NEW APPROACHES TO ENHANCING THE PERFORMANCE OF TEXT CLASSIFICATION

Alper Kürşat UYSAL

Ph.D. Dissertation

Computer Engineering Program
March, 2013

This dissertation is supported in part by Anadolu University, Fund of Scientific Research Projects under grant no. 1103F054.

ANADOLU ÜNİVERSİTESİ

JÜRİ VE ENSTİTÜ ONAYI

Alper Kürşat UYSAL’ın **“New Approaches to Enhancing the Performance of Text Classification”** başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki Doktora Tezi 22.02.2013 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Yard. Doç. Dr. Serkan GÜNAL**	**………….**
Üye	**: Yard. Doç. Dr. Semih ERGİN**	**………….**
Üye	**: Doç. Dr. Hüseyin POLAT**	**………….**
Üye	**: Doç. Dr. Yusuf OYSAL**	**………….**
Üye	**: Prof. Dr. Ömer Nezih GEREK**	**………….**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu’nun ………………. tarih ve ……… sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ABSTRACT

Ph.D. Dissertation

**NEW APPROACHES TO
ENHANCING THE PERFORMANCE
OF TEXT CLASSIFICATION**

Alper Kürşat UYSAL

Anadolu University
Graduate School of Sciences
Computer Engineering Program

Supervisor: Assist. Prof. Dr. Serkan GÜNAL

2013, 88 pages

The aim of text classification, also known as text categorization, is to classify texts of interest into appropriate classes. Due to the rapid advance of Internet technologies, the amount of electronic documents has drastically increased worldwide. Consequently, text classification has gained importance in organization of these documents. Important issues in text classification are the high dimensionality of feature space and misclassification concerns regarding the feature space. In this dissertation, various solutions are proposed to overcome both of these concerns of the text classification problems. Specifically, a novel filter-based feature selection method, namely distinguishing feature selector, is introduced. Besides, genetic algorithm oriented latent semantic features, which are originated from feature selection and transformation operations, are proposed. Moreover, the impact of several feature extraction and selection approaches on SMS spam filtering problem, a special case of text classification, is extensively investigated for two different languages. Finally, the impact of preprocessing methods on text classification is examined for different domains and different languages as well. Extensive experiments conducted on benchmark datasets revealed that all the proposed solutions offer better dimensionality reduction and/or classification performance depending on their contributions.

Keywords: Text Classification, Feature Extraction, Feature Selection,
Feature Transformation.

ÖZET

Doktora Tezi

METİN SINIFLANDIRMA BAŞARIMINI İYİLEŞTİRMEK İÇİN YENİ YAKLAŞIMLAR

Alper Kürşat UYSAL

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Serkan GÜNAL
2013, 88 sayfa

Metinlerin kategorize edilmesi olarak da bilinen metin sınıflandırmanın amacı metinleri uygun sınıflara atamaktır. İnternet teknolojilerinin hızlı bir şekilde gelişmesine bağlı olarak dünya genelindeki elektronik belge miktarında yüksek miktarda bir artış görülmüştür. Dolayısıyla metin sınıflandırma, bu belgelerin organizasyonunda büyük bir önem kazanmıştır. Metin sınıflandırmadaki önemli sorunlar öznitelik uzayının yüksek boyutluluğu ve bundan kaynaklı hatalı sınıflandırmalardır. Bu tez çalışmasında, metin sınıflandırmadaki bu iki sorunun üstesinden gelebilmek için çeşitli çözümler önerilmiştir. Özel olarak, ayırt edici öznitelik seçici adında yeni bir filtre tabanlı öznitelik seçim yöntemi ortaya çıkarılmıştır. Bunun yanı sıra, öznitelik seçim ve öznitelik dönüşüm işlemlerinden oluşan genetik algoritma yönelimli gizli anlamsal öznitelikler önerilmiştir. Ayrıca, çeşitli öznitelik çıkarım ve öznitelik seçim yöntemlerinin metin sınıflandırmanın bir türü olan istenmeyen kısa mesaj filtreleme problemi üzerindeki etkisi iki farklı dil için detaylı bir şekilde araştırılmıştır. Son olarak, ön işleme yöntemlerinin metin sınıflandırma üzerinde etkisi farklı konu başlıkları ve farklı diller için incelenmiştir. Kıyaslama veri kümeleri üzerinde yapılan kapsamlı deneyler, önerilen tüm çözümlerin daha iyi boyut indirgeme ve/veya sınıflandırma başarımı sağladığını ortaya koymuştur.

Anahtar Kelimeler: Metin Sınıflandırma, Öznitelik Çıkarımı, Öznitelik Seçimi, Öznitelik Dönüşümü.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Assist. Prof. Dr. Serkan Günal for his support and guidance to my research. I have taken his great amount of time and effort during research studies up-to-now.

I would also like to thank Prof. Dr. Ömer Nezih Gerek, Assoc. Prof. Dr. Hüseyin Polat, Assoc. Prof. Dr. Yusuf Oysal, and Assist. Prof. Dr. Semih Ergin for serving on my dissertation committee and for their valuable contributions.

A heartful thanks go to my parents. I am very grateful for their devotion and efforts on me.

Finally, I would like to thank my wife for her substantial support and patience during all my studies.

Alper Kürşat Uysal
March, 2013

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

BoW	:	Bag of Words
CHI2	:	Chi Square
CVA	:	Common Vector Approach
DFS	:	Distinguishing Feature Selector
DP	:	Deviation from Poisson Distribution
DR	:	Dimension Reduction
DT	:	Decision Tree
ETSI	:	European Telecommunications Standards Institute
GA	:	Genetic Algorithm
GALSF	:	Genetic Algorithm Oriented Latent Semantic Features
GI	:	Gini Index
IDF	:	Inverse Document Frequency
IG	:	Information Gain
kNN	:	k Nearest Neighbor
LDA	:	Linear Discriminant Analysis
LSI	:	Latent Semantic Indexing
NB	:	Naïve Bayes
NN	:	Neural Network
SF	:	Structural Feature
SMS	:	Short Message Service
SVD	:	Singular Value Decomposition
SVM	:	Support Vector Machine
TF	:	Term Frequency
TF-IDF	:	Term Frequency Inverse Document Frequency

1. INTRODUCTION

The fundamental goal of the text classification, which is also known as text categorization, is to classify texts of interest into appropriate classes (Gunal, 2012; Uysal and Gunal, 2012). With rapid advance of Internet technologies, the amount of electronic documents has drastically increased worldwide. As a consequence, text classification has gained importance in hierarchical organization of these documents. So far, text classification has been successfully applied to various domains such as topic detection (Ghiassi et al., 2012), spam e-mail filtering (Gunal et al., 2006; Lopes et al., 2011), SMS spam filtering (Delany et al., 2012; Uysal et al., 2012b), author identification (Cheng et al., 2011), web page classification (Golub, 2006; Ozel, 2011) and sentiment analysis (Na and Thet, 2009; Maks and Vossen, 2012).

A typical text classification framework, just like other pattern classification systems, consists of preprocessing, feature extraction, feature selection, and classification stages. Feature transformation stage can also be adapted to this framework as a separate or parallel stage. Structure of this framework is visualized in Figure 1.1.

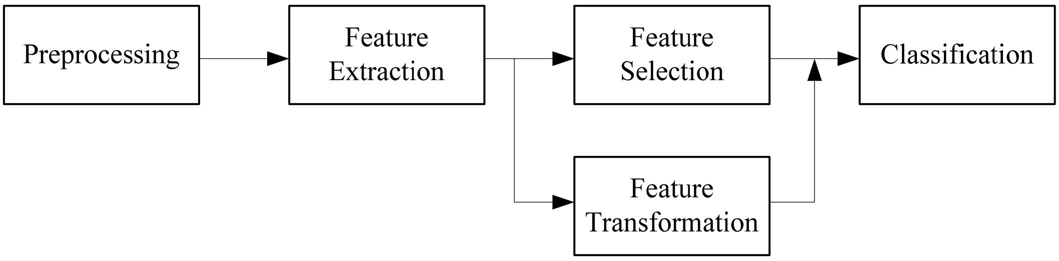


Figure 1.1. Structure of a text classification framework.

Preprocessing stage is one of the key components in a typical text classification framework. The aim of the preprocessing step is to prepare raw text for the feature extraction stage by applying certain language-dependent and language-independent preprocessing algorithms, which will be explained in Section 2.

Feature extraction stage extracts numerical information from raw text documents by considering unique term frequencies. At the end of this, text documents are represented with numeric values, namely feature vectors. However, due to the nature of text classification, there is a big amount of features extracted in this stage. Excessive numbers of features not only increase computational time but also degrade classification accuracy. Because of this, a feature selection stage is a necessity as a subsequent process.

One of the most important issues in text classification is dealing with high dimensionality of the feature space. Therefore, feature selection is also an essential topic for text classification. Its main aim is to decrease feature dimension by removing irrelevant features from the feature set. As a consequence, feature selection plays a critical role in text classification regarding speeding up the computation as well as improving the accuracy.

Moreover, feature transformation approaches can also be used to reduce feature dimension. The difference between feature selection and feature transformation is that feature transformation reduces the dimension by projecting the original feature space into a new lower-dimensional subspace rather than selecting from the original set of features. Its main aim is to obtain a better representation of data. Feature transformation approaches can be applied individually or in conjunction with feature selection.

In classification step, a classifier carries out the classification process using a prior knowledge of labeled data, and documents are classified into appropriate classes. As electronic documents are represented with numeric values, any classifier used in pattern recognition problems can be integrated to text classification process. However, selection of appropriate classifier increases success ratio of classification.

1.1. Problems in Text Classification

One of the most important issues in text classification is the high dimensionality of feature space. While some of the features are discriminative, many of them are not. These irrelevant features not only degrade the performance of text classification but also increase running time. Therefore, the aim is to obtain a feature set including only discriminative features. This can be realized by altering preprocessing, feature extraction, and feature selection stages in text classification. While using appropriate preprocessing and feature extraction stages may help obtain an ideal feature set, an improved feature selection stage may also lead to selection of more discriminative features among all. Besides, feature transformation stages may provide better representation of data in reduced dimensions.

Another significant issue is to obtain classification process as accurate and precise as possible. Choosing an appropriate classifier in this stage may improve performance of text classification; on the other hand, employing inappropriate classifiers may degrade the performance.

1.2. Contributions

In this dissertation, various solutions are proposed to the problems mentioned in the previous subsection. The contributions are specifically the answers to the research questions below:

i. *How do we select more discriminative features to obtain improved dimension reduction and accuracy for text classification?*

ii. *How can we obtain a better representation of text data to improve performance of text classification and provide dimension reduction?*

iii. *What are the ideal feature extraction and selection strategies to improve accuracy of text classification?*

iv. *What is the appropriate combination of preprocessing methods enhancing the accuracy of text classification?*

Considering the abovementioned questions, the first contribution of the dissertation is a novel filter-based probabilistic feature selection method, namely distinguishing feature selector (DFS), for text classification. The proposed method is compared with well-known filter approaches including chi square, information gain, Gini index and deviation from Poisson distribution. The comparison is carried out for different datasets, classification algorithms, and success measures. Experimental results explicitly indicate that DFS offers a competitive performance with respect to the abovementioned approaches in terms of classification accuracy, dimension reduction rate and processing time.

As the second contribution, genetic algorithm oriented latent semantic features (GALSF) are proposed for text classification. The proposed method consists of two stages, namely feature selection and feature transformation. The feature selection stage is carried out using the state-of-the-art filter-based methods. The feature transformation stage employs latent semantic indexing (LSI) empowered by genetic algorithm (GA) such that a better projection is attained using appropriate singular vectors, which are not limited to the ones corresponding to the largest singular values, unlike standard LSI approach. In this way, the singular vectors with small singular values may also be used for projection whereas the vectors with large singular values may be eliminated as well to obtain better discrimination. Effectiveness of the proposed method is comparatively evaluated against feature selection, and the combination of feature selection and transformation on two-class and multi-class text collections. For both collections, GALSF surpasses the other methods in terms of classification performance. In the meantime, GALSF offers reasonable dimension reduction performance.

The third contribution is related to the impact of feature extraction and selection methods on text classification. Specifically, this dissertation investigates the impact of several feature extraction and selection approaches on Short Message Service (SMS) spam filtering problem in two different languages, namely Turkish and English. The entire feature set of filtering framework consists of the features originated from the bag-of-words (BoW) model along with the ensemble of structural features (SF) specific to spam problem. The distinctive BoW features are identified using information theoretic feature selection methods. Various combinations of the BoW and SF are then fed into widely used pattern classification algorithms to classify SMS messages. The filtering framework is evaluated on both Turkish and English SMS message datasets. Comprehensive experimental analysis on the respective datasets revealed that the combinations of BoW and SFs, rather than BoW features alone, provide better classification accuracy on both datasets.

The last contribution of this dissertation is an extensive examination of the impact of preprocessing tasks on text classification in terms of various aspects such as classification accuracy, text domain, text language, and dimension reduction. For this purpose, all possible combinations of widely used preprocessing tasks are comparatively evaluated on two different domains, namely e-mail and news, and in two different languages, namely Turkish and English. In this way, contribution of the preprocessing tasks to classification success at various feature dimensions, possible interactions among these tasks, and also dependency of these tasks to the respective languages and domains are comprehensively assessed. Experimental analysis on benchmark datasets reveals that choosing appropriate combinations of preprocessing tasks, rather than enabling or disabling them all, may provide significant improvement on classification accuracy depending on the domain and language studied on.

1.3. Organization of the Dissertation

The components of text classification framework are explained in Section 2. This section covers preprocessing, feature extraction, feature selection, feature transformation, and classification stages in text classification. Following this section, four consecutive sections are reserved for the contributions of this dissertation. The novel feature selection method, DFS, and the related experiments are presented in Section 3. In Section 4, genetic algorithm oriented latent semantic features, and the corresponding experimental analysis are presented. Section 5 presents the study of the impact of feature extraction and selection on SMS spam filtering. The impact of preprocessing on text classification is provided in Section 6. Finally, overall concluding remarks of the dissertation and potential future works are discussed in Section 7.

2. COMPONENTS OF TEXT CLASSIFICATION FRAMEWORK

The stages of a typical text classification framework were presented in the previous section. In this section, the details of these stages and related methodologies used in this dissertation are explained.

2.1. Preprocessing

Widely used preprocessing steps in text classification are tokenization, stop-word removal, lowercase conversion, and stemming. Each step is explained in the following subsections.

2.1.1. Tokenization

In text processing, tokenization is the procedure of splitting a text into words, phrases, or other meaningful parts, namely tokens. In other words, tokenization is a form of text segmentation. Typically, the segmentation is carried out considering only alphabetic or alphanumeric characters that are delimited by non-alphanumeric characters (e.g., punctuations, whitespace). Tokenization of texts belonging to different languages may vary (Manning et al., 2008). While removing non-ASCII characters can be enough to tokenize documents in English language, this may not be enough for documents in Turkish language. Character sets of these languages are not same, and some Turkish characters cannot be expressed with ASCII character set. This condition is handled in different ways such as replacing non-ASCII characters with their ASCII counterparts for texts in Turkish language (Kucukyilmaz et al., 2006). While these types of approaches can be helpful for some tasks, preserving the original form of tokens can be a necessity for many cases. Table 2.1 shows an example to tokenization of a simple sentence in English.

Table 2.1. An example to tokenization

Language	**Sentence**	**Tokens (Separated with commas)**
English	I want the money.	I, want, the, money

2.1.2. Stop-word Removal

Stop-words are the words that are commonly encountered in texts without dependency to a particular topic (e.g., conjunctions, prepositions, articles, etc.).

Therefore, the stop-words are usually assumed to be irrelevant in text classification studies and removed prior to the classification. While the stop-words are specific to the language being studied as in the case of stemming, there is not a definite list of stop-words in any language. Sample stop-words are provided in Table 2.2 for Turkish and English languages.

Table 2.2. Sample stopword list

Language	Stop-words
Turkish	ama, ancak, bile, böyle, dolayısıyla, her, ki, kim, olmak, sadece, ve, zaten
English	a, able, about, above, according, across, actually, after, are, at, before, then

2.1.3. Lowercase Conversion

Another widely used preprocessing step for text classification is the lowercase conversion. Since uppercase or lowercase forms of words are assumed to have no difference, all uppercase characters are usually converted to their lowercase forms prior to the classification. Lowercase conversion reduces the total number of extracted features by grouping near-identical words. Usage of lowercase conversion can also vary in some cases related with characteristics of languages. Table 2.3 provides examples to both cases that lowercased versions of the same characters are different and same for Turkish and English languages, respectively.

Table 2.3. Lowercased forms of some characters for different languages

Original form	Lowercased (Turkish)	Lowercased (English)
I	ı	i
U	u	u

2.1.4. Stemming

The aim of stemming is to obtain stem or root forms of derived words. Since derived words are semantically similar to their root forms, word occurrences are usually computed after applying the stemming on a given text. Stemming algorithms are indeed specific to the language being studied. Though there are different approaches (Can et al., 2008; Zemberek, 2013), the fixed-prefix algorithm is computationally simple but very effective stemming tool for Turkish language (Can et al., 2008). On the other hand, the stemming algorithm introduced in (Porter, 1980) is commonly employed by researchers for English. An example to stemming is presented in Table 2.4 for a word having the same meaning in both Turkish and English languages, respectively.

Table 2.4. Stemming: An example

Language	Original word form	Stemmed word
Turkish	gösterir	göster
English	shows	show

2.2. Feature Extraction

Majority of text classification studies utilizes the BoWs technique to represent a document such that the order of terms within the document is ignored but frequencies of the terms are considered (Joachims, 1997; Gunal, 2012; Uysal and Gunal, 2012). Each distinct term in a document collection therefore constitutes an individual feature. Hence, a document is represented by a multi-dimensional feature vector. Representation of documents as feature vectors in such a way is known as the vector space model (Salton et al., 1975). In the feature vectors, each dimension corresponds to a weighted value of the regarding term within the document collection. There are different approaches in the literature to obtain weighted values. The widely used weighting approaches are explained in the following subsection.

2.2.1. Feature Weighting

As mentioned before, a document is represented by a multi-dimensional feature vector, where each dimension corresponds to a weighted value of the regarding term. For this purpose, there are various weighting approaches such as term frequency (*TF*), term frequency inverse document frequency (*TF-IDF*), some other approaches using combination of term frequency and feature selection scores (Liu et al., 2009b). However, *TF* and *TF-IDF* are widely-known feature weighting approaches for text classification. Formula of *TF* for term *t* and document *d* is as follows:

$$TF(t,d) = \begin{cases} occurrence\,count\,of\,term\;t\;in\,document\;d, & if\;term\;t\;occurs\,in\,document\,d \\ 0, & otherwise \end{cases} \tag{2.1}$$

The second feature weighting approach is *TF-IDF*. Formula of *TF-IDF* requires calculation of inverse document frequency. Inverse document frequency can be formulated for term *t* as follows:

$$IDF(t) = \log(D/docs), \tag{2.2}$$

where *D* is the number of documents in the collection and *docs* is the count of a subset of documents in which term *t* occurs. *TF-IDF* value is obtained by multiplying term frequency (*TF*) and inverse document frequency (*IDF*) as in Eq. (2.3).

$$TF - IDF = TF(t,d) \times IDF(t) \tag{2.3}$$

Briefly, the value of *TF-IDF* is the highest when term *t* occurs frequently within a small number of documents. On contrary, it is the smallest when term *t* occurs nearly in all documents (Manning et al., 2008). Terms occurring in all documents are generally stop-words or any kind of words having smaller discriminative power.

2.3. Feature Selection

Feature selection techniques broadly fall into three categories: filters, wrappers, and embedded methods. Filters assess feature relevancies using various scoring frameworks that are independent from a learning model or classifier, and select top-*N* features attaining the highest scores (Guyon and Elisseeff, 2003). Filter techniques are computationally fast; however, they usually do not take feature dependencies into consideration. On the other hand, wrappers evaluate features using a specific learning model and search algorithm (Kohavi and John, 1997; Gunal et al., 2009). Wrapper techniques consider feature dependencies, provide interaction between feature subset search, and choice of the learning model; but they are computationally expensive with respect to the filters. Embedded methods integrate feature selection into classifier training phase; therefore, these methods are specific to the utilized learning model just like the wrappers. Nevertheless, they are computationally less intensive than the wrappers (Guyon and Elisseeff, 2003; Saeys et al., 2007).

