Feature Selection Using Co-occurrence of Terms for Binary Text Classification

Marzieh Vahabi Mashak

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Serhan Çiftçioğlu
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Hakan Altınçay
Supervisor

Examining Committee

1. Prof. Dr. Hakan Altınçay ______________________________

2. Prof. Dr. Hasan Kömürcügil ______________________________

3. Asst. Prof. Dr. Ahmet Ünveren ______________________________

ABSTRACT

In this thesis, term selection for text categorization is addressed. Three widely used schemes are employed for this purpose, namely *Chi-square*(χ^2), *Gini_index* and *Discriminative Power Measure* (DPM). The performances of these schemes are evaluated on Reuters-21578 separately for document frequencies and term frequencies. In summary, utilizing the term frequencies leads to better macro and micro $\mathcal{F}_1 score$ when compared to using only document frequencies.

As an extension to the conventionally used term selection schemes, we studied the use of co-occurrence statistics of different terms for feature selection. More specifically, the idea is to evaluate the discriminative power of having two different terms in the selected list at the same time. In order to achieve this, an iterative scheme is designed where the next term to be included in the selected list is determined by pairwise evaluation of the already selected terms and the candidate terms. For the pairwise evaluation of different terms, novel metrics based on the existing selection schemes are developed. Experimental results have shown that the proposed iterative scheme has the potential to improve the existing schemes.

Keywords: Term Selection, Text Classification, χ^2, $Gini\_index$, DPM, Bag-of-Words.

ÖZ

Bu tezde metin sınıflandırma için kelime seçme konusu ele alınmıştır. Bu amaçla sıklıkla kullanılan Chi-kare (χ^2), $Gini\_indisi$ ve *Ayırıcı Güç Ölçütü* (AGÖ) isimli üç kelime seçme yöntemi kullanılmıştır. Bu metodların başarımları Reuters-21578 verisi üzerinde döküman frekansları ve kelime frekansları kullanılarak incelenmiştir. Kelime frekansları kullanımının döküman frekanslarına göre daha iyi makro ve mikro F_1 skorları sağladığı gözlenmiştir.

Geleneksel olarak kullanılan kelime seçme yöntemlerine iyileştirme olarak, kelimelerin ayni anda bulunma istatistiklerinin kullanımı üzerinde çalışılmıştır. Daha özel olarak belirtecek olursak esas fikir, iki kelimenin ayni anda seçilmiş listede olmasının öneminin dikkate alınmasıdır. Bunu sağlamak için, daha önce seçilen kelimeler ile seçilmeye aday kelimeleri ikili olarak değerlendiren yinelemeli bir yöntem geliştirilmiştir. Farklı kelimelerin ikili değerlendirilmesi için, mevcut seçme yöntemlerini temel alan yeni metrikler geliştirilmiştir. Deneysel sonuçlar, önerilen yinelemeli yaklaşımın mevcut yöntemleri iyileştirme potansiyeline sahip olduğunu göstermiştir.

Anahtar kelimeler: Kelime Seçme, Metin sınıflandırma, χ^2, $Gini\_indisi$, AGÖ, Kelime-sepeti.

This thesis is dedicated to my parents

for their extremely numerous and continual love and support.

ACKNOWLEDGMENT

Foremost, I am deeply indebted to my supervisor, Prof. Dr. Hakan Altınçay. His motivation, patience, support, advice and assistance in all phases of this research were stunning. I appreciate the shared knowledge and experiences by him and am grateful of having opportunity to work under his supervision.

Thanks to the other member of the committee, Prof. Dr. Hasan Kömürcügil and Asst. Prof. Dr. Ahmet Ünveren for all their consideration.

Special and endless thanks go to who continually encourage and support me with their incredible love and patience, my beloved family. Thanks a lot for loving me as who I am, being with me always and never let me to stop.

Thanks God for creating me with attitude of learning and understanding. I appreciate every single sign you send me to find the way for having better and happier life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TC	Text Classification
BOW	Bag of Words
TF	Term Frequency
DF	Document Frequency
DPM	Discriminative Power Measure
TFF	Term Frequency Factor
CFF	Collection Frequency Factor
RF	Relevance Factor
KNN	K Nearest Neighbor
SVM	Support Vector Machine
POS	Part Of Speech
MAX-TF	Maximum TF
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
INDScore	Individual Score

LIST OF SYMBOLS

χ^2	Chi-square

Chapter 1

INTRODUCTION

1.1 Automated Text Classification

Since early 90's, the number of available digital documents on the web is increasing exponentially as a result of enormous improvement in software and hardware technologies. Working with this massive amount of digital data, which may require preprocessing and organization, is time consuming and costly. These tasks including searching, gathering, ordering, classifying and arranging cannot be afforded by human efforts. In this case, the necessity of having automated solutions is obvious to facilitate these tasks.

Classification is one of the main tasks in effective management of the digital data that is also known as *Text Categorization* (TC). TC is the task of assigning one or more predefined categories (labels) to the natural language text documents automatically. In practice, designing an automated system to classify a given document is based on learning. In other words, a *Classifier* is trained using pre-labeled documents (training data) to predict the labels of unseen data (test data) (Sebastiani, 2002).

1.2 Typical Applications

Information on the web is generally in the form of textual documents. Currently, TC is employed in various contexts in different fields such as indexing the documents, filtering, generating metadata automatically, target marketing, to catalog the news articles (Lewis, 1992) and Web pages (Craven, et al., 1998) automatically, and other applications that need to select, adapt or organize the textual documents. More specifically, the use of TC can be listed as follow:

- Labeling news as politics, sports, business, and fashion.
- Labeling an email as spam, junked, social, work, others.
- Labeling research papers as journal or conference or by the type of journal or conference.
- Labeling books as science, novel, history, others.
- Labeling a text as news, personal document, medical document, biography, others.
- Labeling textual web pages regarding to the subject.

