turkish Email	97.9	94.6	**98.1**	**98.1**	97.7
25 Author	51.5	21.7	57.1	46.7	**77.1**

As can be seen in Table 2, the best performance is achieved with tf for 1150 News and 3000 Tweet, tp and normtf for Turkish Email, tf*idf for 25 Author datasets. tf and tf*idf also have good performance for Turkish Email dataset. In fact, the difference between tf and tf*idf is much smaller than between other term weighting methods. According to the results in Table 2, we can conclude that as we have longer texts in the datasets tf*idf weighting method has better performance. If we have very short texts as in the 3000 Tweet dataset, tf or tp is successful. Therefore, in the subsequent experiments, tf and tf*idf methods are applied as they are the most successful two methods in majority of the datasets.

4.2. Effects of Stemming Methods

In this study, 3 different stemming algorithms (i.e., Zemberek, Affix Stripping, Fixed Prefix) are compared. Fixed Prefix with 3, 5, 7 term lengths are used in this study. The experimental results are presented in Table 3 and 4, where the best values are written in boldface.

In this experiment, first we remove stopwords, after that we apply stemming with Zemberek, Affix Stripping, and Fixed Prefix Stripping to extract terms (features) from the datasets. Then the most successful two term weighting methods that are tf and tf*idf are applied for all cases for computing document-term matrices. We also apply these term weighting methods for the case when no stemming is done. After that classification performances are compared.

In Table 3 and 4, where the experimental results are presented, “No Stem.”, “Z”, “AS”, “FP3”, “FP5”, and “FP7” abbreviations denote “No Stemming”, “Zemberek”, “Affix Stripping”, “Fixed Prefix 3”, “Fixed Prefix 5”, and “Fixed Prefix 7”, respectively. In Table 3, experimental results for 1150 News and 3000 Tweet datasets are presented. In Table 4, experimental results for Turkish Email, and 25 Author datasets are summarized.

Table 3. The classification results in terms of F-Measure values for Zemberek, Afix Stripping, Fixed Prefix 3,5,7 stemming algorithms for 1150 News and 3000 Tweet Datasets

	1150 News *tf*	**1150 News *tf*idf***	**3000 Tweet *tf***	**3000 Tweet *tf*idf***
No Stem.	**93.4**	**93.2**	**52**	47.7
Z	91.4	90.9	49.8	**48.7**
AS	90.7	91.6	48.2	**48.7**
FP3	64.8	68.3	34.8	37
FP5	88.6	88.5	46.5	46.7
FP7	91.8	91.9	50.1	47.6

Table 4. The classification results in terms of F-Measure values for Zemberek, Afix Stripping, Fixed Prefix 3,5,7 stemming algorithms for Turkish E-mail and 25 Author Datasets

	Turkish Email ***tf***	**Turkish Email** ***tf*idf***	**25 Author** ***tf***	**25 Author** ***tf*idf***
No Stem.	**98.6**	97.6	**57.3**	**80.5**
Z	97	97.1	52.2	68.3
AS	97.1	97.2	41.2	62.4
FP3	93.5	94.5	28.2	25.2
FP5	95.6	96.1	42.4	56.5
FP7	97.4	**97.9**	52.2	68.8

As shown in Table 3 and 4, in majority of the cases, stemming reduces classification accuracy as Turkish is an agglutinative language. Therefore, when we apply stemming, meaning of words may change and this fact negatively affects classification results. Among the stemming algorithms, Fixed Prefix7 gives the most successful results due to the fact that it generates longer stems that are similar to the original words. The Fixed Prefix3 algorithm gives the worst result among the stemming algorithms as the number of the stems generated are low, and stems are too short to be meaningful. Classification with raw forms of the terms is more successful than the stemmed terms. Therefore, in the subsequent experiments only the raw forms of the terms are used.

4.3. Effects of Feature Selection Methods

We investigate the effects of our proposed standard deviation-based feature selection method (SD) and compare it with information gain and chi square feature selection methods. The number of features before and after applying feature selection are presented in Table 5 for the 1150 News, and 3000 Tweet datasets; and in Table 6 number for features for Turkish Email, and 25 Author datasets are listed.