In text classification studies, although there are some hybrid approaches combining the filters and wrappers (Uguz, 2011; Gunal, 2012), commonly preferred feature selection methods are the filters due to their relatively low processing time. Term strength (Yang, 1995), odds ratio (Mladenic and Grobelnik, 2003), document frequency (Yang and Pedersen, 1997), mutual information (Liu et al., 2009a), chi-square (Chen and Chen, 2011), information gain (Lee and Lee, 2006), improved Gini index (Shang et al., 2007), measure of deviation from Poisson distribution (Ogura et al., 2009), minimum class difference (Chen and Lu, 2006), a support vector machine (SVM) based feature selection algorithm (Wu et al., 2007), ambiguity measure (Mengle and Goharian, 2009), class discriminating measure (Chen et al., 2009), and binomial hypothesis testing (Yang et al., 2011) are just some examples to the filter methods. Combinations of the features, which are selected by different filter methods, are also considered, and their contributions to the classification accuracy under varying conditions are investigated in (Gunal, 2012).

There is a mass amount of filter-based techniques for the selection of distinctive features in text classification. Among all those techniques, chi square, information gain, Gini index, and deviation from Poisson distribution have been proven to be much more effective (Yang and Pedersen, 1997; Shang et al., 2007; Ogura et al., 2009). Mathematical backgrounds of the current widely-used approaches are provided in the following subsections.

2.3.1. Chi-square

One of the most popular feature selection approaches is CHI2. In statistics, the CHI2 test is used to examine independence of two events. The events, X and Y, are assumed to be independent if

$$p(XY) = p(X)p(Y) \tag{2.4}$$

In text feature selection, these two events correspond to occurrence of particular term and class, respectively. CHI2 information can be computed using

$$CHI2(t,C) = \sum_{t\in\{0,1\}} \sum_{C\in\{0,1\}} \frac{(N_{t,C} - E_{t,C})^2}{E_{t,C}} \tag{2.5}$$

where N is the observed frequency and E is the expected frequency for each state of term t and class C (Manning et al., 2008). CHI2 is a measure of how much expected counts E and observed counts N deviate from each other. A high value of CHI2 indicates that the hypothesis of independence is not correct. If the two events are dependent, then the occurrence of the term makes the occurrence of the class more likely. Consequently, the regarding term is relevant as a feature. CHI2 score of a term is calculated for individual classes. This score can be globalized over all classes in two ways. The first way is to compute the weighted average score for all classes while the second way is to choose the maximum score among all classes. These approaches are formulated as Eq. (2.6) and Eq. (2.7) respectively.

$$CHI2(t) = \sum_{i=1}^{M} P(C_i).CHI2(t, C_i) \tag{2.6}$$

$$CHI2(t) = \max \sum_{i=1}^{M} (CHI2(t, C_i)), \tag{2.7}$$

where $P(C_i)$ is the probability of i*th* class, $CHI2(t, C_i)$ is the class specific CHI2 score of term *t*, and *M* is the number of classes.

2.3.2. Information Gain

IG measures how much information the presence or absence of a term contributes to make the correct classification decision on any class (Forman, 2003). IG reaches its maximum value if a term is an ideal indicator for class association, that is, if the term is present in a document if and only if the document belongs to the respective class. IG for term *t* can be obtained using

$$IG(t) = -\sum_{i=1}^{M} P(C_i) \log P(C_i) + P(t) \sum_{i=1}^{M} P(C_i \mid t) \log P(C_i \mid t) + P(\bar{t}) \sum_{i=1}^{M} P(C_i \mid \bar{t}) \log P(C_i \mid \bar{t}), \tag{2.8}$$

where *M* is the number of classes, $P(C_i)$ is the probability of class *C_i*, $P(t)$ and $P(\bar{t})$ are the probabilities of presence and absence of term *t*, $P(C_i \mid t)$ and $P(C_i \mid \bar{t})$ are the conditional probabilities of class C_i given presence and absence of term *t*, respectively.

2.3.3. Gini Index

GI is another feature selection method, which is an improved version of the method originally used to find the best split of attributes in decision trees (Shang et al., 2007). It has simpler computation than the other methods in general (Ogura et al., 2009). Its formulation is given as

$$GI(t) = \sum_{i=1}^{M} P(t \mid C_i)^2 P(C_i \mid t)^2, \tag{2.9}$$

where $P(t \mid C_i)$ is the probability of term *t* given presence of class C_i, $P(C_i \mid t)$ is the probability of class C_i given presence of term *t*, respectively.

2.3.4. Deviation from Poisson Distribution

DP is derived from Poisson distribution, which is also applied to information retrieval for selecting effective query words and this metric is adapted to feature selection problem to construct a new metric (Ogura et al., 2009). The degree of DP is used as a measure of effectiveness. If a feature fits into Poisson distribution, the result of this metric would be smaller and this indicates that the feature is independent from the given class. Conversely, the feature would be more discriminative if the result of the metric is greater. This method can be formulated as

$$
\begin{aligned}
&DP(t,C)=\frac{(a-\hat{a})^2}{\hat{a}}+\frac{(b-\hat{b})^2}{\hat{b}}+\frac{(c-\hat{c})^2}{\hat{c}}+\frac{(d-\hat{d})^2}{\hat{d}}\\
&\hat{a}=n(C)\{1-\exp(-\lambda)\}\\
&\hat{b}=n(C)\exp(-\lambda)\\
&\hat{c}=n(\bar{C})\{1-\exp(-\lambda)\}\\
&\hat{d}=n(\bar{C})\exp(-\lambda)\\
&\lambda=\frac{F}{N},
\end{aligned}
\tag{2.10}
$$

where F is the total frequency of term t in all documents, N is the number of documents in the training set, $n(C)$ and $n(\bar{C})$ are the numbers of documents belonging to class C and not belonging to class C, respectively, λ is the expected frequency of the term t in a document. The quantities a and b represent the number of documents containing and not containing term t in documents of class C, respectively. While the quantity c represents the number of documents containing term t and not belonging to class C, the quantity d represents the number of documents with absence of term t and class C at the same time. Furthermore, the quantities $\hat{a}$, $\hat{b}$, $\hat{c}$ and $\hat{d}$ are predicted values for a, b, c, d, respectively. In order to globalize class specific scores over the entire collection, the weighted average scoring (Ogura et al., 2009) is used as given below.

$$
DP(t)=\sum_{i=1}^{M}P(C_i).DP(t,C_i) \tag{2.11}
$$

2.4. Feature Transformation

In addition to feature extraction and selection, feature transformation approaches are also used to reduce feature dimension. However, these approaches project the original feature space into a new lower-dimensional subspace rather than selecting from the original set of features. Although there exist many feature transformation methods such as LSI and linear discriminant analysis (LDA), majority of the text classification studies prefer LSI due to its proven performance (Yu et al., 2008; Wang and Yu, 2009; Yang et al., 2009; Meng et al., 2011; Zhang et al., 2011). The underlying idea in LSI is to obtain the projection directions (i.e., singular vectors, eigenvectors, or principal components) providing the largest variations (i.e., largest singular values or eigenvalues) based on singular value decomposition (SVD) or principal component analysis (PCA) so that feature dimension is greatly reduced while keeping the discriminative information (Gud and Shatovska, 2009). LSI is explained in the next subsection in details.

2.4.1. Latent Semantic Indexing

LSI is known as one of the most representative feature transformation approaches which transforms the original data to a more discriminative lower-dimensional subspace (Liu et al., 2004). Although LSI is originated from information retrieval, it is widely used in text classification problems, as well. There exist various studies showing efficiency of LSI in both information retrieval (Kontostathis and Pottenger, 2006; Alhabashneh et al., 2011) and text classification (Yang and King, 2009; Meng et al., 2011). The success of LSI in text classification depends on its capability to reveal some underlying hidden concepts such as synonym and polysemy while projecting term-document matrix into a new subspace (Meng et al., 2011). Suppose that the document collection is represented as a term–document matrix *M*, which is $t \times n$ where t represents unique terms and n represents number of documents. Then, SVD of *M* can be defined as

$$M = U \sum V^T, \tag{2.12}$$

where $\sum$ is a diagonal matrix composed of the sorted singular values, *U* and *V* are the left and right singular vectors which are also term and document vectors, respectively. For dimension reduction, the largest s singular values and corresponding left and right singular vectors are used. The rank *s* approximation of *M* can be expressed as

$$M_s = U_s \Sigma_s V_s^T . \tag{2.13}$$

In this phase, LSI reveals hidden concepts such as synonym and polysemy. Therefore, M_s approximation of *M* represents data better than the original one. After this step, every document in all collection can be projected using the vector U_s as

$$doc_{projected} = doc_{original}{}^T . U_s , \tag{2.14}$$

where $doc_{original}$ is the original representation of the document with the initial feature size and $doc_{projected}$ is the *s*-dimensional projection of the original document.

2.5. Classification

Extracted features are fed into a pattern classifier to carry out the classification process. Some of the widely-used classifiers in text classification research are SVM, decision tree (DT), neural network (NN), and k-nearest neighbor (*k*NN) classifiers. All these classification methods are proven to be significantly successful in text classification (Drucker et al., 1999; Johnson et al., 2002; Yu and Zhu, 2009; Kumar and Gopal, 2010; Uguz, 2011; Gunal, 2012). Mathematical backgrounds of these classifiers are explained in the following subsections.

2.5.1. Support Vector Machine Classifier

SVM is one of the most effective classification algorithms in the literature. SVM algorithm has both linear and nonlinear versions. Linear version of SVM is generally preferred in many studies because of its speed and accuracy for text classification (Dumais et al., 1998; Kumar and Gopal, 2010). The essential point of SVM classifier is the notion of the margin (Joachims, 1998; Theodoridis and Koutroumbas, 2008). Classifiers utilize hyperplanes to separate classes. Every hyperplane is characterized by its direction (w) and its exact position in space (w_0). Thus, a linear classifier can be simply defined as

$$w^T x + w_0 = 0 \tag{2.15}$$

Then, the region between the hyperplanes $w^T x + w_0 = 1$ and $w^T x + w_0 = -1$, which separates two classes, is called as the margin. Width of the margin is equal to $2 / \|w\|$. Achieving the maximum possible margin is the underlying idea of SVM algorithm. Maximization of the margin requires minimization of

$$J(w, w_0, \varepsilon) = \frac{1}{2}\|w\|^2 + K\sum_{i=1}^{N} \varepsilon_i \tag{2.16}$$

which is subject to

$$\begin{aligned} &w^T x_i + w_0 \geq 1 - \varepsilon_i, \quad \text{if} \quad x_i \in c_1 \\ &w^T x_i + w_0 \leq -1 + \varepsilon_i, \quad \text{if} \quad x_i \in c_2 \\ &\varepsilon_i \geq 0. \end{aligned} \tag{2.17}$$

In Eq. (2.16), K is a user defined constant, and ε is the margin error. Margin error occurs if data belonging to one class is located on the wrong side of the hyperplane. Minimizing the cost is therefore a trade-off issue between a large margin and a small number of margin errors. Solution of this optimization problem is obtained as

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i \tag{2.18}$$

which is the weighted average of the training features. Here, λ_i is a Lagrange multiplier of the optimization task and y_i is a class label. Values of λs are nonzero for all the points lying inside the margin and on the correct side of the classifier. These points are known as support vectors and the resulting classifier as the support vector machine.

In case of multi-class classification problems, one of two common approaches, namely one-against-all and one-against-one, can be preferred to adopt two-class classification to multi-class case (Hsu and Lin, 2002).

2.5.2. Decision Tree Classifier

Decision or classification trees are multi-stage decision systems in which classes are consecutively rejected until an accepted class is reached (Theodoridis and Koutroumbas, 2008). For this purpose, feature space is split into unique regions corresponding to the classes. The most commonly used type of decision trees is binary classification tree that splits the feature space into two parts sequentially by comparing feature values with a specific threshold. Thus, an unknown feature vector is assigned to a class via a sequence of Yes/No decisions along a path of nodes of a decision tree. One has to consider splitting criterion, stop-splitting rule, and class assignment rule in design of a classification tree.

The fundamental aim of splitting feature space is to generate subsets that are more class homogeneous compared to former subsets. In other words, the splitting criterion at any node is to obtain the split providing the highest decrease in node impurity. Entropy is one of the widely used information to define impurity and it can be computed as follows:

$$I(t) = -\sum_{i=1}^{M} P(C_i \mid t) \log_2 P(C_i \mid t), \tag{2.19}$$

where $P(C_i \mid t)$ denotes the probability that a vector in the subset X_t , associated with a node *t*, belongs to class C, i = 1, 2,…, *M*. Assume now that performing a split, N_{tY} points are sent into "Yes" node (X_{tY}) and N_{tN} into "No" node (X_{tN}). The decrease in node impurity is then defined as follows:

$$\Delta I(t) = I(t) - \frac{N_{tY}}{N_t} I(t_{YES}) - \frac{N_{tN}}{N_t} I(t_{NO}) \tag{2.20}$$

where $I(t_{YES})$ and $I(t_{NO})$ are the impurities of the t_{YES} and t_{NO} nodes, respectively. If the highest decrease in node impurity is less than a certain threshold or a single class is obtained following a split, then splitting process is stopped. Once a node is declared to be terminal or leaf, then a class assignment is made. A commonly used assignment method is the majority rule that assigns a leaf to a class to which the majority of the vectors in the corresponding subset belong.

2.5.3. *k* Nearest Neighbor Classifier

kNN algorithm classifies feature vectors based on the closest training examples in the feature space (Theodoridis and Koutroumbas, 2008). More specifically, an unknown feature vector is assigned to the class that is the most common amongst its *k* nearest neighbors, where *k* is a positive integer. The value of *k* is determined empirically, e.g., it may be optimized with respect to the classification error on training dataset. In addition, Euclidean distance is generally preferred to measure the distance between feature vector and its neighbors. In the special case that *k* = 1, the feature vector is simply assigned to the class of its nearest neighbor.

2.5.4. Neural Network Classifier

Neural networks are mathematical models inspired by biological neural networks. One of the widely used application fields of neural networks are pattern recognition problems (Fausett, 1994). While some neural networks such as perceptron is known to be successful for linear classification problems, multi-layer neural networks can solve both linear and non-linear classification problems. A neural network consists of neurons, which are very simple processing elements and connected to each other with weighted links. Multi-layer neural networks consist of input, output, and hidden layer(s). While one hidden layer is sufficient for many cases, using two hidden layers may increase performance in some situations (Fausett, 1994). A simple multi-layer feed-forward neural network is shown in Figure 2.1, where *n* represents the dimension of input vector and *m* represents the number of outputs.

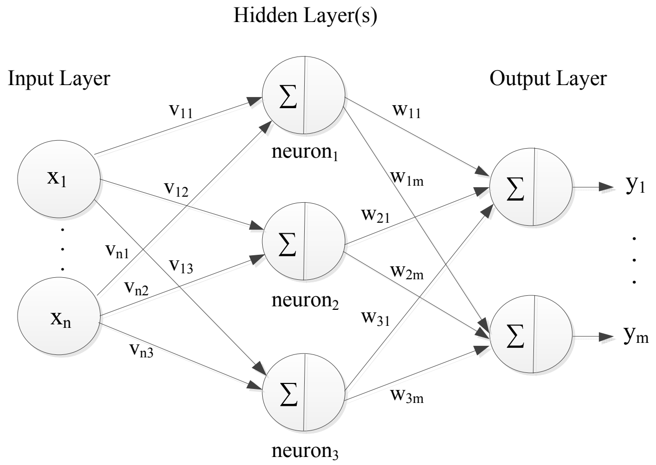


Figure 2.1. A simple multi-layer feed-forward neural network

Back-propagation is one of the most popular training methods for multi-layer feed forward neural networks. Training with back-propagation has three stages given as below:

i. The feed-forward of input training pattern

ii. The back-propagation of error

iii. The adjustment of weights

For the first stage, the net inputs to neurons need to be calculated and some summation and multiplication operations are performed as shown in Eq. (2.21). This formula simulates the calculation of the net input for *neuron$_1$* in Figure 6.1.

$$y\_neuron_1 = v_{11}x_1 + v_{21}x_2 + ... + v_{n1}x_n \tag{2.21}$$

After calculation of the net inputs, transfer functions are used to compute each neurons output from the net inputs. Some examples of the transfer functions are linear, logarithmic sigmoid, and tangent sigmoid transfer functions. There exist many variants of back-propagation training such as Levenberg-Marquardt, gradient descent, and gradient descent with momentum and adaptive learning rate. Following the first stage, the second and the third stages are carried out respectively. These operations are repeated until a predefined stopping criterion is achieved. The stopping criteria can be minimum error goal and/or maximum iteration count. The first stage consisting of straightforward calculations is repeated in order to execute the testing phase of neural network.

3. DISTINGUISHING FEATURE SELECTOR

In spite of numerous approaches in the literature, feature selection is still an ongoing research topic for text classification research. Researchers are still looking for new techniques to select distinctive features so that the classification accuracy can be improved and the processing time can be reduced, as well. For this purpose, a novel filter-based probabilistic feature selection method, namely distinguishing feature selector (DFS), is proposed for text classification. DFS selects distinctive features while eliminating uninformative ones considering certain requirements on term characteristics. Theoretical background of DFS and the corresponding experiments are given in the following subsections.

3.1. Theoretical Background

An ideal filter-based feature selection method should assign high scores to distinctive features while assigning lower scores to irrelevant ones. In case of text classification, each distinct term corresponds to a feature. It is an important point to decide criteria to make a feature relevant or irrelevant. For this purpose, some general requirements are determined in order to construct an effective feature selection method for text classification. Then, ranking of terms should be carried out considering the following requirements:

- A term, which frequently occurs in a single class and does not occur in the other classes, is distinctive; therefore, it must be assigned a high score.
- A term, which rarely occurs in a single class and does not occur in the other classes, is irrelevant; therefore, it must be assigned a low score.
- A term, which frequently occurs in all classes, is irrelevant; therefore, it must be assigned a low score.
- A term, which occurs in some of the classes, is relatively distinctive; therefore, it must be assigned a relatively high score.

Based on the first and the second requirements, an initial scoring framework is constituted as follows:

$$\sum_{i=1}^{M}\frac{P(C_i \mid t)}{P(\bar{t} \mid C_i)+1} \tag{3.1}$$

where M is the number of classes, $P(C_i \mid t)$ is the conditional probability of class C_i given presence of term t and $P(\bar{t} \mid C_i)$ is the conditional probability of absence of term t given class C_i. It is obvious from this formulation that a term occurring in all documents of a class and not occurring in the other classes will be assigned 1.0 as the top score. Moreover, features rarely occurring in a single class while not occurring in the other classes would get lower scores. However, this formulation does not satisfy the third requirement because the features occurring in every document of all classes are invalidly assigned 1.0 as well. In order to resolve this issue, the formulation is extended to

$$DFS(t)=\sum_{i=1}^{M}\frac{P(C_i \mid t)}{P(\bar{t} \mid C_i)+P(t \mid \bar{C}_i)+1} \tag{3.2}$$

where $P(t \mid \bar{C}_i)$ is the conditional probability of term t given the classes other than C_i. Since addition of $P(t \mid \bar{C}_i)$ to the denominator decreases scores of the terms occurring in all classes, the third requirement is also satisfied. Considering the entire formulation, the fourth and the last requirement is satisfied, as well. The formulation provides global discriminatory powers of the features over the entire text collection rather than class specific scores. It is obvious from this scoring scheme that DFS assigns scores to the features between 0.5 and 1.0 according to their significance. In other words, the most discriminative terms have an importance score that is close to 1.0 while the least discriminative terms are assigned an importance score that converges to 0.5. Once the discriminatory powers of all terms in a given collection are attained, top-N terms can be selected just as in the case of the other filter techniques. A sample collection is provided in Table 3.1 to illustrate how DFS works. Also, calculation of feature scores according to DFS is shown in Eq. (3.3).

Table 3.1. Sample collection

Document Name	Content	Class
Doc 1	cat	C1
Doc 2	cat dog	C1
Doc 3	cat dog mouse	C2
Doc 4	cat mouse	C2
Doc 5	cat fish	C3
Doc 6	cat fish mouse	C3

$$
\begin{aligned}
DFS('cat') &= \frac{(2/6)}{(0/2+4/4+1)} + \frac{(2/6)}{(0/2+4/4+1)} + \frac{(2/6)}{(0/2+4/4+1)} = 0.5000 \\
DFS('dog') &= \frac{(1/2)}{(1/2+1/4+1)} + \frac{(1/2)}{(1/2+1/4+1)} + \frac{(0/2)}{(2/2+2/4+1)} = 0.5714 \\
DFS('mouse') &= \frac{(0/3)}{(2/2+3/4+1)} + \frac{(2/3)}{(0/2+1/4+1)} + \frac{(1/3)}{(1/2+2/4+1)} = 0.7000 \\
DFS('fish') &= \frac{(0/2)}{(2/2+2/4+1)} + \frac{(0/2)}{(2/2+2/4+1)} + \frac{(2/2)}{(0/2+0/4+1)} = 1.0000
\end{aligned}
\tag{3.3}
$$

In this sample scenario, maximum score is assigned to ‘fish’ that occurs in all documents of just a single class, namely C3. The successor is determined as ‘mouse’ due to its occurrence in all documents of class C2 and just a single document of C3. The term ‘dog’ is selected as the third informative feature since it appears once in both class C1 and C2 out of three classes. Finally, the least significant term is determined as ‘cat’ due to its occurrence in all documents of all three classes. Here, ‘fish’ and ‘cat’ represent two extreme cases in terms of discrimination. While ‘fish’ is present in all documents of just a single class, ‘cat’ is present in all documents of the collection. Therefore, ‘fish’ is assigned an importance score of 1.0, which is the highest possible DFS score, whereas ‘cat’ is assigned an importance score of 0.5, which is the lowest possible DFS score. In summary, DFS sensibly orders the terms based on their contributions to class discrimination as ‘fish’, ‘mouse’, ‘dog’, and ‘cat’.