1.3 Implementation of a TC system

TC is a pattern recognition problem where the main goal is to recognize the patterns that can describe the relation and information in data. Depending on the method used to discover the patterns, pattern recognition can be categorized in two categories. In *supervised learning*, pre-labeled training data are employed to learn the relation between data and their labels. TC is a supervised learning task. On the other hand, in *unsupervised learning*, no labeled training data is used. In this case, meaningful patterns in data are determined. For instance, clustering is a typical unsupervised learning task.

Generally, a supervised learning task contains three main objects:

- *Training set*, which consists of a set of labeled samples
- *Model*, which is the output of the learning procedure for which the training dataset is employed.

- *Test* set, which is a set of unseen instances in the same format of the training dataset. Test and train datasets are disjoint.

The dataset in TC is a document corpus that might be collected from different domains such as the comments on social networks (Facebook, twitter, etc.), newspapers, handwriting, web pages, academic papers, etc. A text document is a set of alphanumeric characters and (or) images. For automated classification, it must firstly be transformed into a vector of discriminative attributes. The most frequently used technique is splitting the text into the words, known as *Bag Of Words* (BOW). In this technique, a given document is represented as a vector of term weights where the grammatical relations and orders of terms are ignored (Badawi & Altınçay, 2014).

The first step of BOW representation is the removal of redundant terms known as *stop-words* such as “a”, “is” and “the”, from BOW. Then, stemming is generally applied to the remaining terms to avoid the multiple use of the same word. Some of the terms that occurred at least once in at least one document in the training set may be irrelevant or useless for the classification task. In other words, some terms may not provide valuable information for classifying the documents. Such terms are generally discarded from the BOW list. This task is also known as *term selection*. Term selection techniques process the extracted features with the aim of selecting a subset of the highest discrimination capacity that can play a critical role in classifying the documents (Chen, Huang, Tian, & QU, 2009). Term selection schemes are based on *Term Frequencies* (TF) or *Document Frequencies* (DF). The term frequency of a term corresponds to the number of times it appears in the given document. Document frequency of a term presents the number of training documents that contain the term (Erenel & Altınçay, 2012). While high term frequency is a measure to show the importance of a term in the feature selection methods that are based on TF, presence or absence of a term in positive or negative documents plays a similar role for those that are DF based. In general, DF based schemes are more widely used. *Chi-square* ($\mathcal{X}^2$), $Gini\_index$, *Discriminative Power Measure* (DPM) are examples of such schemes (Man L. , Tan, Low, & Sung, 2005).

Each document may have a different length. The frequencies of terms in longer documents are more likely to be larger than the frequencies of the terms in the shorter ones. For better document representation, document lengths are normalized after a discriminative set of terms is selected.

After the preprocessing and term selection, each document is represented as a vector of words. In text classification domain, each of these words corresponds to a *feature*. More specifically, each document is represented as a *feature vector* where the entries correspond to the term weights of the selected terms in the given document. Term weights are generally defined as the product of *term frequency factor* (TFF) and *collection frequency factor* (CFF) of the corresponding term (Erenel & Altınçay, 2012). The term frequency factor depends on the number of times the term appears in the document whereas the collection frequency factor is a measure of the importance of the term for categorization.

Several weighting methods are studied. Some of them are unsupervised where the distribution of the term in different classes is not considered. On the other hand, supervised factors take this information into account (Man L. , Tan, Jian , & Yue , 2009). For instance, *Relevance Factor* (RF) that is also used in this study is a supervised scheme.

The last step of designing an automated TC system is training a classifier. Several approaches are considered so far such as *K Nearest Neighbors* (KNN), *Naïve Bayes* and *Support Vector Machine* (SVM). It is generally observed that SVM provides better scores compared to the others (Colas & Brazdil, 2006), (Erenel & Altınçay, 2012). Therefore, SVM is considered in this study.

The text categorization problem can be defined as *binary* where the main aim to decide whether the document under concern belongs to the target category or not. In binary TC, the documents belonging to the target category are named as p*ositive documents* while n*egative documents* are the remaining documents belonging to a different category. In this thesis, we studied binary TC.

1.4 Motivation

One of the main challenges in the text classification task is the problem of high dimensionality. This problem is a direct consequence of the richness of the natural languages in terms of different words. Having tens of thousands of terms is really common in a text classification domain. Since each term corresponds to a different dimension in document representation, it is not simple at all to model a particular category using all existing terms in the corpus. Moreover, the resulting system may be inaccurate and run slowly when used to categorize unseen documents.

On the other hand, having a small set of terms can also affect the system's performance in terms of accuracy. Even if all the selected terms are highly valuable, if the selected set is not rich enough, the resultant system may not provide a satisfactory level of accuracy. As a matter of fact, employing a good subset of terms is rather crucial for achieving a satisfactory performance.

In this thesis, the main focus is term selection. We firstly evaluated the existing term selection schemes to investigate their relative performances. Then, we proposed two new methods of term selection, which correspond to a modification in the way that the existing schemes are applied. The proposed approaches are iterative schemes that take into account the importance of the co-occurrences of different terms. $\mathcal{X}^2$, $Gini\_index$ and DPM are used for this purpose. When compared to the existing schemes, it is shown that co-occurrences in an important factor that must be considered during term selection.

1.5 Thesis Outlines

An overview of widely used preprocessing, document normalization, term selection and term weighting methods and a brief description of KNN, Naïve Bayes and SVM classifiers are provided in Chapter 2. In Chapter 3, the main purpose of this thesis is discussed and the proposed schemes are presented. Chapter 4 presents the experimental results and a comparative evaluation of different feature selection schemes. In Chapter 5, the conclusions drawn and the future work are presented.

Chapter 2

LITERATURE SURVEY

2.1 Document Representation for Text Categorization

Two major sub-problems in document representation for text categorization are term selection and term weighting to quantify the importance of the features. In the following subsection, these problems will be addressed.

2.1.1 Bag of Words Representation

In automatic text categorization, the electronic documents must be transformed into a vector form to be employed by the learning algorithm. In Bag of words (BOW) approach, the terms within the documents are considered as features and their frequencies are employed in setting the term weights.