Table 5. Number of terms selected for 1150 News and 3000 Tweet datasets

	Datasets	
FS Methods	**1150 News**	**3000 Tweet**
No Selection (No FS)	44,804	10,664
Terms with SD>0	32,275	7,806
Top 20% SD terms	7,055	1,561
Top 15% SD terms	5,291	1,171
Top 10% SD terms	3,528	781
Top 5% SD terms	1,764	390
Top 1% SD terms	353	78
Top 0.5% SD terms	176	39
Top 0.1% SD terms	35	8
Information Gain	1890	144
Chi Square	1889	144

Table 6. Number of terms selected for Turkish Email and 25 Author datasets

	Datasets	
FS Methods	**Turkish Email**	**25 Author**
No Selection (No FS)	46,159	94,199
Terms with SD>0	33,063	70,467
Top 20% SD terms	6,613	14,093
Top 15% SD terms	4,959	10,570
Top 10% SD terms	3,306	7,047
Top 5% SD terms	1,653	3,523
Top 1% SD terms	331	705
Top 0.5% SD terms	165	352
Top 0.1% SD terms	33	70
Information Gain	4671	210
Chi Square	4671	208

As it can be easily seen from Table 5 and 6, applying feature selection reduces number of features very sharply, therefore also reduces the dataset size. As an example, according to Table 5, when no feature selection is applied the number of features to be used for classification is 44,804 and 10,664 for 1150 News and 3000 Tweet datasets, respectively. When our SD based feature selection method is applied and top 1% features are selected, the number of features to be used for classification is reduced to 353 and 78 for 1150 News, and 3000 Tweet datasets, respectively. Similar reductions in feature spaces are also observed for the Turkish Email, and 25 Author datasets as shown in Table 6.

First of all, we apply our proposed feature selection method to compute classification

accuracy of the datasets. During the experimental evaluation, we observed that applying feature selection when tf*idf weighting is used yields better classification accuracy. Therefore, in Table 7 classification F-measure values for tf*idf weighting when standard deviation-based feature selection is applied are presented. Classification results for tf weighting are not presented to save space.

As shown in Table 7, applying SD based feature selection improves classification accuracy for almost all datasets while reducing the number of features very sharply (see Table 5 and 6). For all datasets, the best results are observed when top scored 15% features are selected except the 1150 News dataset. For 1150 News dataset, using all features have better classification accuracy; however, selecting only 15% of features reduces accuracy by only 1.6% while reducing the feature size by 85%.

Table 7. Classification F-measure values for SD based feature selection for tf*idf weighting

	Datasets			
FS Method	**1150 News**	**3000 Tweet**	**Turkish Email**	**25 Author**
No FS	**93.2**	47.7	97.6	80.5
SD>0	93	47.3	97.1	81.7
%20	91.8	59.3	**98.4**	90.7
%15	91.6	**59.9**	**98.4**	**91.1**
%10	92.3	57	97.9	91
%5	92	54.1	97	90
%1	87.9	45.9	94.2	90
%0.5	79.8	43.4	94.6	78.3
%0.1	57.5	32.1	93.9	50.1

In Table 8, the best classification F-measure values of the SD based method are compared with the F-measure values obtained when information gain and chi square feature selection methods are applied. The classification results in Table 8 are computed by selecting the features having information gain or chi square values that are greater than zero. As the number of features having information gain or chi square values that are greater than zero is not high (see Table 5 and 6), we do not apply any further reduction in the feature space. We also use tf*idf weighting for this comparison.

Table 8. Best classification F-measure values of SD, information gain, and chi square methods for tf*idf weighting

Dataset	**No FS**	**SD**	**Info Gain**	**Chi square**
1150 News	**93.2**	93	92.3	92.3
3000 Tweet	47.7	**59.9**	44	44
Turkish Email	97.6	**98.4**	97.2	97.2
25 Author	80.5	**91.1**	40	40

According to the results given in Table 8, only for 1150 News dataset feature selection reduces classification performance slightly (i.e., 0.2% for SD, 0.9% for Information Gain and Chi Square methods), for all other datasets SD based feature selection positively affects classification performance. When the three feature selection methods are compared, it can be easily observed that information gain and chi square have almost the same performance as the number of features selected are also quite similar, and both methods have worse performance with respect to the proposed method. Our SD based method is very successful when the number of classes is high as in the 25 Author dataset.

We also investigate the effect of the feature selection on time required for classification, and the results in terms of seconds are given in Table 9.

As it can be easily seen from Table 9, when feature selection is applied, number of features used in the classification process is reduced therefore time required to train and test the classifier is also reduced sharply without decreasing the classification accuracy (see Table 8).

Table 9. Time required (in seconds) for classification when the best F-measure values obtained

	No FS		SD	
Dataset	**# of features**	**Time**	**# of features**	**Time**
1150 News	44,804	15	15,509	7
3000 Tweet	10,664	7	1,561	1
Turkish Email	46,159	8	33,063	7
25 Author	94,199	94	3,523	9

5. CONCLUSION

In this study, we investigated the effects of feature extraction techniques that include stemming, stopwords removal, term weighting, and feature selection on the accuracy of classification of Turkish documents from different problem domains. As the results of the experiments show, the most successful term weighting methods are tf and tf*idf methods. When feature selection method is used tf*idf method becomes the best weighting method in almost all problem domains. For very short texts, as in 3000 Tweet datasets; tp is also a good performer weighting method.