The sample collection and the related results are provided to show briefly how DFS method works. Actual performance of DFS on various benchmark datasets with distinct characteristics is thoroughly assessed in the next subsection.

3.2. Experimental Work

In this section, an in-depth investigation is carried out to compare DFS against the other state-of-the-art feature selection methods mentioned in Section 4 in terms of feature similarity, classification accuracy, dimension reduction rate, and processing time. For this purpose, four different datasets with varying characteristics and two different success measures were utilized to observe effectiveness of DFS method under different circumstances. These datasets are news, spam e-mail and spam SMS datasets. The first dataset consists of the top-10 classes of the celebrated Reuters-21578 ModApte split (Asuncion and Newman, 2007). The second dataset contains 10 classes of another popular text collection, namely 20 Newsgroups (Asuncion and Newman, 2007). The third dataset is an SMS message collection introduced in (Almeida et al., 2011). The fourth dataset is a spam e-mail collection, namely Enron1, which is one of the six datasets used in (Metsis et al., 2006). The characteristics of these datasets are shown in Table 3.2-3.5, respectively.

Table 3.2. Reuters dataset

Class No	Class Label	# Training Samples	#Testing Samples
1	earn	2877	1087
2	acq	1650	719
3	money-fx	538	179
4	grain	433	149
5	crude	389	189
6	trade	369	117
7	interest	347	131
8	ship	197	89
9	wheat	212	71
10	corn	181	38

Table 3.3. Newsgroups dataset

Class No	Class Label	# Training Samples	#Testing Samples
1	alt.atheism	500	500
2	comp.graphics	500	500
3	comp.os.ms-windows.misc	500	500
4	comp.sys.ibm.pc.hardware	500	500
5	comp.sys.mac.hardware	500	500
6	comp.windows.x	500	500
7	misc.forsale	500	500
8	rec.autos	500	500
9	rec.motorcycles	500	500
10	rec.sport.baseball	500	500

Table 3.4. SMS dataset

Class No	Class Label	# Training Samples	#Testing Samples
1	spam	238	509
2	legitimate	1436	3391

Table 3.5. Enron1 dataset

Class No	Class Label	# Training Samples	#Testing Samples
1	spam	1000	500
2	legitimate	2448	1224

The success measures used to make comparisons in these experiments are widely-known Micro-F1 and Macro-F1 values. In micro-averaging, F-measure is computed globally without class discrimination. Hence, all classification decisions in the entire dataset are considered. In case that the classes in a collection are biased, large classes would dominate small ones in micro-averaging. Computation of Micro-F1 can be formulated as

$$MicroF1 = \frac{2\times p\times r}{p+r} \tag{3.4}$$

where pair of (*p*, *r*) corresponds to precision and recall values, respectively, over all the classification decisions within the entire dataset not individual classes.

In macro-averaging, F-measure is computed for each class within the dataset and then the average over all classes is obtained. In this way, equal weight is assigned to each class regardless of the class frequency. Computation of Macro-F1 can be formulated as

$$MacroF1 = \frac{\sum_{k=1}^{C} F_k}{C}, \qquad F_k = \frac{2\times p_k \times r_k}{p_k + r_k} \tag{3.5}$$

where pair of (p_k, r_k) corresponds to precision and recall values of class *k*, respectively.

Similarity, accuracy, dimension reduction, and timing analysis are presented in next subsections. It should also be noted that stop-word removal and stemming (Porter, 1980) were carried out as the two preprocessing steps.

3.2.1. Term Similarity Analysis

Profile of the features that are selected by a feature selection method is one of the good indicators to effectiveness of that method. If distinctive features are assigned high scores by a feature selection method, the classification accuracy obtained by those features will most likely be higher. On the contrary, if irrelevant features are assigned high scores by a feature selection method, the accuracy obtained by those features would be degraded. For this purpose, similarities and dissimilarities of the features that are selected by DFS were first compared against the other selection techniques. Initially, top-10 terms selected by each method are presented in Table 3.6 through Table 3.9. The terms that are specific to an individual selection method are indicated in bold. One can note from the tables that DFS selects similar as well as dissimilar terms in each dataset with respect to the other methods. As an example, in Reuters dataset, nine out of ten terms selected by DFS were also selected by the other methods. However, the remaining one term, namely “corn”, is specific to DFS. Considering that “corn” has an occurrence rate of 73% in class-10 and much lower occurrence rate in the other classes, this term can be regarded as a discriminative feature. Therefore, presence of this term in the top-10 list is quite meaningful.

Table 3.6. Top-10 features in Reuters dataset

No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CHI2	cts	net	shr	qtr	rev	loss	**acquir**	**profit**	**note**	**dividend**
IG	cts	net	wheat	bank	shr	qtr	ton	**export**	trade	agricultur
GI	cts	net	shr	wheat	oil	barrel	qtr	**march**	rev	**crude**
DP	**mln**	**dlr**	cts	loss	net	bank	**pct**	**billion**	trade	**share**
DFS	cts	wheat	net	oil	shr	tonn	**corn**	barrel	qtr	agricultur

Table 3.7. Top-10 features in Newsgroups dataset

No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CHI2	basebal	forsal	auto	atheism	motor cycl	rec	comp	sport	hardwar	**sys**
IG	comp	window	rec	**car**	**dod**	**misc**	**sale**	**refer**	**apr**	hardwar
GI	atheism	basebal	motorcycl	forsal	auto	sport	os	mac	ms	**graphic**
DP	window	**path**	**Id**	**newsgroup**	**date**	**messag**	**subject**	**organ**	**line**	**cantaloup**
DFS	atheism	basebal	motorcycl	forsal	auto	sport	os	mac	ms	hardwar

Table 3.8. Top-10 features in SMS dataset

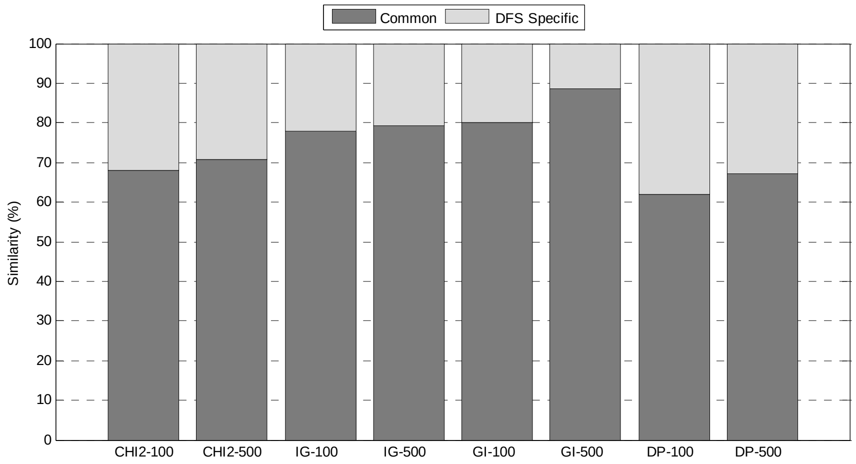
No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CHI2	txt	call	free	mobil	servic	text	award	box	**stop**	contact
IG	call	txt	free	mobil	www	text	claim	servic	award	ur
GI	call	txt	free	www	claim	mobil	prize	text	ur	servic
DP	txt	call	free	mobil	Service	text	award	box	contact	**urgent**
DFS	txt	free	www	claim	mobil	call	prize	**guarante**	**uk**	servic

Table 3.9. Top-10 features in Enron1 dataset

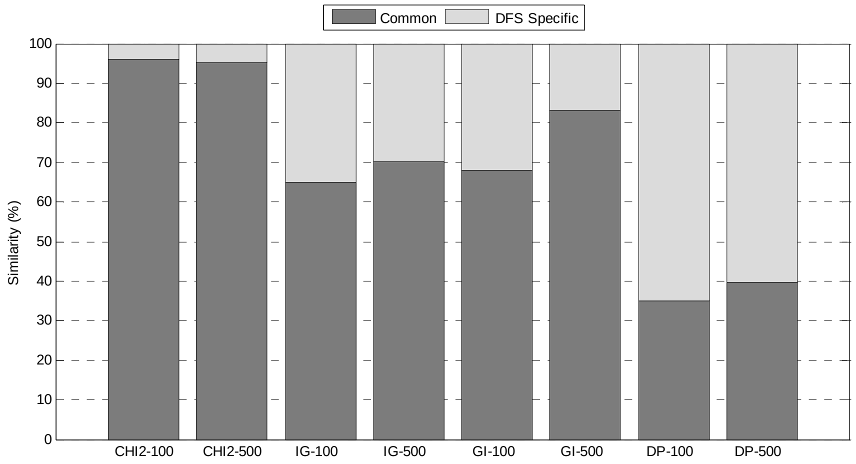
No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CHI2	http	cc	enron	gas	ect	pm	meter	forward	hpl	**www**
IG	cc	gas	ect	pm	meter	http	corp	**volum**	**attach**	forward
GI	subject	enron	cc	hpl	gas	forward	ect	daren	hou	pm
DP	ect	hou	enron	meter	**deal**	subject	gas	pm	cc	corp
DFS	enron	cc	hpl	gas	ect	daren	hou	pm	forward	meter

To observe the generalized behavior of DFS, similarities and dissimilarities of top-100 and top-500 features selected by DFS were analyzed for each dataset. Results of this analysis are presented in Figure 3.1a through Figure 3.1d. For instance, in Reuters dataset, 71% of top-500 features, which are selected by DFS, are common with the ones selected by CHI2 whereas the remaining 29% of the features are specific to DFS method. In general, while DFS selected particular amount of similar features to the ones selected by CHI2 in balanced dataset (Newsgroups), and by GI in imbalanced datasets (Reuters, SMS, and Enron1), it also selected completely distinct features with varying quantities in each dataset.

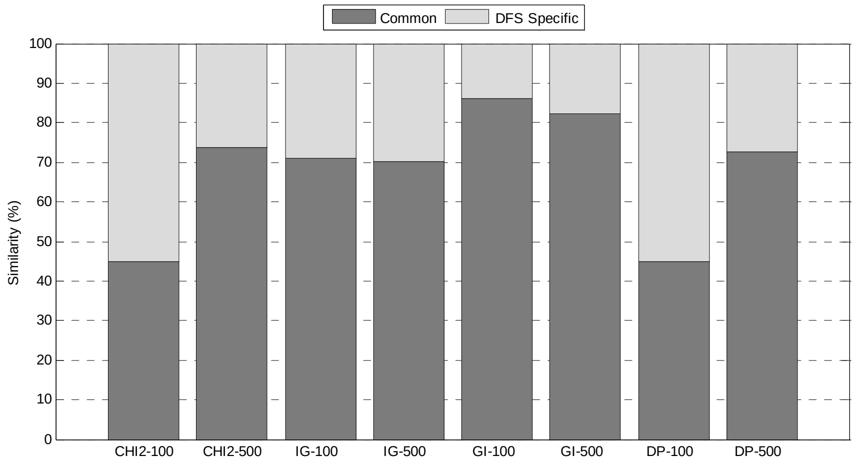
Further analysis on the selected features revealed that GI and DP may include some uninformative terms in their top-*N* lists. For instance, “subject” term occurs in all documents of Newsgroups dataset that makes “subject” uninformative. However, this term was present within the top-50 features selected by GI and DP. Nevertheless, this term and other terms with similar characteristics were not available even in the top-500 features selected by DFS.


Common
DFS Specific
Similarity (%)
0
10
20
30
40
50
60
70
80
90
100
CHI2-100
CHI2-500
IG-100
IG-500
GI-100
GI-500
DP-100
DP-500

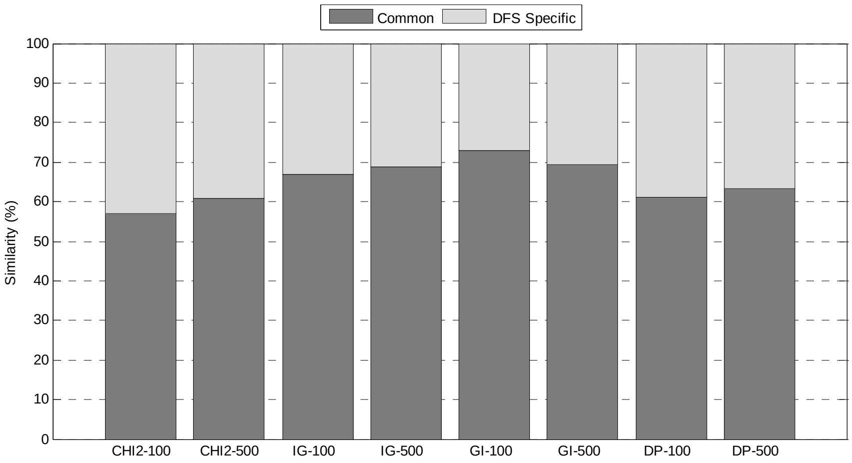
(a)


Common
DFS Specific
Similarity (%)
0
10
20
30
40
50
60
70
80
90
100
CHI2-100
CHI2-500
IG-100
IG-500
GI-100
GI-500
DP-100
DP-500

(b)


Common
DFS Specific
Similarity (%)
0
10
20
30
40
50
60
70
80
90
100
CHI2-100
CHI2-500
IG-100
IG-500
GI-100
GI-500
DP-100
DP-500

(c)



(d)

Figure 3.1. Similarity of the features selected by DFS against the other methods for (a) Reuters (b) Newsgroups (c) SMS (d) Enron1 datasets

3.2.2. Accuracy Analysis

Varying numbers of the features, which are selected by each selection method, were fed into DT, SVM, and NN classifiers. Resulting Micro-F1 and Macro-F1 scores are listed in Tables 3.10-3.21 for each dataset, respectively, where the highest scores are indicated in bold. Considering the highest scores, DFS is either superior to all other methods or runner up with just a slight difference. For instance, in Reuters dataset, the features selected by DFS provided both the highest Micro-F1 and Macro-F1 scores using DT classifier. Similarly, in Newsgroups dataset, the highest Micro-F1 and Macro-F1 scores were obtained with SVM and NN classifiers that use the features selected by DFS. As another example, in SMS dataset, both the highest Micro-F1 and Macro-F1 scores were attained by DT and NN classifiers using the features that are selected by DFS, as well. Finally, in Enron1 dataset, both the highest Micro-F1 and Macro-F1 scores were attained by all of the three classifiers using the features that are selected by DFS.

Table 3.10. Success measures (%) for Reuters dataset using DT classifier

Feature	Micro-F1						Macro-F1					
Size	10	50	100	200	300	500	10	50	100	200	300	500
CHI2	61.97	80.70	82.63	82.63	83.06	82.74	17.36	56.74	57.28	57.09	58.75	58.99
IG	69.61	81.34	83.03	82.89	83.21	82.81	35.32	58.75	58.04	57.64	59.18	59.01
GI	67.74	81.63	81.84	83.21	81.88	82.74	27.61	58.99	57.40	58.74	58.54	58.58
DP	68.35	79.26	81.27	81.70	82.38	82.42	37.13	54.72	59.07	57.39	58.10	58.15
DFS	70.15	82.78	82.63	82.92	83.10	**83.28**	33.91	**61.25**	58.40	58.65	59.22	59.42

Table 3.11. Success measures (%) for Reuters dataset using SVM classifier

Feature	Micro-F1						Macro-F1					
Size	10	50	100	200	300	500	10	50	100	200	300	500
CHI2	62.04	83.17	85.83	85.76	85.97	85.90	15.11	55.97	60.67	64.05	64.96	64.26
IG	69.72	83.57	85.68	**86.33**	86.01	85.86	34.90	59.66	61.48	**66.55**	65.16	64.92
GI	68.82	83.42	85.79	86.04	85.94	**86.33**	28.38	59.95	62.13	64.62	65.41	65.91
DP	67.17	81.99	85.47	85.76	85.83	86.11	32.36	55.43	61.26	64.48	66.04	65.47
DFS	70.11	84.18	85.72	86.04	85.90	85.79	32.12	61.39	62.55	64.68	65.98	64.93

Table 3.12. Success measures (%) for Reuters dataset using NN classifier

Feature	Micro-F1						Macro-F1					
Size	10	50	100	200	300	500	10	50	100	200	300	500
CHI2	61.43	81.66	84.74	85.35	85.57	85.64	15.06	54.65	60.97	63.14	63.09	**64.74**
IG	68.68	81.34	83.04	85.39	85.27	85.80	32.58	57.89	59.56	64.02	62.49	63.16
GI	67.87	81.64	84.28	85.40	85.36	85.57	28.49	59.52	63.42	62.76	62.84	63.83
DP	65.86	79.52	84.03	85.62	85.78	85.68	26.80	52.49	59.93	64.37	63.73	63.51
DFS	69.26	81.07	84.07	85.40	**85.96**	85.92	30.30	60.08	62.09	63.18	64.33	64.31

Table 3.13. Success measures (%) for Newsgroups dataset using DT classifier

Feature	Micro-F1						Macro-F1					
Size	10	50	100	200	300	500	10	50	100	200	300	500
CHI2	71.32	97.84	97.86	97.68	97.70	97.78	69.32	97.85	97.87	97.69	97.71	97.79
IG	78.38	97.80	97.78	97.70	97.72	97.62	78.37	97.81	97.79	97.71	97.73	97.63
GI	87.62	97.86	97.88	**97.90**	97.70	97.72	85.03	97.87	97.89	**97.91**	97.71	97.73
DP	22.56	97.62	97.58	97.74	97.74	97.64	15.83	97.63	97.59	97.75	97.75	97.65
DFS	88.10	97.84	97.76	97.76	97.80	97.78	86.04	97.85	97.77	97.77	97.81	97.79

Table 3.14. Success measures (%) for Newsgroups dataset using SVM classifier

Feature	Micro-F1						Macro-F1					
Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	70.36	97.02	97.20	96.84	96.60	96.22	66.12	97.01	97.19	96.85	96.61	96.23
IG	78.40	97.24	97.14	96.14	95.88	96.32	76.89	97.23	97.14	96.15	95.89	96.32
GI	87.96	97.20	96.84	97.04	96.14	96.28	85.41	97.19	96.83	97.05	96.15	96.28
DP	20.44	96.96	97.12	95.90	95.20	95.50	10.64	96.95	97.12	95.91	95.20	95.50
DFS	88.18	97.06	**97.32**	96.88	96.56	96.18	86.00	97.05	**97.32**	96.88	96.57	96.19

Table 3.15. Success measures (%) for Newsgroups dataset using NN classifier

Feature	Micro-F1						Macro-F1					
Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	67.13	96.12	96.42	96.94	96.65	96.44	60.66	96.11	96.40	96.93	96.64	96.42
IG	68.79	94.60	95.75	95.76	96.40	96.16	66.42	94.55	95.74	95.75	96.39	96.15
GI	86.52	95.72	95.99	96.58	96.76	96.29	82.84	95.67	95.98	96.57	96.76	96.28
DP	20.29	95.32	96.61	96.54	96.28	95.87	10.65	95.29	96.60	96.53	96.27	95.85
DFS	86.62	96.00	96.36	**96.98**	96.68	96.42	83.06	95.96	96.35	**96.97**	96.67	96.41

Table 3.16. Success measures (%) for SMS dataset using DT classifier

Feature	Micro-F1						Macro-F1					
Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	93.97	95.85	96.13	96.03	96.18	96.33	84.50	90.20	91.01	90.55	91.05	91.45
IG	94.67	96.23	96.36	96.18	96.13	96.23	86.89	91.23	91.49	90.91	90.94	91.21
GI	94.69	96.41	96.23	96.21	96.26	96.28	86.78	91.56	91.21	91.19	91.36	91.40
DP	93.80	95.77	96.13	96.05	96.18	96.33	83.89	89.98	91.01	90.72	91.05	91.45
DFS	94.41	**96.49**	96.00	96.23	96.26	96.33	85.81	**91.77**	90.53	91.26	91.33	91.48