One of the main parameters of representing a document in the BOW representation is selecting part of the document to be employed in classification. In practice, only a particular part of the documents such as title, abstract, conclusion, the combination of some parts or the full length of the text can be used for classification. In general, better scores are reported when the full-length documents are employed during classification (Hulth & Megyesi, 2006).

Some terms such as stop words may not convey any discriminative information. The next step is the elimination of these terms from BOW. In general, considering them in BOW will not contribute to the effectiveness of classifier despite the added dimensionality. In the case of English, it is reported that there are more than 400 stop words (Aggarwal & Zhai, 2012). Typically, stop words constitute 20% - 30% of the documents words (HaCohen-Kerner & Yishai Blitz, 2010). “the”, “and” and “or” are examples of stop words.

Several derivatives or inflected forms of words having the same root can exist in the written documents. For instance, the words “care”, “cared”, “careful” and “carefully” have the same root and should not be treated as different words. Similarly, the terms “car”, “cars” and “automobile” have the same meaning. In general, the process of computing the root of words is referred as *Word Normalization*. Stemming or lemmatization can be used for this purpose. Stemming is the process of returning the standard format of the terms, known as the *stem* by removing the prefixes or suffixes of the terms or using the look up tables. *Rule based stemmers* consider a set of rules to eliminate the affixes. For instance, if the token ends with “tion”, then it is removed. Alternatively, a lookup table, known as *dictionary,* that contains the stems of the terms may also be used. These are the *lookup stemmers.* The advantages of the latter approach are speed and simplicity. Despite the fact that the derivative or inflected form of the words should be the same as in the lookup table, exception handling is much easier when compared to the former approach. Rule-based algorithms are generally slower. These algorithms contain a set of rules based on the used languages. The rules about the morphology of the particular language are needed to be specified by the language experts. This is a time consuming and expensive process. The most common used stemmer is Porter stemmer (Porter, 1980), (Willett, 2006).

Lemmatization considers the *part of speech* (POS) of the terms. The process of extracting the *lemma* (dictionary form of the word) contains two major steps: first the POS of the word is determined. Then, different rules based on each POS are applied on the words. The performance of lemmatization heavily depends on the correct assignment of the POS to each term.

Lemmatization achieves better results in returning the lemma of the terms (Kettunen, Kunttu, & Järvelin, 2005). For example, the term "running" might be used in different contexts as "running is my favorite sport" or "I was running". It has a different POS in these different cases. Stemmers return "run" for both cases while lemmatization provides a different lemma for each case. Lemmatizers are also successful in computing the lemma of the terms like "saw" as "see" where stemmers miss the link between them. On the other hand, stemmer performs much better when the token is inflected. Stemming and Lemmatization are experimentally evaluated when used individually (Kettunen, Kunttu, & Järvelin, 2005) or in combination to get the richer methods (Ingason, Helgadóttir, Loftsson, & Rögnva, 2008).

2.1.2 Term Selection

In general, a subset of the terms is used for classification due to two major reasons. Firstly, some terms may convey negligible information about the label of the document. Secondly, using too many terms may lead to the curse of dimensionality. Therefore, term selection is generally applied on the extracted feature set before classifier design. In this task, the importance of the individual terms is firstly quantified and then a subset is selected.

Table 2.1 describes the distribution of the term t_i in positive or negative documents.

Table 2.1: The definition of the information elements for term t_i and category c (Sebastiani, 2002).

	$\boldsymbol{t_i}$	$\boldsymbol{\bar{t}_i}$
$\boldsymbol{c}$	A	B
$\boldsymbol{\bar{c}}$	C	D

In particular, A represents the number of positive documents and C shows the numbers of negative documents that contain t_i. Similarly, B represents the number of positive documents and D denotes the number of negative documents in which t_i didn't occur. The total number of documents in corpus can be represented by N where, $N = A + B + C + D$. The standard feature selection schemes are employing these information elements to compute the importance of different terms (Erenel, Altınçay, & Varoglu, 2011).

Alternatively, A, B, C and D can be modified to take into account the frequencies of terms as well (Azam & Yao, 2012). More specifically, A_{TF} is defined as the sum of the normalized term frequency of t_i in the positive documents and $B_{TF} = N - A_{TF}$. Similarly, C_{TF} is defined as the sum of the normalized term frequencies of t_i in the negative documents and $\mathrm{D}_{TF} = N - C_{TF}$. The definition of the information elements for term selection schemes based on document frequency and term frequency are presented in Table 2.2.

Table 2.2: The definition of the information elements for document and term frequency based term selection schemes.

	Document Frequency		***Term Frequency***
A	*The number of positive documents that contains t_i*	*A_{TF}*	*The sum of the normalized term frequency of t_i in the positive documents*
B	*$N^+ - A$*	*B_{TF}*	*$N^+ - A_{TF}$*
C	*The number of negative documents that contains t_i*	*C_{TF}*	*The sum of the normalized term frequency of t_i in the negative documents*
D	*$N^- - C$*	*D_{TF}*	*$N^- - C_{TF}$*

There are various schemes used for term selection. In this study, we considered three methods, namely Chi-square (χ^2), $Gini\_index$ and discriminative power measure (DPM).

2.1.2.1 Chi-square (χ^2)

χ^2 is one of the most widely used symmetric term weighting schemes. It is based on measuring the dependency between t_i and the target class, *c* (Yang & Pedersen, 1997). A lower value of $\chi^2(t_i, c)$ represents a lower dependency between t_i and *c*. Since we are interested in terms with high dependency, those t_i with the highest $\chi^2(t_i, c)$ value will be selected. (Ogura, Amano, & Kondo, 2009)

The χ^2 value of a term can be computed using the two-way contingency table (Table 2.1) (Man L. , Tan, Jian , & Yue , 2009). $\chi^2(t_i, c)$, which denotes the χ^2 value of t_i when the class *c* is considered, calculate using Eq. 2.1 (Erenel , Altınçay, & Varoglu, 2011).

$$\chi^2(t_i, c) = \frac{N(AD-BC)^2}{Max\{1,(A+C)*(B+D)*(A+B)*(C+D)\}} \qquad (2.1)$$

In most of the empirical studies conducted for text classification such as (Forman, 2003), (Deng, Tang, Yang, Li, & Xie, 2004), (Debole & Sebastiani, 2004) and (Man L. , Tan, Low, & Sung, 2005), χ^2 is reported to perform better than many of its competitors.