When the stemming methods are compared, it is observed that the classification of the documents using the raw forms of the terms gives more successful results. This result has occurred due to the fact that Turkish is an agglutinative language, and when stemming is applied, the meaning and polarity of the words may change which affect the classification process in a negative way. Therefore, more improvements are needed on the stemmers used for Turkish.

It is also observed that removing stopwords reduces number of features without decreasing classification accuracy. When the classification results of feature selection methods are compared, it is found that standard deviation-based feature selection has more improvement than information gain and chi square methods on classification accuracy especially when the number of classes is large. Also, feature selection reduces time required for the classification process.

As future work, our standard deviation-based feature selection method may be combined with a search method to find optimal number of features for different datasets.

6. REFERENCES

1. Hand, D., Mannila, H., Smyth, P., 2001. Principles of Data Mining, the MIT Press, England, 546.

2. İlhan, S., Duru, N., Karagöz, Ş., Sağır, M., 2008. Metin Madenciliği ile Soru Cevaplama Sistemi, ELECO-2008, 356-359.

3. Amasyalı, M.F., Diri, B., 2006. Automatic Turkish Text Categorization in Terms of Author, Genre and Gender. C. Kop et al. (Eds.): NLDB 2006, LNCS 3999, 221–226.

4. Yıldız, H.K., Gençtav, M., Usta N., Diri B., Amasyalı M.F., 2007. Metin Sınıflandırmada Yeni Özellik Çıkarımı, Signal Processing and Communications Applications (SIU 2007), Eskişehir, Turkey.

5. Cataltepe, Z., Turan, Y., Kesgin, F., 2007. Turkish Document Classification Using Shorter Roots, Signal Processing and Communications Applications (SIU 2007), Eskisehir, Turkey.

6. Güran, A., Akyokuş, S., Bayazıt, N.G., Gürbüz, M.Z., 2009. Turkish Text Categorization Using N-Gram Words. International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2009), Trabzon, Turkey.

7. Torunoğlu, D., Çakırman, E., Ganiz, M., Akyokuş, S., Gürbüz, Z., 2011. Analysis of Preprocessing Methods on Text Classification of Turkish Texts, International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), İstanbul, 112-117.

8. Uysal, K.U., Günal, S., 2013. The Impact of Preprocessing on Text Classification, Information Processing and Management, 104-112.

9. Amasyalı, M.F., Balcı, S., Varlı, E.N., Mete, E., 2012. Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması, EMO Bilimsel Dergi.

10. Açıkalın, B., Beyazıt, N.G., 2016. The Importance of Preprocessing in Turkish Text

Classification, Signal Processing and Communications Applications (SIU 2016), Zonguldak.
11. Parlar T., Özel S.A., 2018. An Investigation of Term Weighting and Feature Selection Methods for Sentiment Analysis, Majlesi Journal of Electrical Engineering, 12(2), 63-68.
12. Amasyalı, M.F., Beken, A., 2013. Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Sınıflandırmada Kullanılması, Signal Processing and Communications Applications (SIU 2009), Antalya, Turkey.
13. Amasyalı, M.F., Çetin, M., 2013. Eğiticili ve Geleneksel Terim Ağırlıklandırma Yöntemleriyle Duygu Analizi, Signal Processing and Communications Applications (SIU 2013), KKTC.
14. Ergin, S., Sora Gunal, E., Yigit, H., Aydin, R., 2012. Turkish Anti-spam Filtering Using Binary and Probabilistic Models, AWER Procedia Information Technology & Computer Science, 1, 1007-1012.
15. Yıldız Teknik Üniversitesi Kemik Grubu Veri Kümeleri, http://www.kemik.yildiz.edu.tr
16. Akın, A.A., Akın, M.D., 2007. Zemberek, an Open Source NLP Framework for Turkish Languages, Structure, 10, 1-5.
17. Eryiğit, G., Adalı, E., 2004. An Affix Striping Morphological Analyzer for Turkish, International Conference Artificial Intelligence and Applications, Austria, 299-304.
18. Can, F., Koçberber, S., Balçık, E., Kaynak, C., Öcalan, H.Ç., Vursavaş, O.M., 2008. Information Retrieval on Turkish Texts, Journal of the American Society for Information Science and Technology, 59, 407-421.
19. Han J., Kamber M., Pei, J.P., 2012. Data Mining Concepts and Techniques, Elsevier, 740.
20. Leung, K.M., 2007. Naive Bayesian Classifier, Polytechnic University Department of Computer Science/Finance and Risk Engineering. Lecture Notes.
21. Tunalı, V., Bilgin, T.T., 2012. PRETO: A High-Performance Text Mining Tool for Preprocessing Turkish Texts, International Conference on Computer Systems and Technologies.
22. Weka data mining tool http://www.cs.waikato.ac.nz/ml/weka