Table 3.17. Success measures (%) for SMS dataset using SVM classifier

Feature	Micro-F1						Macro-F1					
Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	93.05	96.54	96.64	97.05	97.05	97.08	84.03	91.84	92.07	93.11	93.14	93.17
IG	94.08	96.74	97.23	97.13	96.87	97.31	85.78	92.23	93.56	93.20	92.57	93.68
GI	94.00	96.82	97.26	97.33	97.15	97.41	85.56	92.51	93.67	93.83	93.37	**94.00**
DP	93.33	96.67	96.82	97.18	97.15	96.95	83.83	92.18	92.54	93.42	93.35	92.84
DFS	94.18	96.95	96.90	97.18	97.05	**97.44**	85.92	92.98	92.85	93.43	93.09	93.94

Table 3.18. Success measures (%) for SMS dataset using NN classifier

	Micro-F1						Macro-F1					
Feature Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	94.19	96.37	96.70	97.33	97.19	97.15	85.56	91.38	92.19	93.77	93.37	93.21
IG	94.50	96.65	97.15	97.08	97.02	97.12	86.57	92.09	93.36	93.10	92.91	93.10
GI	94.50	96.82	97.23	97.32	97.28	97.26	86.62	92.53	93.51	93.69	93.59	93.47
DP	94.11	96.48	96.60	97.31	97.17	97.28	85.30	91.63	91.90	93.70	93.31	93.57
DFS	94.55	97.00	97.15	97.17	96.94	**97.39**	86.47	93.02	93.36	93.40	92.74	**93.82**

Table 3.19. Success measures (%) for Enron1 dataset using DT classifier

	Micro-F1						Macro-F1					
Feature Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	81.15	91.13	91.01	90.31	90.26	91.88	79.89	89.62	89.50	88.49	88.42	90.39
IG	78.48	89.62	90.55	90.08	90.43	89.50	76.90	87.88	88.86	88.16	88.52	87.51
GI	80.97	90.08	90.84	91.30	91.13	91.07	79.70	88.32	88.93	89.53	89.39	89.33
DP	74.88	91.13	90.14	89.15	89.56	89.56	73.91	89.72	88.43	87.05	87.58	87.54
DFS	83.64	90.31	90.02	91.24	**92.00**	91.65	82.27	88.60	88.16	89.54	**90.42**	90.02

Table 3.20. Success measures (%) for Enron1 dataset using SVM classifier

	Micro-F1						Macro-F1					
Feature Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	79.52	89.79	90.95	90.49	90.43	92.75	78.28	88.23	89.60	88.90	88.79	91.38
IG	78.89	88.34	91.18	92.00	91.59	92.58	77.34	86.58	89.80	90.61	90.07	91.25
GI	80.63	89.79	89.50	91.36	91.71	91.76	79.34	88.20	87.67	89.86	90.04	90.09
DP	75.58	90.08	91.18	91.47	92.34	90.89	74.20	88.57	89.80	89.95	90.91	89.21
DFS	83.30	89.97	89.85	92.63	**93.21**	93.10	81.90	88.37	88.17	91.39	**91.96**	91.70

Table 3.21. Success measures (%) for Enron1 dataset using NN classifier

	Micro-F1						Macro-F1					
Feature Size	**10**	**50**	**100**	**200**	**300**	**500**	**10**	**50**	**100**	**200**	**300**	**500**
CHI2	79.91	90.75	91.37	91.37	91.25	91.96	78.66	89.38	90.05	89.90	89.78	90.44
IG	79.18	89.93	91.16	91.97	91.59	92.31	77.53	88.38	89.74	90.56	90.083	90.82
GI	80.92	90.91	91.33	92.16	92.24	93.75	79.58	89.30	89.65	90.72	90.74	92.48
DP	75.87	90.99	91.85	91.89	92.38	92.04	74.42	89.60	90.51	90.42	90.97	90.55
DFS	83.41	91.04	91.42	92.63	93.48	**94.35**	81.99	89.59	89.94	91.34	92.27	**93.19**

3.2.3. Dimension Reduction Analysis

In addition to accuracy, dimension reduction rate is another important aspect of feature selection. Therefore, an analysis for dimension reduction was also carried out during the experiments. To compare the efficiency of DFS in terms of dimension reduction rate together with accuracy, a scoring scheme (Gunal and Edizkan, 2008) is used. This scoring scheme that combines these two information was described below:

This scheme favors better accuracy at lower dimensions as given in

$$Score = \frac{1}{k}\sum_{i=1}^{k}\frac{\dim_N}{\dim_i}R_i \tag{3.6}$$

where N is the maximum feature size utilized, k is the number of trials, $\dim_i$ is the feature size at the ith trial, and R_i is the success rate of the ith trial.

The result of dimension reduction analysis using the described scoring scheme is presented in Table 3.22 through Table 3.25 where the highest scores are indicated in bold. The success rate of trials, used for calculation of performance scores, are Micro-F1 or Macro-F1 scores as indicated in these tables. It is apparent from these tables that DFS provides comparable and even better performance with respect to the other methods most of the time.

Table 3.22. DR scores based on Micro-F1/Macro-F1 in Reuters dataset

	SVM		**DT**		**NN**	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**
CHI2	801	325	791	337	792	322
IG	866	**498**	856	491	851	**472**
GI	858	444	840	427	845	444
DP	842	469	840	**500**	825	416
DFS	**870**	478	**863**	484	**856**	459

Table 3.23. DR scores based on Micro-F1/Macro-F1 in Newsgroups dataset

	SVM		**DT**		**NN**	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**
CHI2	912	877	923	906	883	829
IG	979	967	982	982	893	874
GI	1059	1038	1059	1037	1044	1013
DP	495	414	516	460	491	411
DFS	**1061**	**1043**	**1063**	**1046**	**1045**	**1016**

Table 3.24. DR scores based on Micro-F1/Macro-F1 in SMS dataset

	SVM		DT		NN	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**
CHI2	1100	1010	1106	1009	1110	1023
IG	1110	1027	1112	**1031**	1113	1033
GI	1110	1026	**1113**	1030	1114	**1035**
DP	1103	1010	1104	1003	1109	1021
DFS	**1111**	**1029**	1110	1022	**1114**	1034

Table 3.25. DR scores based on Micro-F1/Macro-F1 in Enron1 dataset

	SVM		DT		NN	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**
CHI2	966	951	982	966	972	957
IG	960	942	956	937	965	946
GI	975	959	979	963	982	965
DP	935	918	928	914	940	923
DFS	**999**	**982**	**1001**	**984**	**1003**	**987**

3.2.4. Timing Analysis

In order to measure algorithm complexities, it is necessary to analyze especially loops in the execution of methods. As the pseudo-code of all methods are same as below:

```
for classcount=1:M      //M is the number of classes
            //some arithmetic operations specific to feature selection method
end
```

All of the five feature selection methods have time complexity of *O(N)* in this case. Algorithmic complexities of all feature selection methods considered in this part of the dissertation were computed to be the same. Therefore, the processing time of DFS, rather than its algorithmic complexity, was investigated and compared to the other methods. For this purpose, the computation time of importance score for a single term was considered. The measurements were taken on a computer equipped with Intel Core i7 1.6 GHz processor and 6 GB of RAM. The results of the timing analysis, which are given in Table 3.26, indicate that DFS is the fastest method among all.

Table 3.26. Timing analysis

	CHI2	IG	GI	DP	DFS
Computation time (sec)	0.0632	0.0693	0.0371	0.0797	0.0343

3.3. Conclusions

A novel filter based feature selection method namely DFS was introduced for text classification research. DFS assesses the contributions of terms to the class discrimination in a probabilistic approach and assigns certain importance scores to them. Using different datasets, classification algorithms and success measures, effectiveness of DFS was investigated and compared against well-known filter techniques. The results of a thorough experimental analysis clearly indicate that DFS offers a considerably successful performance in terms of accuracy, dimension reduction rate, and processing time.

4. GENETIC ALGORITHM ORIENTED LATENT SEMANTIC FEATURES

In addition to feature extraction and feature selection, feature transformation approaches are also used to reduce feature dimension. While either feature selection or feature transformation methods can be individually used for dimension reduction, combinations of these methods are also possible. Moreover, these combinations may provide even better performance. As an example, a two-stage feature selection strategy consisting of various feature selection methods and LSI is proposed for text classification (Meng et al., 2011). In this work, feature selection methods are initially applied to obtain a discriminative subset of the original feature set. Then, LSI is used to transform the subset into a further discriminative lower-dimensional set. Experimental results on two spam e-mail datasets demonstrate that this two-stage method performs better against the individual methods. In another example, information gain-based feature selection method and PCA is sequentially applied on multi-class text collections (Uguz, 2011). Yet again, the combination of feature selection and transformation further improves the classification performance.

Considering the feature transformation, there are also several efforts projecting the data in a different way than that of LSI or PCA. For instance, selection of the best subset of principal components among all rather than using those with the highest eigenvalues are found as an efficient method to determine the optimal multivariate regression model in (Barros and Rutledge, 1998). As another example, a new framework that selects principal components efficiently is constructed in (Zheng et al., 2005) for face recognition task, and it is concluded that some smaller principal components are useful whereas some larger ones can be removed as well. Another transformation method, namely common vector approach (CVA), also states that the directions corresponding to the smallest eigenvalues rather than the largest ones may provide more discrimination (Gulmezoglu et al., 2001; Gunal and Edizkan, 2008).

Inspiring from the abovementioned approaches; in this dissertation, genetic algorithm oriented latent semantic features (GALSF) are proposed for text classification task. The proposed method consists of two stages, namely feature selection and feature transformation. The feature selection stage is carried out using the state-of-the-art filter-based methods. The feature transformation stage employs LSI empowered by genetic algorithm (GA) such that a better projection is attained using appropriate singular vectors, which are not limited to the ones corresponding to the largest singular values, unlike standard LSI approach. In this way, the singular vectors with small singular values may also be used for projection whereas the vectors with large singular values may be eliminated as well to obtain better discrimination. Effectiveness of the proposed method is comparatively evaluated against feature selection methods and the combination of feature selection and transformation methods on two-class and multi-class text collections, namely Enron1 and Reuters-21578. For both collections, GALSF surpasses the other methods in terms of classification performance.

In the following subsections, GALSF is explained in details. Since this method includes GA, some basic concepts are also described in the next subsection.

4.1 Genetic Algorithms

GA is a suboptimal search method stimulated from biological evolution process (Goldberg, 1989; Gunal, 2012). The underlying idea of GA is the survival of the fittest solutions among a population of potential solutions for a given problem. Hence, new generations formed by the surviving solutions are expected to provide better approximations to the optimum solution. The solutions correspond to chromosomes that are encoded with an appropriate alphabet. The fitness value for each chromosome is computed by a fitness function. New generations are obtained by applying the genetic operators, namely crossover and mutation, onto the fittest members of the population. While crossover uses more than one parent solutions and produces a child solution from them, mutation alters one or more gene values within a chromosome. Initial population can be arbitrarily or manually defined. Population size, number of generations, probability of crossover, and mutation are specified empirically.

When GA is used for attribute or feature selection task, chromosome length is set to the dimension of the original set of features. The chromosomes are then encoded with binary (0, 1) alphabet. Hence, in a chromosome, the indices represented with “1” indicate the selected features, whereas “0” indicates the unselected ones. As an example, the chromosome {1 0 1 0 0 0 1 0} specifies that the 1st, 3rd, and 7th features are selected while the others are eliminated.

4.2. Genetic Algorithm Oriented Latent Semantic Features

As explained previously, *s* singular vectors corresponding to the largest singular values are used to constitute projection matrix in standard LSI. The proposed framework, on the other hand, may employ the singular vectors corresponding to not only large but also small singular values. Appropriate singular vectors for the projection providing better representation are determined using GA. Hence, *k* singular vectors, which are not limited to the ones corresponding to the largest singular values, can be acquired. Therefore, approximation of term-document matrix *M* can now be expressed as follows:

$$M_k = U_k \Sigma_k V_k^T . \tag{4.1}$$

According to this approximation, each document can be represented as follows:

$$doc_{projected} = doc_{original}{}^{T} . U_k \tag{4.2}$$

The fitness value in GA is defined as the well-known micro-F1 measure (Manning et al., 2008; Gunal, 2012) where F-measure is computed globally without class discrimination. Hence, all classification decisions in the entire dataset are considered. Micro-F1 values are attained from the classification of the projected features that are obtained using the selected singular vectors. Consequently, a new subset of singular vectors providing better discrimination in the projected subspace is obtained with the help of GA.

All the steps in GALSF approach can be listed as follows:

Step 1. Perform filter-based feature selection to obtain relevant features among all and to reduce dimension.

Step 2. Utilize GA oriented LSI method on the selected feature subset in previous step to obtain a new projection providing better discrimination and further dimension reduction.

Step 3. Project the selected features in Step 1 into the new semantic subspace computed in Step 2 to obtain GALSF.

Step 4. Feed GALSF to a pattern classifier for the recognition of the given document.

4.3. Experimental Work

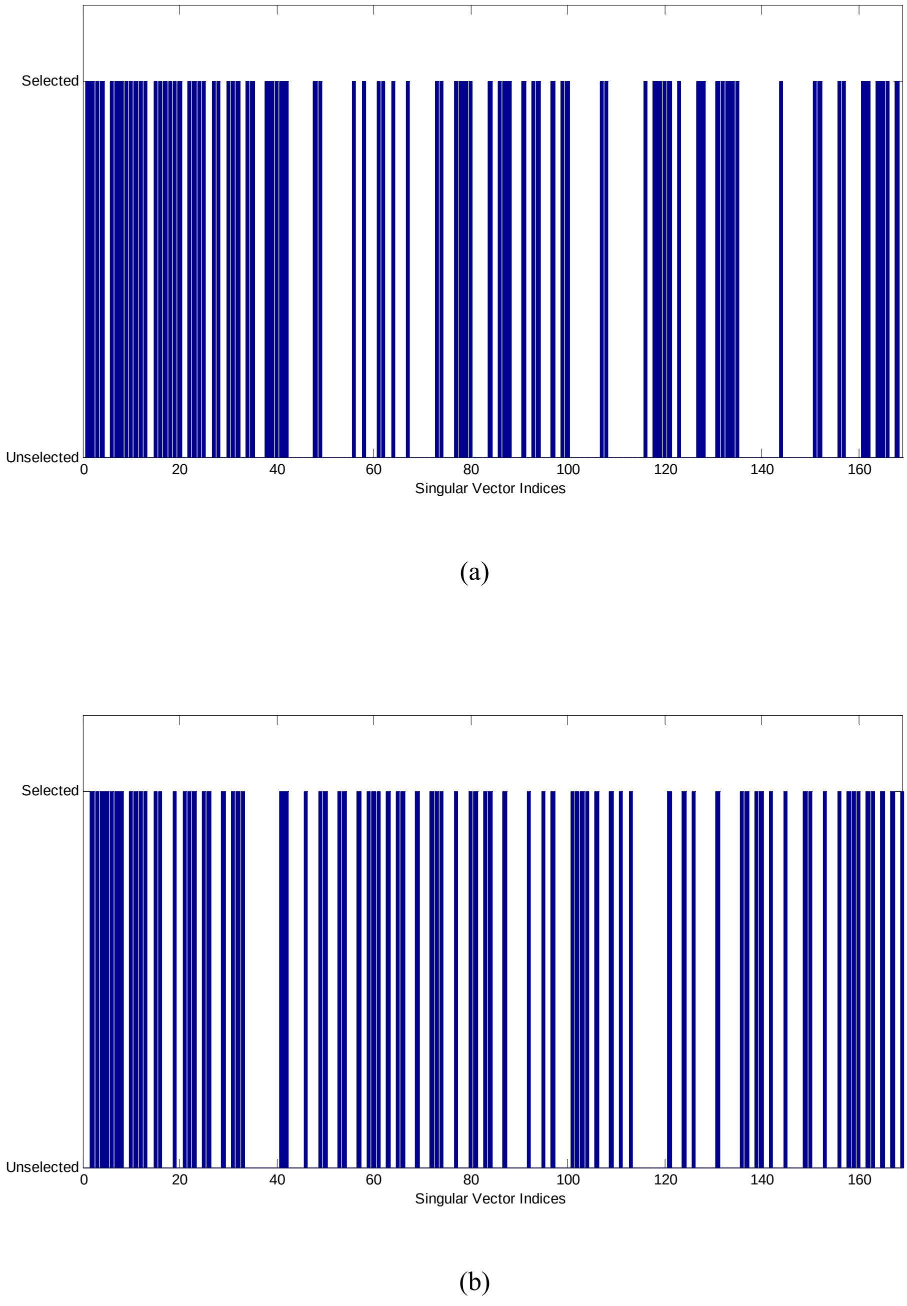
In the experiments, two distinct datasets with varying characteristics were used for the assessment. The first dataset consists of top-10 classes of the celebrated Reuters-21578 ModApte split (Asuncion and Newman, 2007). The second dataset is a spam e-mail collection, namely Enron1, which is one of the six datasets used in (Metsis et al., 2006). While Reuters is a multi-class collection, Enron1 consists of just two classes. Characteristics of these two datasets were presented in Table 3.2 and Table 3.5, respectively.

During feature extraction from text documents, two preprocessing tasks, namely stop-word removal and stemming, were carried out. Also, TF-IDF (Manning et al., 2008) weighting scheme was employed.

For classification task, SVM, which is one of the state-of-the-art pattern classification algorithms, was employed. GA parameters of GALSF method were defined as follows: population size is 100, number of generations is 20, probability of crossover is 0.8, and probability of mutation is 0.08. As indicated before, the fitness value is defined as the Micro-F1 score obtained from classification of the test samples in the datasets.

4.3.1. Profile of Singular Vectors

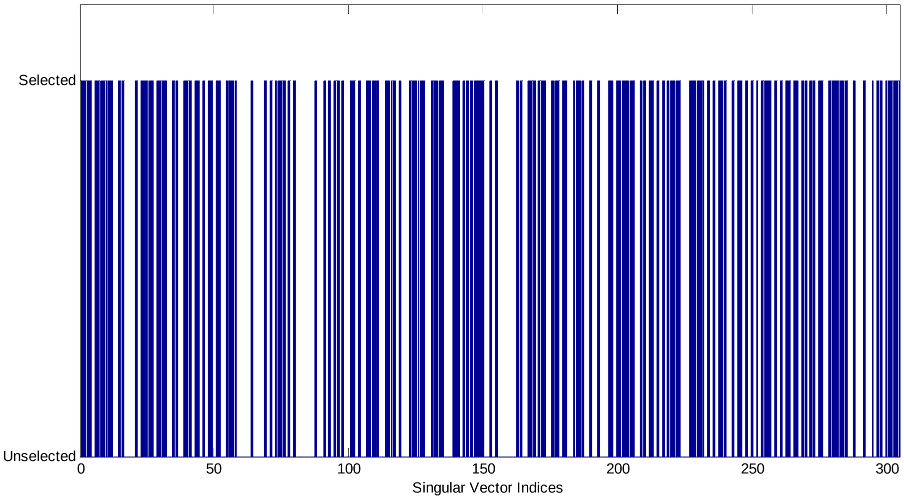
GALSF was obtained by applying filter-based feature selection first and then GA oriented LSI. In the standard LSI procedure, the singular vectors constituting the feature transformation matrix always correspond to the largest singular values. On the contrary, GA oriented LSI has no such limitation depending on the idea that the singular vectors corresponding to not only large but also small singular values may form a transformation matrix that provides more discrimination. Figures 10.1-10.2 display the indices of the selected and unselected singular vectors after applying GALSF approach on the utilized datasets, respectively. In these figures, indices of the singular vectors are listed in descending order of the corresponding singular values. In the standard LSI procedure, the singular vectors corresponding to the largest singular values are employed. In other words, the indices of the selected singular vectors would be between 0 and a predefined number *s*. However, in the proposed framework, it is apparent that the set of selected singular vectors contain both small and large ones with varying numbers.