2.1.2.2 Gini_index

$Gini\_index$ is another symmetric terms selection method that is based on the purity of features (Dong, Shang, & Zhu, 2011). This metric also used in decision trees to split the attributes. It is also extensively used for text feature selection (Shang, Huanga, Zhu, Lin, Qu, & Wang, 2007). The experiments results show that $Gini\_index$ provides comparable and, in some cases, better performance than other term selection schemes (Ogura, Amano, & Kondo, 2009). The $Gini\_index$ value of t_i when class c is considered can be obtained using Eq. 2.2 (Ogura, Amano, & Kondo, 2009).

$$Gini\_index_{(t_i,c)} = \frac{1}{Max\{1,(A+C)^2\}}((\frac{A^2}{(A+B)})^2 + (\frac{C^2}{(C+D)})^2) \qquad (2.2)$$

$Gini\_index_{(t_i,c)}$ represents the goodness of t_i with the respect to *c*. The $Gini\_index$ of better terms are bigger.

2.1.2.3 Discriminative Power Measure (DPM)

Discriminative power measure is the third method that is used in this study to evaluate the importance of the terms (Chen, Leeb, & Changc, 2009). The DPM value of t_i is calculated using Eq. 2.3 (Azam & Yao, 2012).

$$DPM_{(\,t_i\,,c)} = \sum_{i=1}^{A+C}(\frac{A}{A+C} - \frac{B}{B+D}) \qquad (2.3)$$

2.1.3 Term Weighting

After a subset of terms is selected using one of the aforementioned schemes, the document vectors can be constructed. This will be achieved by computing the feature values of different terms that is known as term weights. Term weights reflect importance of the terms with the respect to the target category. The relative importance of different terms may differ and this is represented by the differences in their magnitudes. Term weights are generally made up of the product of two factors, namely term frequency factor and collection frequency factor (Altınçay, 2013).

2.1.3.1 Term Frequency Factor

The term frequency factor defines based on the number of times the term occurs in the concerned document (Erenel & Altınçay, 2012). It may be the raw term frequency value or its transformed form using the logarithm function. Several other transformations are studied and it is generally observed that logarithm function provides superior performance (Erenel & Altınçay, 2012), (Man L., Tan, Jian, & Yue , 2009). In this study, the raw value of term frequency is used in the simulations. It should be noted that the length of a document can heavily affect the term frequency of the terms. The frequency of a term is expected to be larger in longer documents. Therefore, the frequencies of a term in documents having different lengths are not comparable. Document length normalization aims to eliminate the effect of differences (Erenel & Altınçay, 2012) in document lengths on term frequency. All the documents in the corpus should be normalized so as to have equal lengths. The normalized term frequency is a real number in [0, 1]. *Cosine normalization* and *Maximum TF Normalization* (MAX-TF) are the most popular methods in text categorization (Singhal, Buckley, & Mitra, 1996).

Cosine normalization is the most frequently used in text categorization (Salton, Wong, & Yang, 1975). Assuming that *N* terms are selected to be employed for TC, the normalized term frequency value of $t_i$after applying cosine normalization will be computed using Eq. 2.4 (Singhal, Buckley, & Mitra, 1996).

$$CosineTF_{t_i} = TF_{t_i} / \sqrt{\sum_{n=1}^{N} TF_{t_n}^2} \qquad (2.4)$$

Maximum TF (Max-TF) is another scheme for document length normalization (Azam & Yao, 2012). In this technique, the terms' TF values are normalized using the maximum frequency value in the same document. Assuming that $maximumTF$ denotes the maximum TF value in the document under concern, the normalized TFs can be computed using Eq. 2.5 (Singhal, Buckley, & Mitra, 1996).

$$MAX - TF_{t_i} = TF_{t_i} / Max\{maximumTF, 1\} \qquad (2.5)$$

Cosine normalization is more commonly used in text categorization (Chowdhury, Mccabe, Grossman, & Frieder, 2002).

2.1.3.2 Collection Frequency Factor

Collection frequency factor quantifies the importance of the terms. More specifically, this factor represents the discriminative ability of the term when the whole training corpus is considered. The distribution of the terms in positive and negative documents is considered for this purpose. In particular, terms that mainly occur in either positive or negative terms are expected convey discriminative information about the target category *Relevance frequency* (RF) is a supervised CFF that is experimentally shown to surpass many other methods (Man L. , Tan, Jian , & Yue , 2009), (Erenel, Altınçay, & Varoglu, 2011). Using the information elements presented in Table 2.1, the RF weight of t_i can be calculated employing Eq. 2.6 (Man L. , Tan, Low, & Sung, 2005).

$$RF_{(t_i, c)} = \log_2(2 + \frac{A}{\max\{1, C\}}) \qquad (2.6)$$

In the computation of RF, the main idea is that the terms having higher frequency in the positive documents are more discriminative than the terms that mainly appear in the negative documents (Man L. , Tan, Low, & Sung, 2005).

In summary, the product of the normalized term frequency (TF) and the collection frequency factor (RF) represents the term's weight ($TF \times RF$).

After the term weights are computed, the document vectors are constructed. The next step is the design of the classification scheme.