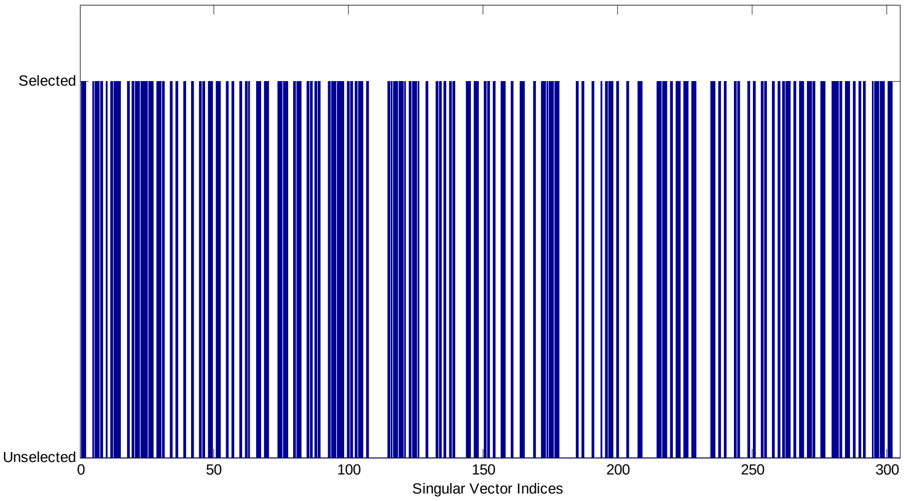
(a)

(b)

Figure 4.1. Reuters dataset: Singular vector selection based on (a) DFS+GALSF (b) CHI2+GALSF



(a)



(b)

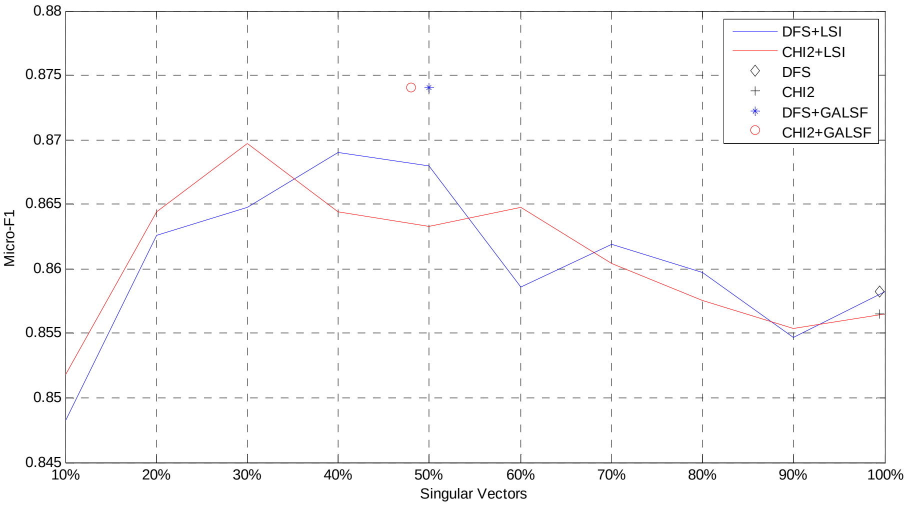
Figure 4.2. Enron1 dataset: Singular vector selection based on (a) DFS+GALSF (b) CHI2+GALSF

4.3.2. Accuracy Analysis

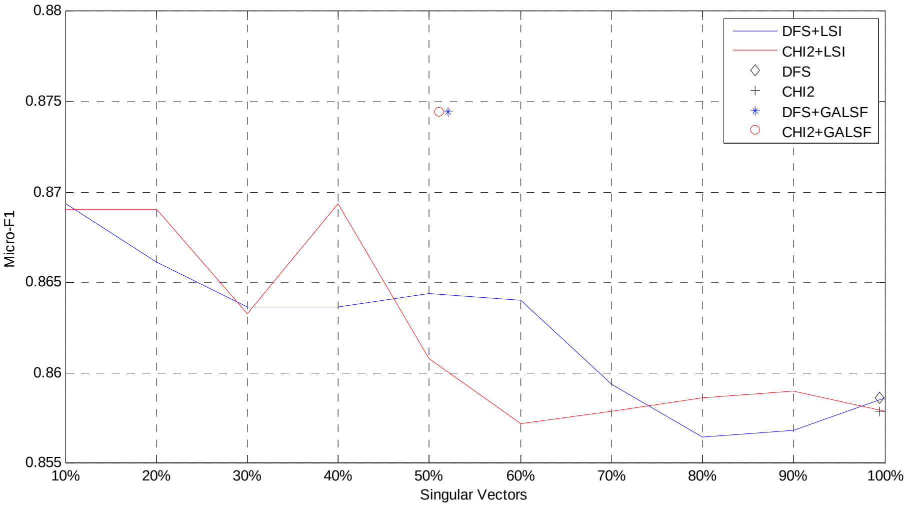
In this part, contribution of GALSF to the classification performance is comparatively evaluated against the existing approaches. The evaluation is carried out using various numbers of the selected features to observe efficiency of the proposed method in each case. Specifically, 1%, 2.5%, 5%, and 10% of the whole feature set, which are initially selected by the filter methods (DFS, CHI2), are fed into GALSF to obtain the proposed features (DFS+GALSF, CHI2+GALSF). Those features are finally fed into SVM for classification. GALSF are compared (in terms of the attained micro-F1 scores) to the features directly selected by feature selection methods (DFS, CHI2), and to the features obtained by the combinations of feature selection and transformation (DFS+LSI, CHI2+LSI) for each dataset. In case of (DFS+LSI, CHI2+LSI), various numbers of singular vectors, ranging from 10% to 100% of all vectors, were used to form the transformation matrix. Obviously, if 100% of the singular vectors are used, no further dimension reduction is applied. Hence, the resulting transformation will yield the same feature subset selected by the regarding feature selection method in previous step. The results of the conducted experiments on Reuters dataset are presented in Figures 4.3a through 4.3d, whereas the results belonging to Enron1 dataset are available in Figures 4.4a through 4.4d.

It is obvious from these figures that the proposed framework (either DFS+GALSF or CHI2+GALSF) outperformed the others in all cases. The amount of singular vectors selected by GA algorithm was always around 50% of all vectors. The runner-up of this analysis was the combination of feature selection and transformation obtained by standard LSI (DFS+LSI or CHI2+LSI) with just a single exception, where DFS beated (DFS+LSI) and (CHI2+LSI) on Enron1 dataset if 1% of the features are selected. Thus, individual feature selection methods (DFS and CHI2) took the last place in this analysis.

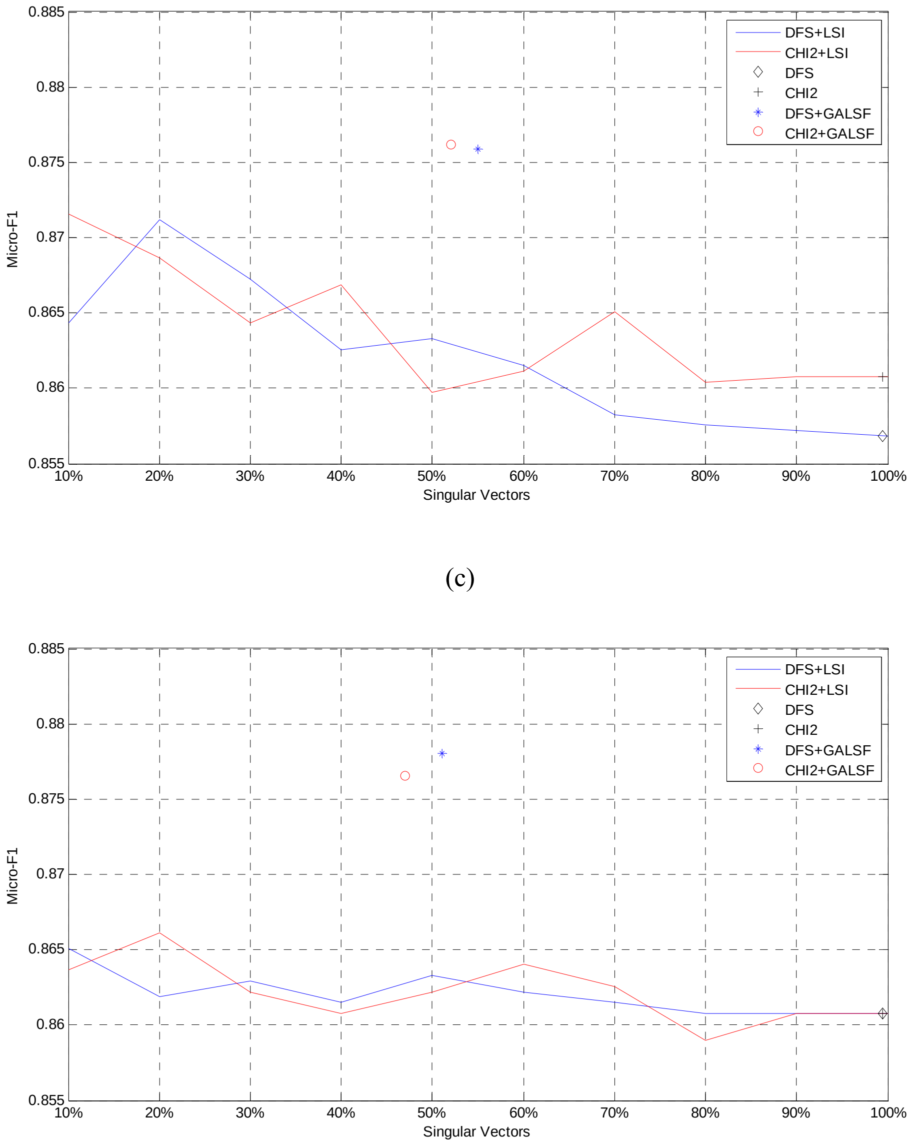
Based on this analysis, it can be stated that the proposed method not only provides improved accuracy over both feature selection and combination of selection and transformation but also offers further dimension reduction with respect to individual feature selection methods. Since both (DFS+GALSF) and (CHI2+GALSF) are superior to the other approaches, one can also state that the efficiency of the proposed framework is independent from the utilized feature selection method. These statements are valid for both datasets.


0.88
0.875
0.87
0.865
0.86
0.855
0.85
0.845
Micro-F1
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Singular Vectors
DFS+LSI
CHI2+LSI
◇ DFS
+ CHI2
* DFS+GALSF
○ CHI2+GALSF

(a)


0.88
0.875
0.87
0.865
0.86
0.855
Micro-F1
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Singular Vectors
DFS+LSI
CHI2+LSI
◇ DFS
+ CHI2
* DFS+GALSF
○ CHI2+GALSF

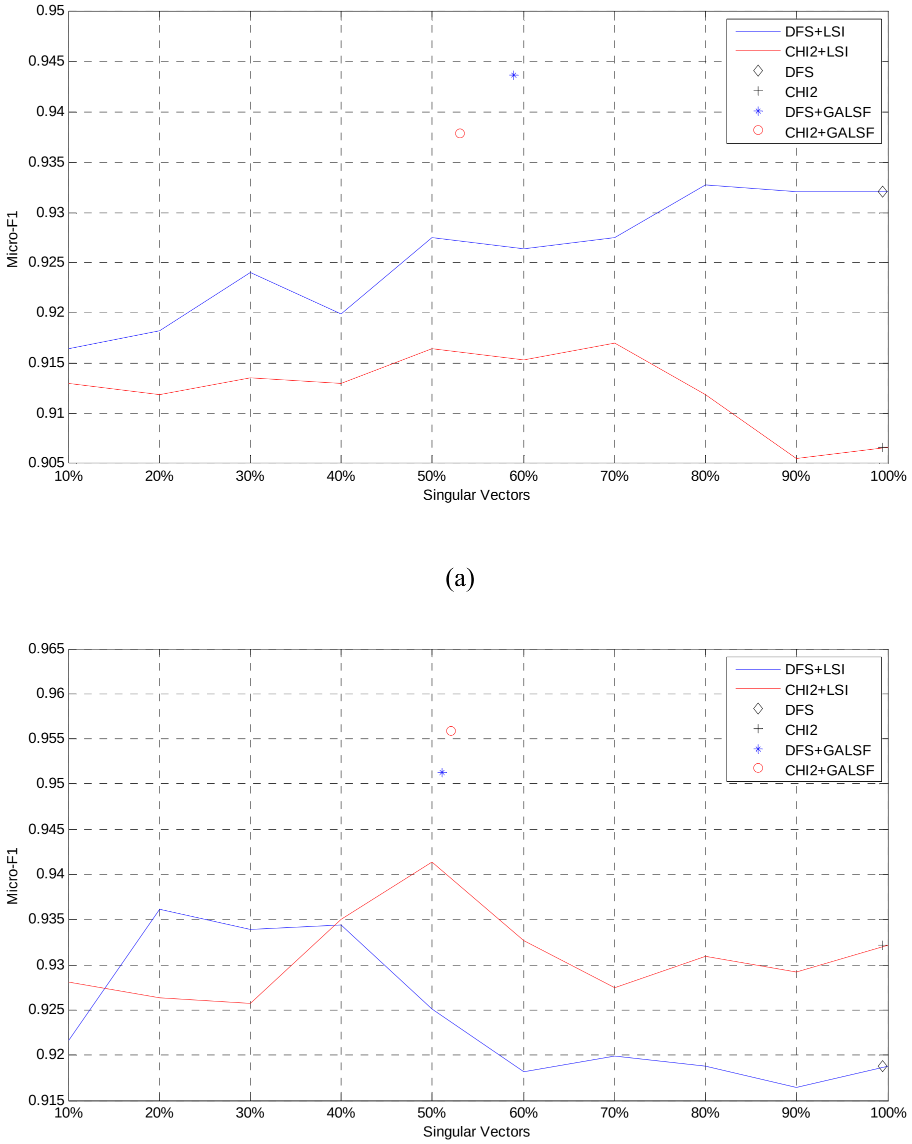
(b)



(c)

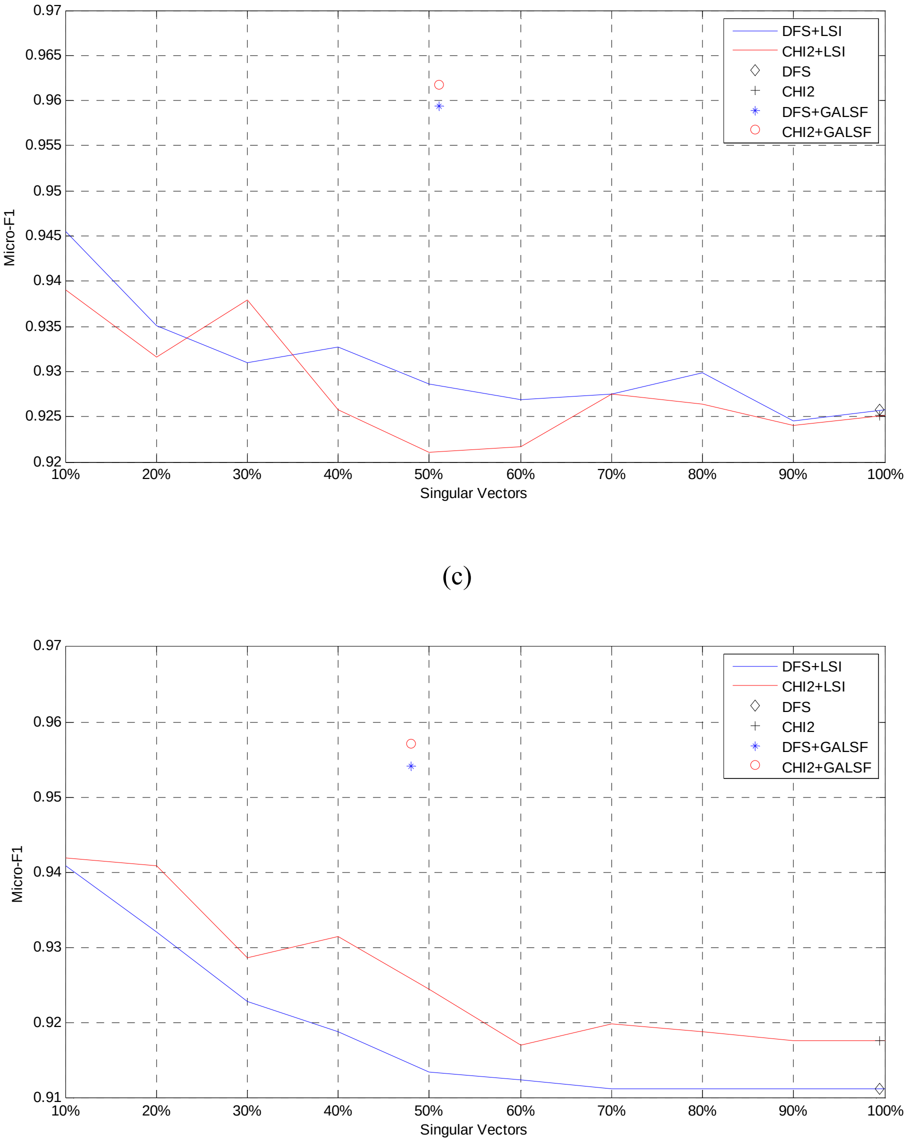
(d)

Figure 4.3. Experiments on Reuters dataset with (a) 1% (b) 2.5% (c) 5% (d) 10% of features


0.95
0.945
0.94
0.935
0.93
0.925
0.92
0.915
0.91
0.905
Micro-F1
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Singular Vectors
DFS+LSI
CHI2+LSI
DFS
CHI2
DFS+GALSF
CHI2+GALSF
0.965
0.96
0.955
0.95
0.945
0.94
0.935
0.93
0.925
0.92
0.915
Micro-F1
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Singular Vectors
DFS+LSI
CHI2+LSI
DFS
CHI2
DFS+GALSF
CHI2+GALSF

(a)

(b)



(c)

(d)

Figure 4.4. Experiments on Enron1 dataset with (a) 1% (b) 2.5% (c) 5% (d) 10% of features

4.4. Conclusions

The contribution of this part of the dissertation was the proposal of genetic algorithm oriented latent semantic features in order to represent text data in a better way. The proposed method consists of two stages, namely feature selection and feature transformation. Effectiveness of the proposed method was evaluated against feature selection and the combination of feature selection and transformation on two-class and multi-class text collections. For both collections, GALSF surpasses the other methods in terms of classification performance. However, the improvement rate of GALSF differs for different datasets and feature dimensions. GALSF generally reduces the respective dimension approximately to the half besides the performance improvement on accuracy. Since it is a secondary dimension reduction step, it is possible to say that the dimension reduction rate is also reasonable. Based on the acquired results, one can conclude that the singular vectors corresponding to small singular values may be useful to obtain a projection providing better discrimination whereas the vectors with large singular values may be useless for this purpose unlike the idea of standard LSI.

5. IMPACT OF FEATURE EXTRACTION AND SELECTION: A CASE STUDY FOR SMS SPAM FILTERING

In recent years, Short Message Service (SMS) has become one of the most common communication methods due to rapid increase in the number of mobile phone users worldwide. This increase has unavoidably attracted spammers and caused SMS spam (unsolicited) message problem just as in the case of spam e-mails. Today, majority of SMS messages received by mobile phones are unfortunately disturbing spam messages such as credit opportunities of banks, promotion and discount announcements of stores, and new tariffs of communications service providers.

Simple techniques including white and black list methods fail to categorize SMS messages without user intervention. Even worse, a phone number inserted into the black list may send legitimate messages beside spam, e.g., a bank may send a spam message including new credit opportunities and a legitimate message containing online banking password as well. In this case, smarter methods such as content based classification are needed.

Though the problem of SMS spam is not as old as of email spam (Puniškis et al., 2006; Puniškis and Laurutis, 2007), there have been several efforts in the literature to detect SMS spam messages. Some examples to those efforts are as follows. Bayesian filtering techniques were employed in (Hidalgo et al., 2006). Feature-based and compression-model-based filters were evaluated in (Cormack et al., 2007). Another filter system using support vector machine and a thesaurus was proposed in (Joe and Shim, 2010). A framework utilizing the content based filtering and challenge-response was introduced in (Yoon et al., 2010). Another SMS anti-spam system combining behavior-based social network and temporal analysis was presented in (Wang et al., 2010). Performances of a number of classifiers in SMS spam filtering were compared in (Almeida et al., 2011). Bayesian learning and support vector machine classification were used in (Yadav et al., 2011). Local-concentration-based (Yuanchun and Ying, 2011) and stylistically motivated features (Sohn et al., 2012) were employed for the filtering process. Bayesian based classifiers were utilized together with the distinctive features determined by information theoretic feature selection methods in (Uysal et al., 2012b). Also, a mobile application using pattern recognition methods is developed particularly for the mobile phones with Android™ operating system (Uysal et al., 2012a). Finally, a number of recent studies on SMS spam filtering are reviewed in (Delany et al., 2012).

In regard to the abovementioned studies, this study as a part of the dissertation extensively analyses the effects of several feature extraction and feature selection methods together on filtering SMS spam messages in two different languages, namely Turkish and English. The entire feature set of the filtering scheme is composed of the features originated from the BoW model (Joachims, 1997), and also an ensemble of SFs adopted for the spam problem. The distinctive features based on the BoW model are determined using chi-square and Gini index based feature selection methods. The selected features are then combined with the structural features, and fed into two distinct pattern classification algorithms, namely *k*NN and SVM, to classify SMS messages as either spam or legitimate. The filtering framework is evaluated on two separate SMS message datasets consisting of Turkish and English messages. For this purpose, the first publicly available Turkish SMS message collection (Uysal et al., 2013) was constituted whereas an existing dataset in English is employed as well. Extensive experimental analysis on both datasets revealed that the combinations of BoW and SFs, rather than BoW features alone, provide better classification performance. Nevertheless, efficacy of the feature selection methods slightly differs in each language. Current feature extraction approaches for SMS spam filtering are explained in the next subsection.

5.1. Feature Extraction Approaches for SMS Spam Filtering

Detection of SMS spam messages is actually a subset of spam e-mail detection problem. While an e-mail may contain text, graphics, hyperlinks, and even attached files (Gunal et al., 2006), an SMS message contains text only limited with 160 characters (ETSI, 1992). Consequently, detection of spam messages corresponds to a two-class text classification problem, where the classes are defined as “spam” or “legitimate”.

Even if SMS spam filtering can be treated as conventional text classification task, the structure of spam messages can be significantly different than that of formal texts. Since the size of an SMS message is limited with just 160 characters, both the message length and number of terms have great importance. Also, the usage of upper or lower case characters can be indicator of spam. Similarly, some non-alphanumeric characters (e.g., “!”, “$”) and numeric characters (e.g., phone numbers) are commonly encountered in spam messages. Finally, URL links are usually observed in SMS spam as well. Considering all those characteristics, in this part of the dissertation, an ensemble of structural features is adopted along with the features originated from the BoW model. The SFs extracted from a given SMS message are summarized in Table 5.1.

Table 5.1. List of structural features

No	Name	Description
SF1	Message length	Number of all characters
SF2	Number of terms	Number of terms obtained using alphanumeric tokenization
SF3	Uppercase character ratio	Number of uppercase characters normalized by the message length
SF4	Non-alphanumeric character ratio	Number of non-alphanumeric characters normalized by the message length
SF5	Numeric character ratio	Number of numeric characters normalized by the message length
SF6	Presence of URL	Presence of “http” and/or “www” terms

It should also be noted that only stemming and lower case conversion are carried out as the preprocessing steps during the feature extraction. Since two different languages, namely Turkish and English, are in consideration within the scope of this work, the stemming stage is specific to the language. In case of Turkish messages, fixed-prefix stemming algorithm with prefix length 5 (Can et al., 2008) is employed, whereas well-known Porter stemming algorithm (Porter, 1980) is utilized for the messages in English. Stopword removal is not applied due to relatively short length of the messages.

5.2. Experimental Work

The impacts of various feature extraction, feature selection, and pattern classification methods on filtering SMS spam messages in Turkish and English were analyzed in the experimental work. These experiments are realized using SMS spam datasets shown in Table 5.2 and Table 5.3. The first dataset is a

Turkish spam SMS dataset which is constructed as a part of research project (Uysal et al., 2013) and the second dataset is an English spam SMS collection including messages originated from British English (Nuruzzaman et al., 2011).

Table 5.2. Turkish SMS dataset

Class No	Class Label	Total Samples
1	spam	420
2	legitimate	430

Table 5.3. English SMS dataset

Class No	Class Label	Total Samples
1	spam	425
2	legitimate	450

For this purpose, eight different feature sets were considered. Those sets are listed in Table 5.4. The first feature set contains only BoW features. The sets between two and seven contain BoW features and a single SF. The last feature set is composed of BoW features and all six SFs together. From now on, the last feature set (BoW + SF1 + SF2 + SF3 + SF4 + SF5+ SF6) will be represented by (BoW + SF1:SF6) for convenience.

Table 5.4. List of feature sets

No	Feature Set
1	BoW
2	BoW + SF1
3	BoW + SF2
4	BoW + SF3
5	BoW + SF4
6	BoW + SF5
7	BoW + SF6
8	BoW + SF1 + SF2 + SF3 + SF4 + SF5+ SF6

During the experiments, selection of BoW features were carried out using CHI2, GI, and DFS methods, where the number of selected features ranged from 1% to 100% of the entire BoW features. As an example, top-10 terms determined by CHI2, GI and DFS methods are listed in Table 5.5 and Table 5.6 for each dataset. It should be noted that several stopwords specific to Turkish (e.g., “ve”, “ile”, “icin”) and English languages (e.g., “i”, “to”, “that”, “your”) are surprisingly present in these lists. Total numbers of preprocessed distinct terms in Turkish and English datasets are 2.690 and 3.179, respectively.

Table 5.5. Top-10 discriminative terms in Turkish dataset

Selection	Terms
CHI2	com, ve, gonde, icin, tl, tr, sadec, hemen, kazan, ile
GI	com, ve, icin, indir, tl, firsa, gonde, tr, ozel, sadec
DFS	com, firsa, indir, gonde, ve, ozel, sadec, tl, tr, icin

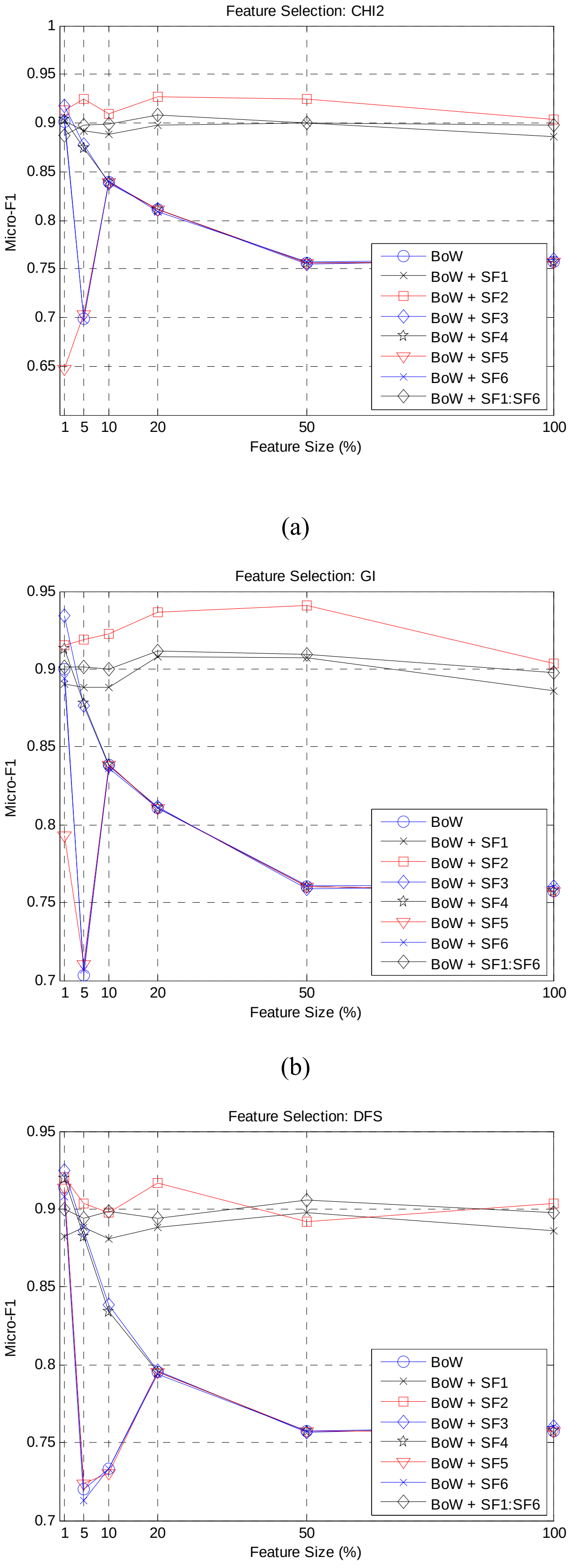
Table 5.6. Top-10 discriminative terms in English dataset

Selection	Terms
CHI2	call, your, i, txt, stop, free, 1, to, that, now
GI	to, call, i, your, now, you, a, txt, stop, for
DFS	call, i, your, txt, stop, free, that, 1, repli, now

The feature sets were then fed into kNN and SVM classifiers. Since both datasets are balanced (i.e., the number of SMS messages in legitimate and spam classes are almost equal), well known Micro-F1 score (Manning et al., 2008) was employed to assess the classification performance. The classification results are presented in Figure 5.1 and Figure 5.2 for Turkish dataset and in Figure 5.3 and 5.4 for English dataset, respectively. The results were obtained using 3-fold cross validation to evaluate the datasets objectively.

In general, rather than BoW features alone, combinations of BoW (regardless of the utilized feature selection method) and SFs provided higher scores in most cases. Particularly, the contributions of SF1, SF2, and SF1:SF6 to classification performance were more obvious than that of the other SFs.

In case of Turkish messages, the highest Micro-F1 score was approximately 0.98. This score was obtained using SF2, and 50% of BoW features selected by DFS, which were together applied on SVM classifier. The score, following the highest Micro-F1, was obtained using SF2 (or, SF1:SF6), and 50% of BoW features selected by CHI2 with the same classifier. On the other hand, the maximum score achieved by kNN classifier was around 0.95 with the combination of SF2 and 50% of BoW features selected by GI.

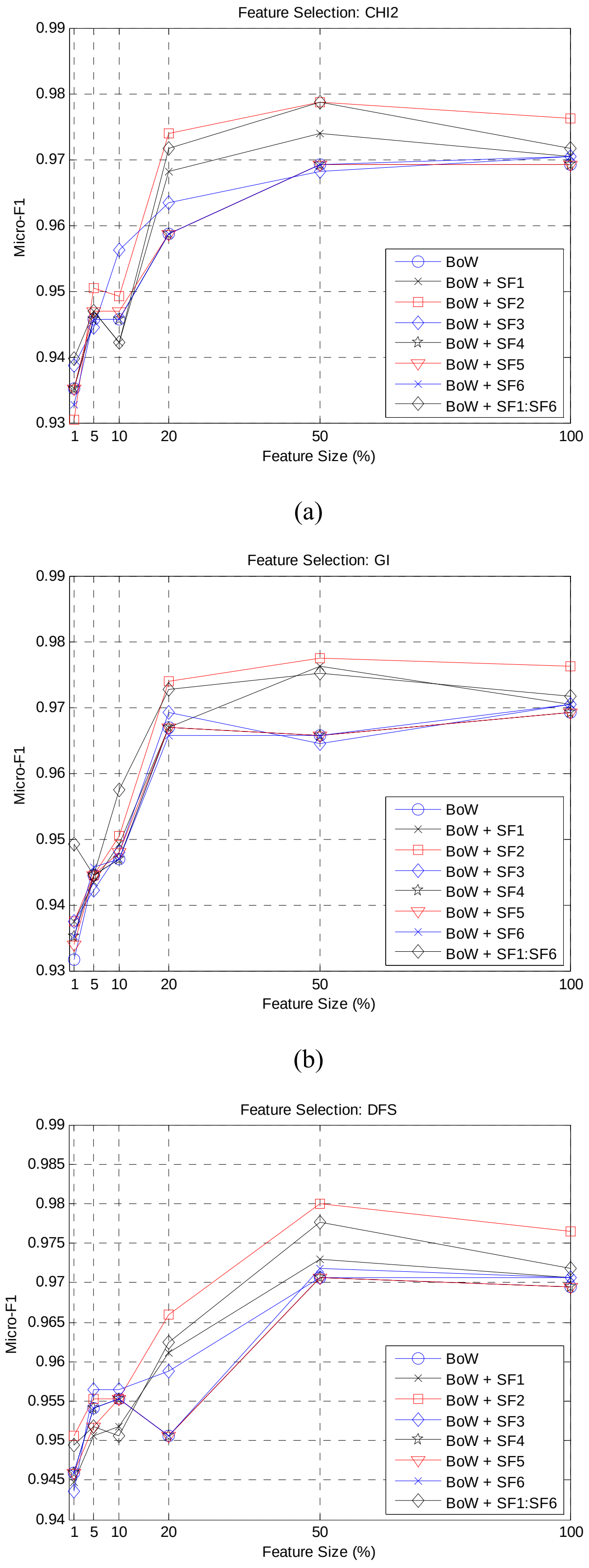


(a)

(b)

(c)

Figure 5.1. kNN classification results for Turkish dataset with (a) CHI2 (b) GI (c) DFS based features

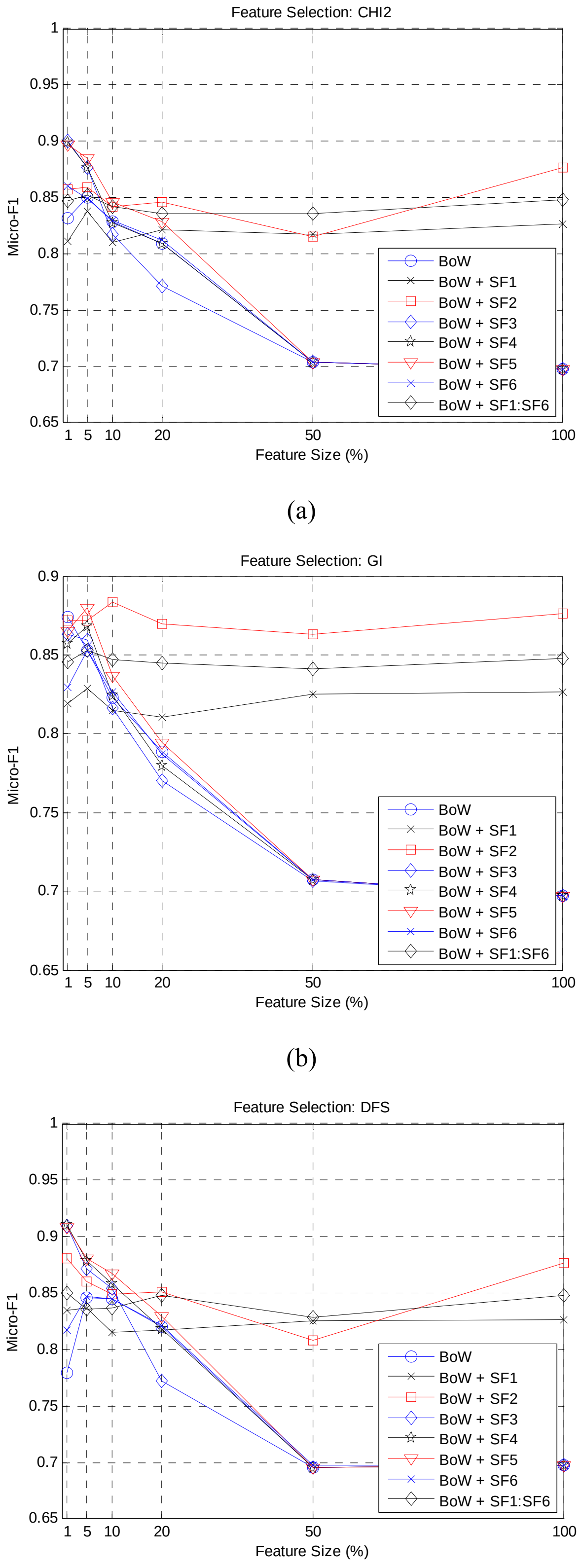


(a)

(b)

(c)

Figure 5.2. SVM classification results for Turkish dataset with (a) CHI2 (b) GI (c) DFS based features

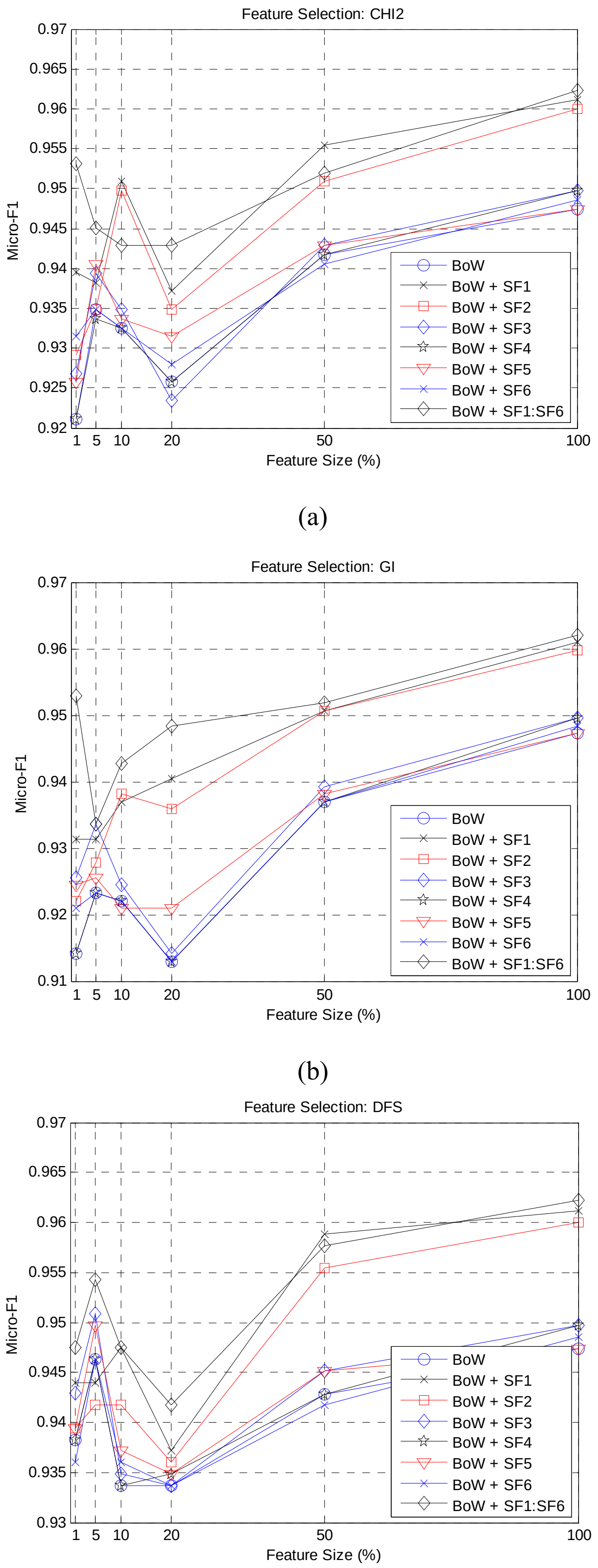


(a)

(b)

(c)

Figure 5.3. kNN classification results for English dataset with (a) CHI2 (b) GI (c) DFS based features



(a)

(b)

(c)

Figure 5.4. SVM classification results for English dataset with (a) CHI2 (b) GI (c) DFS based features

In case of English messages, the highest Micro-F1 score was around 0.96. This value was achieved using SF1:SF6 and 100% of BoW features, which were together applied on SVM classifier. Since all BoW features were employed to attain the highest score, no particular feature selection method was superior to another. In contrast, the maximum score achieved by kNN classifier was around 0.91 with the combination of SF4 and just 1% of BoW features selected by DFS.

In addition to the classification performance, dimension reduction rate is another important aspect of recognition process. Consequently, an analysis for dimension reduction was also carried. In order to compare efficacy of the feature combinations in terms of dimension reduction rate and Micro-F1 values, a dimension reduction (DR) scoring scheme (Gunal and Edizkan, 2008) was adopted for this work.

Since the classification results of SVM classifier were better than that of kNN in all cases as illustrated by Figures 5.1 through 5.4, the scores attained only by SVM were considered during this analysis. Top-5 DR scores for both datasets were computed and listed in Table 5.7 and Table 5.8.

Table 5.7. Top-5 results of dimension reduction analysis for Turkish dataset

No	DR Score	Feature Set	Feature Selection
1	21.913	BoW + SF2	DFS
2	21.870	BoW + SF1:SF6	DFS
3	21.866	BoW + SF1:SF6	GI
4	21.815	BoW + SF6	DFS
5	21.814	BoW + SF4	DFS

Table 5.8. Top-5 results of dimension reduction analysis for English dataset

No	DR Score	Feature Set	Feature Selection
1	21.871	BoW + SF1:SF6	CHI2
2	21.838	BoW + SF1:SF6	GI
3	21.814	BoW + SF1:SF6	DFS
4	21.720	BoW + SF1	DFS
5	21.693	BoW + SF3	DFS

One can easily note from this table that the feature set (BoW + SF2) and DFS based selection method surpass the other combinations for Turkish messages. DR performance of the feature set (BoW + SF1:SF6) follows the feature set (BoW + SF2) with different feature selection metrics. In case of English messages, on the other hand, CHI2 based selection method replaces DFS whereas the feature set is (BoW + SF1:SF6). The results show that the feature set (BoW + SF2) and DFS based selection method surpasses the others in terms of accuracy and dimension reduction for Turkish messages. Although highest DR performance is obtained with CHI2 for English messages, DR performance of DFS is better than CHI2 and GI in most of the cases.