2.2 Classification Techniques

By studying the available training documents, classification techniques are employed to construct a general model to be used for predicting the category of the unseen documents. Various kinds of classifiers are developed based on different assumptions and methodologies. Choosing a classification scheme is a critical decision in TC task since the feature vectors are high-dimensional and sparse. In this section, three widely used classification techniques employed in document categorization are discussed briefly:

2.2.1 K Nearest Neighbors

One of the simplest classification techniques is supervised K Nearest Neighbors (KNN) (Fix, Hodges, & Joseph, 1951). It is an instance based machine-learning algorithm that remembers all the instances in training data without constructing any decision model. In fact, all the computation and prediction process is done when the system tests an unseen instance to predict its label. Because of this, this algorithm is referred as a lazy learning algorithm (Wettschereck, Aha, & Mohri, 1997). During the classification process, the first step in KNN is finding the K closest (similar or have shorter distance) samples in training data to the tested sample. Then, the label of the tested instance is set to be the label of the majority these instances in training data.

K is an integer number. The optimal value of K can be computed using cross-validation. When K=1, the unseen sample will be classified as the class of the nearest neighbor in training data.

The similarity between different training samples can be measured by using a distance measure such as Euclidean distance, Manhattan distance and Cosine distance. It is generally observed that the best-fitting distance metric is domain dependent.

The size of the dataset directly affects the speed of the KNN classification system. When large numbers of training instances having high-dimensional feature vectors exist, the computation times become large. In summary, the value of K and the distance metric are the design parameters of KNN.

2.2.2 Naïve Bayes

$Naïve\ Bayes$ is a simple but powerful probabilistic classification technique based on Bayes' theorem (Murphy, 2006). Unlike KNN, Naïve Bayes studies the training samples and comes up with the decision function. This function takes into account the dependency of each term with the different classes and the prior probability of each class. $Naïve\ Bayes$ assumes that the terms presences or absences of different terms in the documents of a given class are independent events. The prior probability of the class c ($P(c)$) is generally calculated as the proportion of the training samples that belong to c. In $Naïve\ Bayes$, the decision about the label of the given document is computed by taking into account the a posteriori probability of all classes, which is defined as Eq. 2.7 (Murphy, 2006).

$$P(c|t_1, t_2, \dots, t_T) = \frac{P(c) \times P(t_1, t_2, \dots, t_T|c)}{P(t_1, t_2, \dots, t_T)} \qquad (2.7)$$

The class receiving the maximum posteriori probability is selected as the most likely. Due to the independency assumption, $Naïve\,Bayes$ systems need to learn the conditional density function of each term separately. In addition to its simplicity, the execution time of $Naïve\ Bayes$ is much less when compared to KNN.

2.2.3 Support Vector Machines

Support vector machine (SVM) is proven to be one of the most robust and accurate classifiers for text classification (Joachims, 1998). They are basically designed for binary classification. However, by transforming the multiclass problem to binary, SVM can be used for any classification problem (Burges, 1998).

One of the main design parameters of SVM is the kernel type. With the use of a linear kernel, a linear classifier can be designed. However, nonlinear classifiers can be obtained by using other kernel types such as polynomial or radial basis function. Experiments on binary text categorization have shown that, employing the linear kernel generally provides better performance scores compared to nonlinear kernels (Man L. , Tan, Low, & Sung, 2005), (Zhan & Loh, 2009).

The linear kernel separates the negative (represented by -1) and positive (represented by +1) classes by designing a linear hyperplane defined as it is shown in Eq. 2.8.

$$\mathcal{F}(x) = \ \omega \, . \, \phi(x) - b \tag{2.8}$$

In above equation, $\mathcal{F}(x) = 0$ defines the decision boundary. If $\mathcal{F}(x) > 0$, classifier will classify the samples as positive class (1) where the samples will be classified as negative (-1) when $\mathcal{F}(x) < 0$.

In order to compute the optimal hyperplane, SVM considers two hyperplane as $\mathcal{F}(x) = 1$ and $\mathcal{F}(x) = -1$. We would like to compute w and b which satisfies

$\mathcal{F}(x) > 1$ for positive samples and $\mathcal{F}(x) < -1$ for the negative samples as illustrated in Figure 2.1. This means that we want the samples to be away from the decision boundary for better generalization. In other words, the hyperplanes must have maximum distance from each other. This can be formulated as maximizing the margin that can be computed as $\frac{2}{\|\omega\|}$ (Burges, 1998).

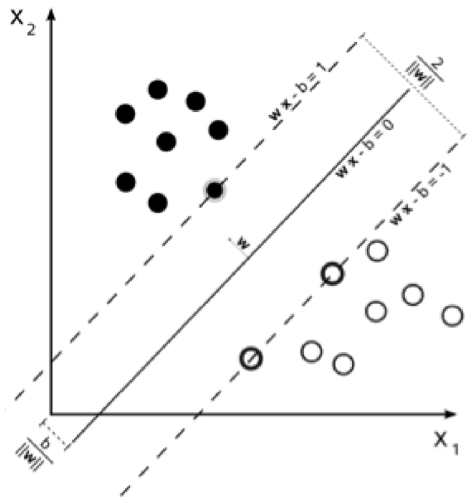


Figure 2.1: Decision and support hyperplanes in SVM with linear kernel (Thakur, 2009).

Extensive researches have been conducted for comparing these three classifiers (Abe, Tsumoto, Ohsaki, & Yamaguchi, 2009), (Colas & Brazdil, 2006). The results show that SVM classifier with a linear kernel is a reasonable choice for text classification since it can successfully deal with large number of features whereas $Na\ddot{\imath}ve\ Bayes$ and KNN perform poorly in such cases.

2.3 Performance Evaluation

In order to evaluate the performance a particular scheme in binary text categorization, *precision, recall* and $\mathcal{F}_1 score$ of each category are calculated individually. These scores are computed using the information elements defined in Table 2.2. True positive (TP) presents the numbers of positive documents that are correctly classified whereas true negative (TN) is the number of negative documents that are correctly classified. False positive (FP) presents the number of negative documents that are incorrectly classified as positive and false negative (FN) denotes the number of positive documents that are misclassified as the negative category (Sebastiani, 2002).

Table 2.3: The definitions of TP, FP, FN and TN for category c_i (Liu, Wu, & Zhou, 2009).