5.3. Conclusions

The impact of various feature extraction and selection methodologies on SMS spam filtering in Turkish and English languages were investigated. A number of SFs adopted for the spam problem were used together with well-known BoW features. In the meantime, CHI2, GI, and DFS based feature selection methods were employed to define the discriminative BoWs. A thorough experimental analysis indicated that the combinations of BoW and SFs, rather than BoW features alone, provide better classification performance in both languages most of the time. The experimental results show that contribution of SF2 is more obvious than the other structural features for Turkish messages. Therefore, it is said that number of terms feature is highly discriminative for Turkish SMS messages. The main reason behind this consequence is that spam messages may contain more terms than legitimate ones in general for Turkish SMS messages. In case of English SMS messages, highest accuracies were obtained with the help of BoW features and combination of all structural features. Although performances of feature selection methods are closer to each other, DFS surpasses the others in terms of accuracy and dimension reduction in most of the cases. Since Turkish and English are the leading examples of agglutinative and non-agglutinative languages respectively, the outcome of this work can be an indicator for the other languages with similar characteristics as well.

6. IMPACT OF PREPROCESSING

While it is verified that the feature extraction (Gunal et al., 2006; Yuanchun and Ying, 2011), feature selection (Rocha and Cobo, 2011; Feng et al., 2012; Uysal and Gunal, 2012), and classification methods (Joachims, 1998; Peng and Huang, 2007; Tan et al., 2011) have substantial impact on the success of text classification process, the preprocessing step may also influence this success noticeably. Common behavior in text classification studies is to apply alphabetic tokenization, stop-word removal, lowercase conversion and stemming without deeply examining their contributions to classification accuracy. Few researchers have analyzed the influence of preprocessing tasks on text classification at some depth. For instance, effectiveness of stop-word removal and stemming are investigated for English news datasets in (Song et al., 2005). The effects of lemmatization, stemming, and stop-word removal are examined on English and Czech datasets in (Toman et al., 2006). The use of stop-word removal, stemming and different tokenization schemes on spam e-mail filtering are analyzed in (Méndez et al., 2006). Furthermore, the influence of preprocessing tasks including tokenization, stop-word removal, and stemming are studied on trimmed versions of Reuters 21578, Newsgroups and Springer in (Pomikálek and Rehurek, 2007). The impact of preprocessing tasks such as tokenization, lowercase conversion, stop-word removal, stemming, and pruning on the classification of MEDLINE documents is investigated in (Gonçalves et al., 2010). Besides, the effect of stemming on Arabic documents is analyzed in (Duwairi et al., 2009; Said et al., 2009). The impact of stemming and stop-word removal on Turkish texts is evaluated in (Torunoglu et al., 2011) using self-compiled newspaper articles from the Internet. The influence of stemming on Turkish news articles is studied in (Toraman et al., 2011) as well.

The impact of widely used preprocessing tasks including tokenization, stop-word removal, lowercase conversion, and stemming are investigated as a part of this dissertation in a completely different manner than that of the abovementioned studies, such that all possible combinations of those preprocessing tasks are considered comparatively in two different languages, namely Turkish and English, and on two different text domains, namely news and e-mails. In this way, contribution of the regarding preprocessing tasks to the classification success at various feature dimensions, possible interactions among these tasks, and also the dependency of these tasks to the language, and domain studied on are extensively assessed. The preprocessing tasks analyzed and corresponding experimental results are shown in the following subsections.

6.1. Analysis of the Preprocessing Methods

In this part of the dissertation, all possible combinations of the preprocessing methods are considered as below so that possible interactions between the preprocessing tasks can be revealed.

- Tokenization is either alphanumeric or alphabetic.
- Stop-word removal is either ON or OFF; that is, stop-words are either eliminated or kept within text.
- Lowercase conversion is either ON or OFF; that is, terms are either converted to lowercase or kept in their original forms.
- Stemming is either ON or OFF; that is, terms are either reduced to their root forms or kept in their inflected forms.

Thus, 16 different combinations are obtained as listed in Table 6.1.

Table 6.1. Combinations of the preprocessing methods

No	Tokenization (TK) Alphanumeric (0) / Alphabetic (1)	Stop-word Removal (SR) OFF (0) / ON (1)	Lowercase Conversion (LC) OFF (0) / ON (1)	Stemming (ST) OFF (0) / ON (1)
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
…	…	…	...	…
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

6.2. Experimental Work

During the experiments, all possible combinations of the four preprocessing tasks including tokenization, stopword removal, lowercase conversion, and stemming were considered. Various feature sizes including 10, 20, 50, 100, 200, 500, 1000, and 2000 were investigated in the part of the work so that the impact of preprocessing can be comparatively observed within a wide range of feature dimensions. These features were determined using the CHI2, which is stated in Section 4. In the experiments, four datasets consisting of spam e-mail and news datasets are used. While the first dataset is a Turkish spam e-mail dataset consisting of spam and legitimate e-mails (Ergin et al., 2012), the second one is an English spam e-mail dataset, which is shaped using a subset of well-known Enron dataset (Metsis et al., 2006). The third dataset consist of Turkish news documents and is a subset of Milliyet news collection (Can et al., 2008). Characteristics of these three datasets are shown in Tables 9.2-9.4. The fourth dataset is top-10 classes of the celebrated Reuters-21578 ModApte split, which was previously presented in Table 3.2.

Table 6.2. Turkish e-mail dataset

Class No	Class Label	# Training Samples	# Testing Samples
1	spam	300	100
2	legitimate	300	100

Table 6.3. English e-mail dataset

Class No	Class Label	# Training Samples	# Testing Samples
1	spam	300	100
2	legitimate	300	100

Table 6.4. Turkish news dataset

Class No	Class Label	# Training Samples	# Testing Samples
1	spor	2877	1087
2	yazar	1650	719
3	yaşam	538	179
4	ekonomi	433	149
5	siyaset	389	189
6	magazin	369	117
7	dünya	347	131
8	astro	197	89
9	tv	212	71
10	sanat	121	38

In the experiments, widely-known Micro-F1 success measure is used. The results of the experimental analysis on these four datasets are illustrated in Figures 6.1-6.4.

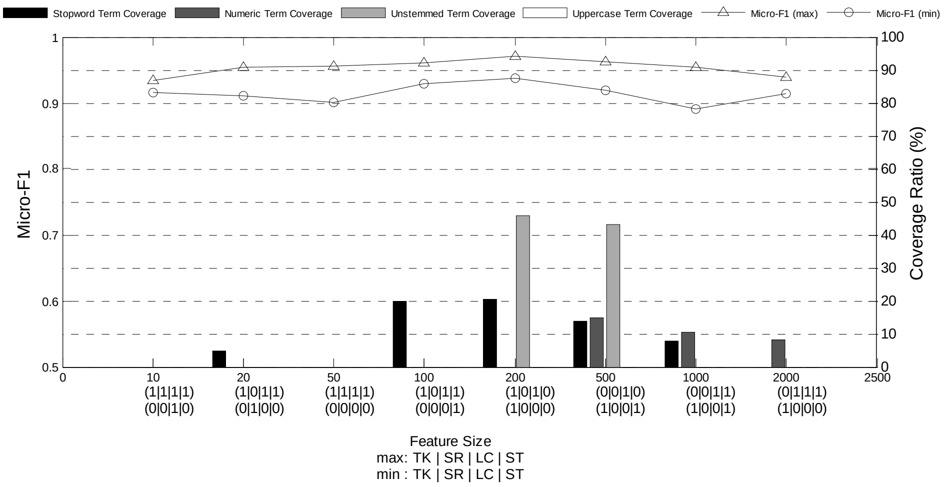


Figure 6.1. Experimental results for Turkish e-mail dataset

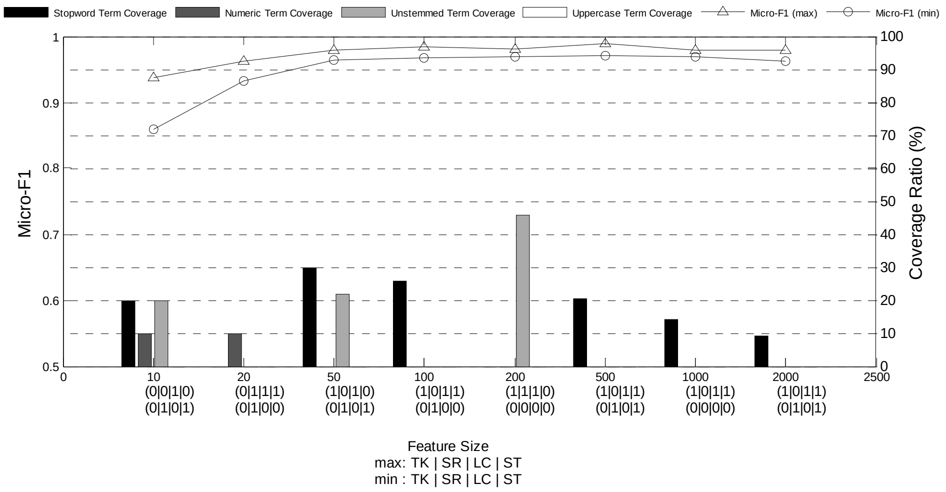


Figure 6.2. Experimental results for English e-mail dataset

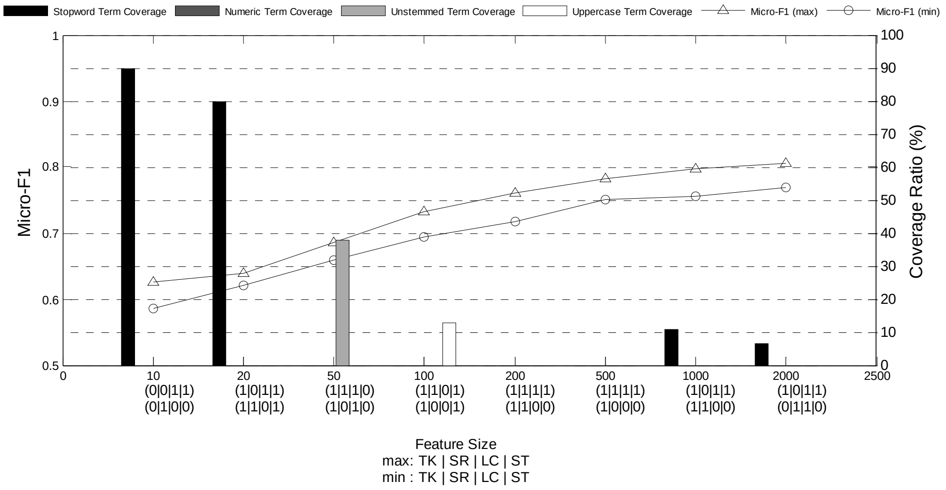


Figure 6.3. Experimental results for Turkish news dataset

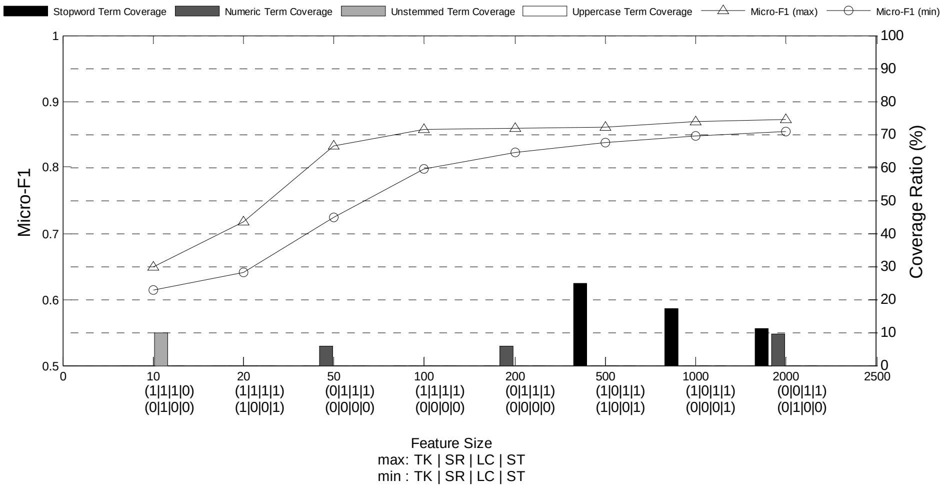


Figure 6.4. Experimental results for English news dataset

Figures 6.1-6.4 include the plots of the maximum and minimum Micro-F1 scores and the corresponding combinations of the preprocessing tasks at different feature sizes. In this way, the best and the worst cases indicating the impact of preprocessing are highlighted. Considering the maximum Micro-F1 scores, the figures also provide bar charts for coverage ratios of the terms arisen from the regarding preprocessing tasks to the selected terms at different feature sizes. In order to clarify the interpretation of those figures, some specific examples are provided as follows:

- In Turkish e-mail dataset, the maximum Micro-F1 score is 0.9713. This score is attained when the feature size is 200 and the preprocessing combination is (TK: 1 | SR: 0 | LC: 1 | ST: 0); that is, tokenization is alphabetic, stop-word removal is OFF, lowercase conversion is ON, and stemming is OFF. In this case, stop-word term coverage and unstemmed term coverage ratios are around 20% and 45%, respectively. In other words, 20% of 200 selected terms are the stop-words whereas 45% of these 200 selected terms are unstemmed terms as well.
- Additionally, the minimum Micro-F1 score is 0.8913 for Turkish e-mail dataset. This score is obtained when the feature size is 1000 and the preprocessing combination is (TK: 1 | SR: 0 | LC: 0 | ST: 1); that is, tokenization is alphabetic, stop-word removal is OFF, lowercase conversion is OFF, and stemming is ON. It should be reminded that, in case of minimum Micro-F1 scores, coverage ratios were not taken into consideration; therefore, they were not displayed in the figures.

Based on all the information provided in these figures, the impact of preprocessing were analyzed according to several aspects including accuracy, domain, language, and feature size.

6.2.1. Accuracy Analysis

In this part, Micro-F1 scores attained by all 16 combinations of the preprocessing tasks were measured to assess the impact of preprocessing in terms of accuracy. The highest and the lowest Micro-F1 scores at each feature size together with the corresponding preprocessing combinations were shown in Figures 6.1-6.4. The maximum Micro-F1 scores among all feature sizes and the corresponding preprocessing combinations are listed in Table 6.5 for each dataset with the help of those figures.

Table 6.5. Max. Micro-F1 scores and the corresponding preprocessing tasks

Dataset	Max. Micro-F1	Preprocessing Tasks
Turkish e-mail	0.9713	TK: 1 | SR: 0 | LC: 1 | ST: 0
English e-mail	0.9888	TK: 1 | SR: 0 | LC: 1 | ST: 1
Turkish news	0.8061	TK: 1 | SR: 0 | LC: 1 | ST: 1
English news	0.8719	TK: 0 | SR: 0 | LC: 1 | ST: 1

Considering all four datasets, the difference between the highest and the lowest Micro-F1 scores at each feature size for all preprocessing combinations ranged from 0.0113 to 0.1084. More specifically, the difference was between 0.0175 – 0.0625 in Turkish e-mail dataset, between 0.0113 – 0.0787 in English e-mail dataset, between 0.0195 – 0.044 in Turkish news dataset, and between 0.0179 – 0.1084 in English news dataset. The amount of differences in accuracies confirms that the appropriate preprocessing combinations depending on the domain and language may improve accuracy considerably. In the meantime, inappropriate preprocessing combinations may degrade accuracy as well.

The impact of preprocessing was also statistically analyzed using two-tailed paired t-test over the highest and lowest Micro-F1 scores at each feature size. *P*-values were obtained as (0.000138, 0.016818, 0.000007, and 0.003908) for Turkish e-mail, English e-mail, Turkish news, and English news datasets, respectively. The result for English e-mail dataset was statistically significant with a significance level of 0.05 whereas the remaining three datasets obtained a significance level of 0.01.

6.2.2. Domain and Language Analysis

In this part, the impact of preprocessing was evaluated for every domain and language considering the maximum Micro-F1 scores at each case.

Tokenization type in e-mail domain for both languages was alphabetic; however, news domain involved alphabetic tokenization in Turkish and alphanumeric tokenization in English. To confirm the impact of alphanumeric tokenization, numeric term coverage within the selected feature set of English news dataset was also computed. The coverage ratio was around 10%; in other words, 10% of the selected terms contained numeric characters.

Stop-word removal is not applied in any of the domains and languages. In order to verify the impact of stop-words on classification success, stop-word coverage within the selected feature sets of each dataset was also computed. The coverage ratios were found as 20.50%, 20.60%, 6.50%, and 11.25% for Turkish e-mail, English e-mail, Turkish news, and English news datasets, respectively. One can see from those ratios that the presence of the stop-words within the selected terms is very obvious in each domain and language. This finding is really remarkable bearing in mind that most of the text classification studies in the literature remove stop-words directly by assuming them irrelevant.

Lowercase conversion is active in both domains and languages. In other words, all characters should be converted to lowercase without dependency to domain or language.

Stemming is required in news domain for both languages; on the contrary, it is not applied in Turkish e-mail domain whereas it is necessary for English e-mails. Again, to validate the impact of not applying the stemming in Turkish e-mail domain, unstemmed term coverage within the selected feature set was computed and found to be around 45%. In other words, almost half of the selected terms consist of unstemmed terms.

One can conclude that stop-words should not be removed and characters should be converted to lowercase without dependency to domain or language. However, tokenization type and stemming status may change depending on the domain and language. Reminding that the e-mail datasets are binary and balanced whereas the news datasets are multi-class and imbalanced, all the statements above may be generalized for different class distributions (balanced vs. imbalanced) and different numbers of classes (binary vs. multi-class) as well.

6.2.3. Feature Size Analysis

In this part, the impact of preprocessing was evaluated in terms of dimension reduction. For this purpose, the preprocessing tasks providing the highest Micro-F1 scores at minimum feature size for each dataset were taken into consideration as listed in Table 6.6.

In e-mail domain, as a common behavior in both languages, lowercase conversion was applied. Status of the remaining preprocessing tasks; however, varied depending on the language.

Table 6.6. Preprocessing tasks providing the highest accuracy at min. feature sizes

Dataset	Min. Feature Size	Preprocessing Tasks
Turkish e-mail	10	TK: 1 | SR: 1 | LC: 1 | ST: 1
English e-mail	10	TK: 0 | SR: 0 | LC: 1 | ST: 0
Turkish news	10	TK: 1 | SR: 0 | LC: 1 | ST: 1
English news	10	TK: 1 | SR: 1 | LC: 1 | ST: 0

In news domain, alphabetic tokenization and lowercase conversion were common preprocessing tasks in both languages whereas status of stop-word removal and stemming were opposite for each language. While stop-word removal was applied in English news, stemming was required for Turkish news.

For Turkish language, stop-word removal was applied only on e-mail domain while alphabetic tokenization, lowercase conversion, and stemming were commonly applied in both domains. Stop-word coverage ratio within the selected feature set of Turkish news dataset was computed as 90%. Hence, it is obvious that the stop-words have a dominant impact among all terms at minimum feature dimension of Turkish news domain only.

For English language, lowercase conversion was applied, but stemming was not active in both domains. Unstemmed term coverage ratios within the selected feature sets were 20% and 10% for e-mail and news datasets in English, respectively. On the other hand, the status of tokenization type and stop-word removal were opposite for each language. While alphabetic tokenization was applied in English news domain, stop-words were kept in English e-mail domain. Stop-word coverage ratio within the selected feature set of English e-mail dataset was computed as 20% and numeric term coverage ratio was found as 10% in English news dataset as well.

6.2.4. Maximum Accuracy versus Minimum Feature Size

In this section, the preprocessing tasks, which provided the maximum Micro-F1 scores, were compared to the ones providing the highest Micro-F1 scores at minimum feature size for each domain and language. The comparison is listed in Table 6.7.

Table 6.7. Preprocessing tasks: Maximum accuracy vs. minimum feature size

Dataset	Preprocessing Tasks (Max. Accuracy)	Preprocessing Tasks (Min. Feature Size)
Turkish e-mail	TK: 1 | SR: 0 | LC: 1 | ST: 0	TK: 1 | SR: 1 | LC: 1 | ST: 1
English e-mail	TK: 1 | SR: 0 | LC: 1 | ST: 1	TK: 0 | SR: 0 | LC: 1 | ST: 0
Turkish news	TK: 1 | SR: 0 | LC: 1 | ST: 1	TK: 1 | SR: 0 | LC: 1 | ST: 1
English news	TK: 0 | SR: 0 | LC: 1 | ST: 1	TK: 1 | SR: 1 | LC: 1 | ST: 0

It is obvious from the table that lowercase conversion is the only preprocessing task that is common in all cases. In other words, lowercase conversion should be applied to achieve either maximum accuracy or minimum feature size with the highest accuracy for all domains and languages. Another common behavior was related to Turkish language such that alphabetic tokenization should be applied in Turkish, regardless of the domain, to achieve either maximum accuracy or minimum feature size. No other common behavior was observed related to the remaining preprocessing tasks for any domain or language.