Category c_i		***Classifier Judgments***	
		YES	***NO***
Expert Judgments	***YES***	TP_i	FN_i
	NO	FP_i	TN_i

Precision is defined as the percentage of positive documents in proportion to all documents that are classified as positive as given in Eq. 2.9. Recall declares the percentage of correctly classified positive documents as given in Eq. 2.10. In binary TC, the harmonic mean of precision and recall that is also known as $\mathcal{F}_1 score$ is also computed for each category. The $\mathcal{F}_1 score$ of the i^{th} category will be calculated using Eq. 2.11 (Liu, Wu, & Zhou, 2009).

$$Precision_i = 100 \times \frac{TP_i}{TP_i + FP_i} \tag{2.9}$$

$$Recall_i = 100 \times \frac{TP_i}{TP_i + FN_i} \tag{2.10}$$

$$\mathcal{F}_1 score_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{2.11}$$

As an overall performance measure for problems including multiple categories, both macro and micro $\mathcal{F}_1 score$ are generally employed in the case of imbalanced datasets. To calculate the macro $\mathcal{F}_1 score$, the overall average of precision and recall for different categories are considered. Moreover, the macro $\mathcal{F}_1 score$ is calculated using the modified precision and recall. Eq. 2.12 and 2.13 present the modified precision and recall formula where to calculate macro the Eq. 2.14 is used (Sebastiani, 2002).

$$macro\ precision = \frac{\sum_{i=1}^{C} Precision_i}{n} \tag{2.12}$$

$$macro\ precision = \frac{\sum_{i=1}^{C} Recall_i}{n} \tag{2.13}$$

$$macro\ \mathcal{F}_1 = \frac{1}{n} \sum_{i=1}^{C} \mathcal{F}_1 score_i \tag{2.14}$$

i represents the index of the category and C is total number of categories. In order to calculate micro $\mathcal{F}_1 score$ the informative elements in Table 2.2 are modified as presented in Table 2.3. Eq. 2.15 and 2.16 present the modified precision and recall and while Eq. 2.17 presents the formula for calculating micro $\mathcal{F}_1 score$ (Sebastiani, 2002).

Table 2.4: The definitions of TP, FP, FN and TN for all categories in dataset (Sebastiani, 2002).

Category set $\boldsymbol{C = \{c_1, c_2, .., c_n\}}$		***Classifier Judgments***	
		YES	***NO***
Expert Judgments	***YES***	$TP = \sum_{i=1}^{C} TP_i$	$FN = \sum_{i=1}^{C} FN_i$
	NO	$FP = \sum_{i=1}^{C} FP_i$	$TN = \sum_{i=1}^{C} TN_i$

$$micro\ precision = 100 \times \frac{TP}{TP + FP} \tag{2.15}$$

$$micro\ recall = 100 \times \frac{TP}{TP + FN} \tag{2.16}$$

$$micro\ \mathcal{F}_1 = \frac{2 \times micro\ precision \times micro\ recall}{micro\ precision + micro\ recall} \tag{2.17}$$

2.3.1 Datasets

There are several datasets that are widely used for binary text classification, such as Reuters-21578, Ohsumed and 20 Newsgroups (Badawi & Altınçay, 2014). In this study, we used the ModApte split of top ten classes of Reuters-21578 since it has a highly imbalanced category distribution that makes it one of the most important datasets. For each of ten categories, the positive class is defined as the set of documents belonging to the target category where the negative class includes all the remaining documents (Erenel & Altınçay, 2012). In this dataset, the training and test sets are fixed to have 6491 and 2545 documents, respectively.

Chapter 3

PROPOSED TERM SELECTION FRAMEWORK

In the BOW representation, each feature corresponds to a different word appearing in the training corpus. The term selection schemes measure the importance of each word *individually* according to the target category using the information elements presented in Table 2.1. The effectiveness of various term selection schemes are studied and comparative evaluations are reported (Chen, Huang, Tian, & QU, 2009), (Debole & Sebastiani, 2004), (Liu, Loh, & Sun, 2009) and (Yang, Liu, Zhu, Liu, & Zhang, 2012). As an extension to the BOW-based representation, with the use of syntactic phrases that take into account the grammatical relations and statistical phrases that are made up of consecutively occurring pairs (bigrams) or triples (trigrams) of words, improved representations can be achieved.

Termsets allow an alternative document representation in which the co-occurrences of terms (two or more) is considered. However, different from statistical phrases, there is no need that the terms consecutively occur in the text. In other words, the member terms can be from any part of document. Recently, the use of termsets for document representation attracted the interest of many researchers in text classification domain (Tesar, Strnad, Jezek, & Poesio, 2006), (Figueiredo, Rocha, Couto, Salles, Gonçalves, & Meira, 2011), (Badawi & Altınçay, 2014).

Since all terms are considered in the construction of termsets, a huge number of termsets will be computed. As a matter of fact, selection of a good subset is very important. As a classical approach, the information elements of the termsets can be used for selection. Alternatively, the co-occurrence statistics of the member terms can be considered (Badawi & Altınçay, 2014). In this thesis, we followed the second path. The motivation can be explained as follows. Consider the termset "rock singer". As it can be seen, the occurrence of both terms supports the "music" category. The occurrence of the second term but not the first one supports the same category. On the other hand, the occurrence of the first term but not the second will not support the music topic strongly and may support another category. Obviously, the assigned weight to this occurrence with the respect to the music category is not high. Hence, considering the co-occurrences of "rock" and "singer" and evaluating them as a termset leads to a more informative feature when compared to their individual evaluation. Moreover, occurrence of one of the terms but not the other may still be informative.

The main idea of the proposed term selection scheme is inspired from termset based representation. In particular, in selecting the terms, an *iterative scheme* is designed which takes into account the co-occurring statistics of the candidate terms and the previously selected ones. Consider a pair of terms, namely t_i and t_j. The presence one term but not the other introduces two possible cases. Let $\{\bar{t}_i, t_j\}$ represents the presence of t_j but t_i and $\{t_i, \bar{t}_j\}$ represents the presence of t_i. Assume that the first case is denoted by "01" and the second case by "10". Let N^+ and N^- denote the numbers of positive and negative documents, respectively. Consider the information elements presented in Table 3.1 which represents the number of documents where the aforementioned events occur.

Table 3.1: The modified information elements used in term selection schemes based on two different co-occurrences of terms as $\{\bar{t}_i, t_j\}$ and $\{t_i, \bar{t}_j\}$.

Information elements for $\{\bar{t}_i, t_j\}$		***Information elements for*** $\{t_i, \bar{t}_j\}$	
AA_{01}	The number of positive documents that contains t_j but not t_i	A_{10}	The number of positive documents that contains t_i but not t_j
B_{01}	$N^+ - A_{01}$	B_{10}	$N^+ - A_{10}$
C_{01}	The number of negative documents that contains the t_j but not t_i	C_{10}	The number of negative documents that contains t_i but not t_j
D_{01}	$N^- - C_{01}$	D_{10}	$N^- - C_{10}$

The term selection schemes are modified so as to employ the elements given in Table 3.1. All the proposed schemes are document frequency based. In particular, the selection schemes denoted by $\mathcal{X}_{01}^2$, $Gini\_index_{01}$ and DPM_{01} for $\{\bar{t}_i, t_j\}$ are formulated after replacing the information elements as follows:

$$\mathcal{X}_{01}^2(t_i, t_j) = \frac{N(A_{01}D_{01} - B_{01}C_{01})^2}{Max\{1,(A_{01}+C_{01})*(B_{01}+D_{01})*(A_{01}+B_{01})*(C_{01}+D_{01})\}} \quad (3.1)$$

$$Gini\_index_{01}(t_i, t_j) = \frac{1}{Max\{1,(A_{01}+C_{01})^2\}}((\frac{{A_{01}}^2}{(A_{01}+B_{01})})^2 + (\frac{{C_{01}}^2}{(C_{01}+D_{01})})^2) \quad (3.2)$$

$$DPM_{01}(t_i, t_j) = \sum_{i=1}^{A_{01}+C_{01}} \sum_{j=1}^{A_{01}+C_{01}} (\frac{A_{01}}{A_{01}+C_{01}} - \frac{B_{01}}{B_{01}+D_{01}}) \quad (3.3)$$

Similarly, to compute the weights for $\{t_i, \bar{t}_j\}$, the selection schemes are modified by replacing the original elements with *A*A_{10}, B_{10}, C_{10} *and* D_{10} and are denoted by $\mathcal{X}_{10}^2$, $Gini\_index_{10}$ and DPM_{10} as follows:

$$\mathcal{X}_{10}^2(t_i, t_j) = \frac{N(A_{10}D_{10} - B_{10}C_{10})^2}{Max\{1,(A_{10}+C_{10})*(B_{10}+D_{10})*(A_{10}+B_{10})*(C_{10}+D_{10})\}} \quad (3.4)$$

$$Gini\_index_{10}(t_i, t_j) = \frac{1}{Max\{1,(A_{10}+C_{10})^2\}}((\frac{{A_{10}}^2}{(A_{10}+B_{10})})^2 + (\frac{{C_{10}}^2}{(C_{10}+D_{10})})^2) \quad (3.5)$$

$$DPM_{10}(t_i, t_j) = \sum_{i=1}^{A_{10}+C_{10}} \sum_{j=1}^{A_{10}+C_{10}} \left(\frac{A_{10}}{A_{10}+C_{10}} - \frac{B_{10}}{B_{10}+D_{10}}\right) \quad (3.6)$$

The terms are firstly sorted according to their *individual scores* (INDScore) where each term is evaluated by the standard selection schemes (document frequency based schemes) ($\mathcal{X}^{2}$, $Gini\_index$ and DPM) using Eq. 2.1, 2.2 and 2.3. Then, for each pair of terms ($\{t_i, t_j\}$), the 01 score (ω_{01}) and 10 score (ω_{10}) using the modified selection schemes ($\mathcal{X}_{01}^2$, $\mathcal{X}_{10}^2$, $Gini\_index_{01}$, $Gini\_index_{10}$, DPM_{01} and DPM_{10}) are computed with respect to the target category.

Two different index lists are used to record the output of each iteration. The *selected-index* list (Figure 3.1 part a) is a grow-up list that contains the indices of the selected terms based on the proposed schemes. The *remaining-index* list (Figure 3.1 part b) contains the indices candidate terms. In this list, terms are placed in descending order with the respect to their INDScore.

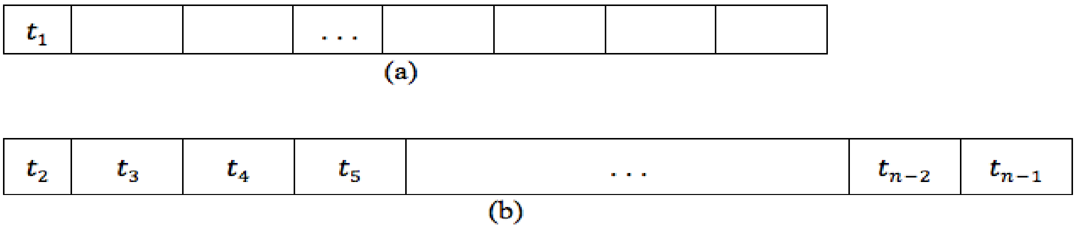


Figure 3.1: Selected-index list (a) and remaining-index list (b).

At the beginning, since the first term's index at the top of the remaining-index list has a highest INDScore, place at the top of selected-index list. The selected terms are removed from the remaining-index list and added to the end of the selected-index list. Then the discriminative powers of the terms in remaining-index list are iteratively evaluated by considering the 01 and 10 weights using the selected-index terms in addition to their INDScores. This combination can be formulated as follows:

$$score(t_j) = INDScore_j + TERMSETscore_j \quad (3.7)$$

TERMSETscore$_j$ denotes the score computed using pairwise evaluation of the candidate term t_j and the selected-index terms and $INDScore_j$ represents the individual score of t_j. This means that both the discriminative powers of the candidate terms and the discriminative capacity provided when considered in pairs with the previously selected terms are considered.