6.3. Conclusions

Within the scope of the dissertation, the influence of widely used preprocessing tasks on text classification was thoroughly examined in two different domains and languages. The examination was carried out using all possible combinations of the preprocessing tasks by considering various aspects such as accuracy, domain, language, and dimension reduction. Extensive experimental analysis revealed that appropriate combinations of preprocessing tasks depending on the domain and language may provide a significant improvement on classification accuracy whereas inappropriate combinations may degrade accuracy as well. Consequently, preprocessing step in text classification is as important as feature extraction, feature selection, and classification steps. Although there are particular preprocessing tasks that improve classification success in terms of accuracy and dimension reduction regardless of domain and language, there is no unique combination of preprocessing tasks providing successful classification results for every domain and language studied on. Therefore, for a text classification problem on any domain and in any language, researchers should carefully analyze all possible combinations of the tasks rather than completely enabling or disabling them. Otherwise, classification results may significantly differ. Another interesting finding of this work was the importance of stop-words while most of the text classification studies in the literature assume the stop-words irrelevant. Since all four datasets studied on this study as a part of the dissertation have distinct characteristics in terms of domain, language, class distribution, and number of classes, the outcome of this study may be generalized for the other text collections as well.

7. CONCLUSIONS AND FUTURE WORK

In this dissertation, various solutions are proposed to overcome high dimensionality and misclassification concerns of the text classification problems.

A novel feature selection method, namely DFS, was developed within the scope of the dissertation. This novel method was proved to be effective in terms of accuracy, dimension reduction rate, and processing time. It has a comparative or even better performance with the other state-of-the-art feature selection metrics in text classification literature.

The GALSF method consisting of feature selection and feature transformation stages was proposed for text classification. In this method, the singular vectors with small singular values may also be used for projection whereas the vectors with large singular values may be eliminated to obtain better discrimination, as well. The performance of the proposed method GALSF was analyzed with extensive experiments. Experimental results reveal that GALSF outperforms both LSI and individual performance of feature selection methods in terms of accuracy. As well as GALSF generally reduces the dimension approximately to the half, total dimension reduction rate is about the half of the reduced dimension by feature selection methods before. Therefore, GALSF not only increases accuracy of text classification but also provides a reasonable dimension reduction performance. It can be also concluded that the singular vectors corresponding to small singular values may be useful to obtain a projection providing better discrimination whereas the vectors with large singular values may be useless for this purpose.

The impact of various feature extraction and selection methodologies on SMS spam filtering in Turkish and English languages were investigated. For this purpose, six different structural features adopted for the spam e-mail problem were used for SMS spam filtering. Experimental analysis reveals that combination of BoW and SFs perform better than the individual performance of BoW in all cases. The impact of SF2 is more obvious for Turkish SMS messages. It can be concluded that number of terms in a message is a discriminative feature for Turkish messages. This finding can be explained with the fact that spam messages may contain more terms than legitimate ones for Turkish SMS messages in general. Since Turkish and English are the leading examples of agglutinative and non-agglutinative languages respectively, the outcome of this work can be an indicator for the other languages with similar characteristics as well.

The influence of widely used preprocessing tasks on text classification was thoroughly analyzed with extensive experiments. All combinations of the four preprocessing tasks are used to this analysis in a reduced dimension. For this purpose, a widely-known feature selection metric is employed with different feature sizes. Consequently, preprocessing step in text classification is found as an important step as feature extraction, feature selection, and classification steps. Although using appropriate preprocessing steps improve accuracy, there is no unique combination of preprocessing tasks providing successful classification results for every domain and language studied on. The importance of stop-words, generally assumed as irrelevant, is also an interesting finding.

As a consequence, all of the proposed solutions provide improvement on performance of text classification in terms of accuracy and/or dimension reduction. As potential future works, performance of DFS can be evaluated on other text classification domains, adaptation of DFS to the other pattern classification problems is also possible, and GALSF can be applied to different tasks related to the text classification where LSI was applied before.

REFERENCES

Alhabashneh, O., Iqbal, R., Shah, N., Amin, S. and James, A. (2011), "Towards the development of an integrated framework for enhancing enterprise search using latent semantic indexing," *Proceedings of the 9th International Conference on Conceptual Structures for Discovering Knowledge*, Derby, UK, 346-352.

Almeida, T. A., Hidalgo, J. M. G. and Yamakami, A. (2011), "Contributions to the study of SMS spam filtering: New collection and results," *Proceedings of the 11th ACM Symposium on Document Engineering*, Mountain View, USA, 259-262.

Asuncion, A. and Newman, D. J. (2007). "UCI Machine Learning Repository." Retrieved January 2013, from http://www.ics.uci.edu/~mlearn/MLRepository.html.

Barros, A. S. and Rutledge, D. N. (1998), "Genetic algorithm applied to the selection of principal components," *Chemometrics and Intelligent Laboratory Systems*, **40**(1), 65-81.

Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H. C. and Vursavas, O. M. (2008), "Information retrieval on Turkish texts," *Journal of the American Society for Information Science and Technology*, **59**(3), 407-421.

Chen, J., Huang, H., Tian, S. and Qu, Y. (2009), "Feature selection for text classification with Naïve Bayes," *Expert Systems with Applications*, **36**(3), 5432-5435.

Chen, Y.-T. and Chen, M. C. (2011), "Using chi-square statistics to measure similarities for text categorization," *Expert Systems with Applications*, **38**(4), 3085-3090.

Chen, Z. P. and Lu, K. (2006), "A preprocess algorithm of filtering irrelevant information based on the minimum class difference," *Knowledge-Based Systems*, **19**(6), 422-429.

Cheng, N., Chandramouli, R. and Subbalakshmi, K. P. (2011), "Author gender identification from text," *Digital Investigation*, **8**(1), 78-88.

Cormack, G., Hidalgo, J. M. G. and Sanz, E. P. (2007), "Spam filtering for short messages," *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, Lisbon, Portugal, 313-320.

Delany, S. J., Buckley, M. and Greene, D. (2012), "SMS spam filtering: Methods and data," *Expert Systems with Applications*, **39**(10), 9899-9908.

Drucker, H., Wu, D. and Vapnik, V. (1999), "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, **10**(5), 1048 - 1054.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998), "Inductive learning algorithms and representations for text categorization," *Proceedings of the 7th International Conference on Information and Knowledge Management*, Bethesda, Maryland, United States, 148-155.

Duwairi, R., Al-Refai, M. N. and Khasawneh, N. (2009), "Feature reduction techniques for Arabic text categorization," *Journal of the American Society for Information Science and Technology*, **60**(11), 2347-2352.

Ergin, S., Gunal, E. S., Yigit, H. and Aydin, R. (2012), "Turkish anti-spam filtering using binary and probabilistic models," *AWERProcedia Information Technology and Computer Science*, **1**, 1007-1012.

ETSI (1992). Technical realization of the Short Message Service - Point to Point. GSM 03.40.

Fausett, L. (1994), "*Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*," Prentice-Hall, Inc.

Feng, G., Guo, J., Jing, B. Y. and Hao, L. (2012), "A Bayesian feature selection paradigm for text classification," *Information Processing & Management*, **48**(2), 283-302.

Forman, G. (2003), "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, **3**, 1289-1305.

Ghiassi, M., Olschimke, M., Moon, B. and Arnaudo, P. (2012), "Automated text classification using a dynamic artificial neural network model," *Expert Systems with Applications*, **39**(12), 10967-10976.

Goldberg, D. E. (1989), "*Genetic Algorithms in Search, Optimization and Machine Learning*," Addison-Wesley Longman Publishing Co., Inc.

Golub, K. (2006), "Automated subject classification of textual web documents," *Journal of Documentation*, **62**(3), 350-371.

Gonçalves, C. A., Gonçalves, C. T., Camacho, R. and Oliveira, E. C. (2010), "The impact of pre-processing on the classification of MEDLINE documents," *Proceedings of the 10th International Workshop on Pattern Recognition in Information Systems*, Funchal, Madeira, Portugal, 53-61.

Gud, A. and Shatovska, T. (2009), "Forecasting and discriminant analysis," *Proceedings of the CAD Systems in Microelectronics (CADSM)*, Lviv-Polyana, Ukraine, 536-538.

Gulmezoglu, M. B., Dzhafarov, V. and Barkana, A. (2001), "The common vector approach and its relation to principal component analysis," *IEEE Transactions on Speech and Audio Processing*, **9**(6), 655-662.

Gunal, S. (2012), "Hybrid feature selection for text classification," *Turkish Journal of Electrical Engineering & Computer Sciences*, **20**(sup.2), 1296-1311.

Gunal, S. and Edizkan, R. (2008), "Subspace based feature selection for pattern recognition," *Information Sciences*, **178**(19), 3716-3726.

Gunal, S., Ergin, S., Gulmezoglu, M. B. and Gerek, O. N. (2006), "On feature extraction for spam e-mail detection," *Lecture Notes in Computer Science*, **4105**, 635-642.

Gunal, S., Gerek, O. N., Ece, D. G. and Edizkan, R. (2009), "The search for optimal feature set in power quality event classification," *Expert Systems with Applications*, **36**(7), 10266-10273.

Guyon, I. and Elisseeff, A. (2003), "An introduction to variable and feature selection," *Journal of Machine Learning Research*, **3**, 1157-1182.

Hidalgo, J. M. G., Bringas, G. C., Sanz, E. P. and Garcia, F. C. (2006), "Content based SMS spam filtering," *Proceedings of the ACM Symposium on Document Engineering*, Amsterdam, The Netherlands, ACM, 107-114.

Hsu, C.-W. and Lin, C.-J. (2002), "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, **13**(2), 415-425.

Joachims, T. (1997), "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," *Proceedings of the 14th International Conference on Machine Learning*, Nashville, USA, 143-151.

Joachims, T. (1998), "Text categorization with support vector machines: Learning with many relevant features," *Proceedings of the 10th European Conference on Machine Learning*, Chemnitz, Germany, 137-142.

Joe, I. and Shim, H. (2010), "An SMS spam filtering system using support vector machine," *Lecture Notes in Computer Science*, **6485**, 577-584.

Johnson, D. E., Oles, F. J., Zhang, T. and Goetz, T. (2002), "A decision-tree-based symbolic rule induction system for text categorization," *IBM Systems Journal*, **41**(3), 428-437.

Kohavi, R. and John, G. H. (1997), "Wrappers for feature subset selection," *Artificial Intelligence*, **97**, 273-324.

Kontostathis, A. and Pottenger, W. M. (2006), "A framework for understanding Latent Semantic Indexing (LSI) performance," *Information Processing & Management*, **42**(1), 56-73.

Kucukyilmaz, T., Cambazoglu, B. B., Aykanat, C. and Can, F. (2006), "Chat mining for gender prediction," *Proceedings of the 4th international conference on Advances in Information Systems*, 274-283.

Kumar, M. A. and Gopal, M. (2010), "A comparison study on multiple binary-class SVM methods for unilabel text categorization," *Pattern Recognition Letters*, **31**(11), 1437-1444.

Lee, C. and Lee, G. G. (2006), "Information gain and divergence-based feature selection for machine learning-based text categorization," *Information Processing & Management*, **42**(1), 155-165.

Liu, H., Sun, J., Liu, L. and Zhang, H. (2009a), "Feature selection with dynamic mutual information," *Pattern Recognition*, **42**(7), 1330-1339.

Liu, T., Chen, Z., Zhang, B., Ma, W.-y. and Wu, G. (2004), "Improving text classification using local latent semantic indexing," *Proceedings of the 4th IEEE International Conference on Data Mining*, Brighton, UK, 162-169.

Liu, Y., Loh, H. T. and Sun, A. X. (2009b), "Imbalanced text classification: A term weighting approach," *Expert Systems with Applications*, **36**(1), 690-701.

Lopes, C., Cortez, P., Sousa, P., Rocha, M. and Rio, M. (2011), "Symbiotic filtering for spam email detection," *Expert Systems with Applications*, **38**(8), 9365-9372.

Maks, I. and Vossen, P. (2012), "A lexicon model for deep sentiment analysis and opinion mining applications," *Decision Support Systems*, **53**(4), 680-688.

Manning, C. D., Raghavan, P. and Schutze, H. (2008), "*Introduction to Information Retrieval*," Cambridge University Press, New York, USA.

Méndez, J. R., Iglesias, E. L., Fdez-Riverola, F., Díaz, F. and Corchado, J. M. (2006), "Tokenising, stemming and stopword removal on anti-spam filtering domain," *Proceedings of the 11th Spanish Association Conference on Current Topics in Artificial Intelligence*, Santiago de Compostela, Spain, 449-458.

Meng, J. N., Lin, H. F. and Yu, Y. H. (2011), "A two-stage feature selection method for text categorization," *Computers & Mathematics with Applications*, **62**(7), 2793-2800.

Mengle, S. S. R. and Goharian, N. (2009), "Ambiguity measure feature-selection algorithm," *Journal of the American Society for Information Science and Technology*, **60**(5), 1037-1050.

Metsis, V., Androutsopoulos, I. and Paliouras, G. (2006), "Spam filtering with naive Bayes – which naive Bayes?," *Proceedings of the 3rd Conference on Email and Anti-Spam*, 28-69.

Mladenic, D. and Grobelnik, M. (2003), "Feature selection on hierarchy of Web documents," *Decision Support Systems*, **35**(1), 45-87.

Na, J.-C. and Thet, T. T. (2009), "Effectiveness of web search results for genre and sentiment classification," *Journal of Information Science*, **35**(6), 709-726.

Nuruzzaman, M. T., Lee, C. and Choi, D. (2011), "Independent and personal SMS spam filtering," *Proceedings of the IEEE 11th International Conference on Computer and Information Technology*, Paphos, Cyprus, 429-435.

Ogura, H., Amano, H. and Kondo, M. (2009), "Feature selection with a measure of deviations from Poisson in text categorization," *Decision Support Systems*, **36**(3), 6826-6832.

Ozel, S. A. (2011), "A Web page classification system based on a genetic algorithm using tagged-terms as features," *Expert Systems with Applications*, **38**(4), 3407-3415.

Peng, F. and Huang, X. (2007), "Machine learning for Asian language text classification," *Journal of Documentation*, **63**(3), 378-397.

Pomikálek, J. and Rehurek, R. (2007), "The influence of preprocessing parameters on text categorization," *International Journal of Applied Science, Engineering and Technology*, **4**, 430-434.

Porter, M. F. (1980), "An algorithm for suffix stripping," *Program*, **14**(3), 130−137.

Puniškis, D. and Laurutis, R. (2007), "Behavior statistic based neural net anti-spam filters," *Electronics and Electrical Engineering*, **6**(78), 35-38.

Puniškis, D., Laurutis, R. and Dirmeikis, R. (2006), "An artificial neural nets for spam e-mail recognition," *Electronics and Electrical Engineering*, **5**(69), 73-76.

Rocha, R. and Cobo, A. (2011), "Feature selection strategies for automated classification of digital media content," *Journal of Information Science*, **37**(4), 418-428.

Saeys, Y., Inza, I. and Larranaga, P. (2007), "A review of feature selection techniques in bioinformatics," *Bioinformatics*, **23**(19), 2507-2517.

Said, D. A., Wanas, N. M., Darwish, N. M. and Hegazy, N. H. (2009), "A study of text preprocessing tools for Arabic text categorization," *Proceedings of the 2nd International Conference on Arabic Language*, Cairo, Egypt, 230-236.

Salton, G., Wong, A. and Yang, C. S. (1975), "A vector space model for automatic indexing," *Communications of the ACM*, **18**(11), 613-620.

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y. and Wang, Z. (2007), "A novel feature selection algorithm for text categorization," *Expert Systems with Applications*, **33**(1), 1-5.

Sohn, D.-N., Lee, J.-T., Han, K.-S. and Rim, H.-C. (2012), "Content-based mobile spam classification using stylistically motivated features," *Pattern Recognition Letters*, **33**(3), 364-369.

Song, F. X., Liu, S. H. and Yang, J. Y. (2005), "A comparative study on text representation schemes in text categorization," *Pattern Analysis and Applications*, **8**(1-2), 199-209.

Tan, S., Wang, Y. and Wu, G. (2011), "Adapting centroid classifier for document categorization," *Expert Systems with Applications*, **38**(8), 10264-10273.

Theodoridis, S. and Koutroumbas, K. (2008), "*Pattern Recognition*," Academic Press.

Toman, M., Tesar, R. and Jezek, K. (2006), "Influence of word normalization on text classification," *Proceedings of the 1st International Conference on Multidisciplinary Information Sciences & Technologies*, Merida, Spain, 354-358.

Toraman, C., Can, F. and Kocberber, S. (2011), "Developing a text categorization template for Turkish news portals," *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey, 379-383.

Torunoglu, D., Cakirman, E., Ganiz, M. C., Akyokus, S. and Gurbuz, M. Z. (2011), "Analysis of preprocessing methods on classification of Turkish texts," *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 112-117.

Uguz, H. (2011), "A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm," *Knowledge-Based Systems*, **24**(7), 1024-1032.

Uysal, A. K. and Gunal, S. (2012), "A novel probabilistic feature selection method for text classification," *Knowledge-Based Systems*, **36**, 226-235.

Uysal, A. K., Gunal, S., Ergin, S. and Gunal, E. S. (2012a), "Detection of SMS spam messages on mobile phones," *Proceedings of the IEEE 20th Signal Processing and Communications Applications Conference*, Fethiye, Mugla, Turkey, 1-4.

Uysal, A. K., Gunal, S., Ergin, S. and Gunal, E. S. (2013), "The impact of feature extraction and selection on SMS spam filtering," *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering),* in press.

Uysal, A. K., Gunal, S., Semih Ergin and Gunal, E. S. (2012b), "A novel framework for SMS spam filtering," *Proceedings of the IEEE International Symposium on Innovations in Intelligent Systems and Applications*, Trabzon, Turkiye, 1-4.

Wang, C., Zhang, Y., Chen, X., Liu, Z., Shi, L., Chen, G., Qiu, F., Ying, C. and Lu, W. (2010), "A behavior-based SMS antispam system," *IBM Journal of Research and Development*, **54**(6), 651-666.

Wang, W. and Yu, B. (2009), "Text categorization based on combination of modified back propagation neural network and latent semantic analysis," *Neural Computing & Applications*, **18**(8), 875-881.

Wu, K., Lu, B. L., Uchiyama, M. and Isahara, H. (2007), "A probabilistic approach to feature selection for multi-class text categorization," *Lecture Notes in Computer Science*, **4491**, 1310-1317.

Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A. and Naik, V. (2011), "SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering," *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, Phoenix, AR, USA, 1-6.

Yang, H. and King, I. (2009), "Sprinkled latent semantic indexing for text classification with background knowledge," *Advances in Neuro-Information Processing*, **5507**, 53-60.

Yang, J., Liu, Y., Liu, Z., Zhu, X. and Zhang, X. (2011), "A new feature selection algorithm based on binomial hypothesis testing for spam filtering," *Knowledge-Based Systems*, **24**(6), 904-914.

Yang, X. Q., Sun, N., Sun, T. L., Cao, X. Y. and Zheng, X. J. (2009), "The application of latent semantic indexing and ontology in text classification," *International Journal of Innovative Computing Information and Control*, **5**(12A), 4491-4499.

Yang, Y. (1995), "Noise reduction in a statistical approach to text categorization," *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WI, USA, 256-263.

Yang, Y. and Pedersen, J. O. (1997), "A comparative study on feature selection in text categorization," *Proceedings of the 14th International Conference on Machine Learning*, Nashville, TN, USA, 412-420.

Yoon, J. W., Kim, H. and Huh, J. H. (2010), "Hybrid spam filtering for mobile communication," *Computers & Security*, **29**(4), 446-459.

Yu, B., Xu, Z. B. and Li, C. H. (2008), "Latent semantic analysis for text categorization using neural network," *Knowledge-Based Systems*, **21**(8), 900-904.

Yu, B. and Zhu, D. H. (2009), "Combining neural networks and semantic feature space for email classification," *Knowledge-Based Systems*, **22**(5), 376-381.

Yuanchun, Z. and Ying, T. (2011), "A local-concentration-based feature extraction approach for spam filtering," *IEEE Transactions on Information Forensics and Security*, **6**(2), 486-497.

Zemberek. (2013). "Zemberek Natural Language Processing Library for Turkic Languages." Retrieved January 2013, from http://code.google.com/p/zemberek/.

Zhang, W., Yoshida, T. and Tang, X. J. (2011), "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, **38**(3), 2758-2765.

Zheng, W. S., Lai, J. H. and Yuen, P. C. (2005), "GA-Fisher: A new LDA-based face recognition algorithm with selection of principal components," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **35**(5), 1065-1078.