It should be noted that, ω_{01} and ω_{10} weights must be computed by using the ranking scheme that was used for obtaining the INDscore. For instance, if the terms are sorted using χ^2, then the $\mathcal{X}_{01}^2$ and/or $\mathcal{X}_{10}^2$ must be used for computing *TERMSETscore$_j$*. Three different techniques are proposed in this thesis for the computation of *TERMSETscore$_j$*.

- $01_{score}(j)$: Defined as the average ω_{01} values computed using t_j and all selected-index terms. Let *m* denote the cardinality of the selected-index list. Then, selected-index is defined as follows:

  $$01_{score}(j) = \frac{1}{m}\left(\sum_{i=1}^{m} \omega_{01}(\{t_i, t_j\})\right),\ j = 1 \dots n \tag{3.8}$$

  ω_{01} denotes the weight of the event $\{\bar{t}_i, t_j\}$ using one of the modified schemes' equations, Eq. 3.1, 3.2 or 3.3 ($\mathcal{X}_{01}^2$, $Gini\_index_{01}$ or DPM_{01}). *n* presents the number of the terms in the remaining-index list.

- $Max(01,10)_{score}(j)$: This score is computed as the average of the maximum value of ω_{01} and ω_{10} for each particular t_j in remaining-index list and all the t_i in the selected-index list as follows:

  $$Max(01,10)_{score}(j) = \frac{1}{m}\left(\sum_{i=1}^{m} Max(\omega_{01}, \omega_{10})\right),\ j = 1 \dots n \tag{3.9}$$

  ω_{10} denotes the weight of the event $\{t_i, \bar{t}_j\}$ and the number of the terms in the remaining-index list is represented by *n*.

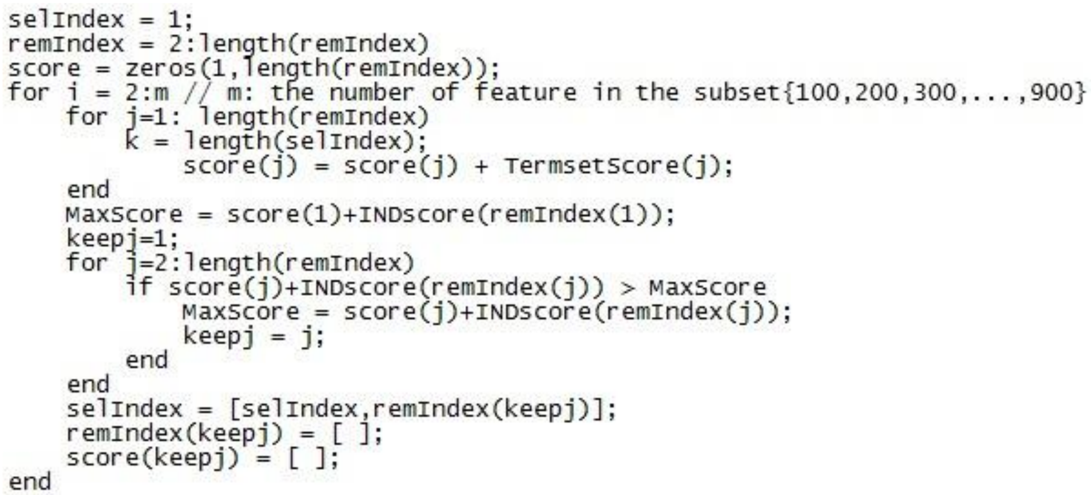
- $Mean(01{,}10)_{score}(j)$: In this technique, first the mean values of the ω_{01} and ω_{10} for the specific t_j and all t_i in selected-index list are calculated. *n* denotes the number of the terms in the remaining-index list. Then the average of this value is used. This calculation is presented in Eq. 3.10.

$$Mean(01{,}10)_{score}(j) = \frac{1}{m}\left(\sum_{i=1}^{m} Mean(\omega_{01}, \omega_{10}))\right), j = 1 \ldots n \qquad (3.10)$$

In this thesis, using these metrics for calculating the score of each candidate term, two different selection schemes are proposed.

3.1 Document Representation Using Individual Terms

In this scheme, after computing $score\ (t_j)$ for all the terms in the remaining-index list, the term with the highest $score\ (t_j)$ value is selected to be added to the end of the selected-index list and it is removed from the remaining-index list. Then, the documents are represented as vectors of the terms in the order of the selected-index list. In this representation, although the terms are selected with the respect to their INDScore and co-occurrence scores simultaneously, as in the BOW representation, each feature corresponds to a different term. The resultant document vectors are normalized and term weighting is applied in the conventional way as described in the previous chapter. The pseudo code presents how proposed selection scheme works (Figure 3.2).



```
selIndex = 1;
remIndex = 2:length(remIndex)
score = zeros(1,length(remIndex));
for i = 2:m // m: the number of feature in the subset{100,200,300,...,900}
    for j=1: length(remIndex)
        k = length(selIndex);
            score(j) = score(j) + TermsetScore(j);
    end
    MaxScore = score(1)+INDscore(remIndex(1));
    keepj=1;
    for j=2:length(remIndex)
        if score(j)+INDscore(remIndex(j)) > MaxScore
            MaxScore = score(j)+INDscore(remIndex(j));
            keepj = j;
        end
    end
    selIndex = [selIndex,remIndex(keepj)];
    remIndex(keepj) = [ ];
    score(keepj) = [ ];
end
```

Figure 3.2: The pseudo code that presents how proposed selection scheme select the individual terms.

3.2 Document Representation Using Individual Terms and Term Pairs

This scheme is applied in two phases. In the first phase, the individual term selection scheme is applied where $01_{score}(j)$ (Eq. 3.8) is used for evaluating the effectiveness of co-occurrences of different terms. In the second phase, novel features are defined as the pairs of different terms. For term pair selection, $pair_{score}(t_i, t_j)$ is computed to quantify the discriminative ability of different pairs. Two different measures are employed for this purpose:

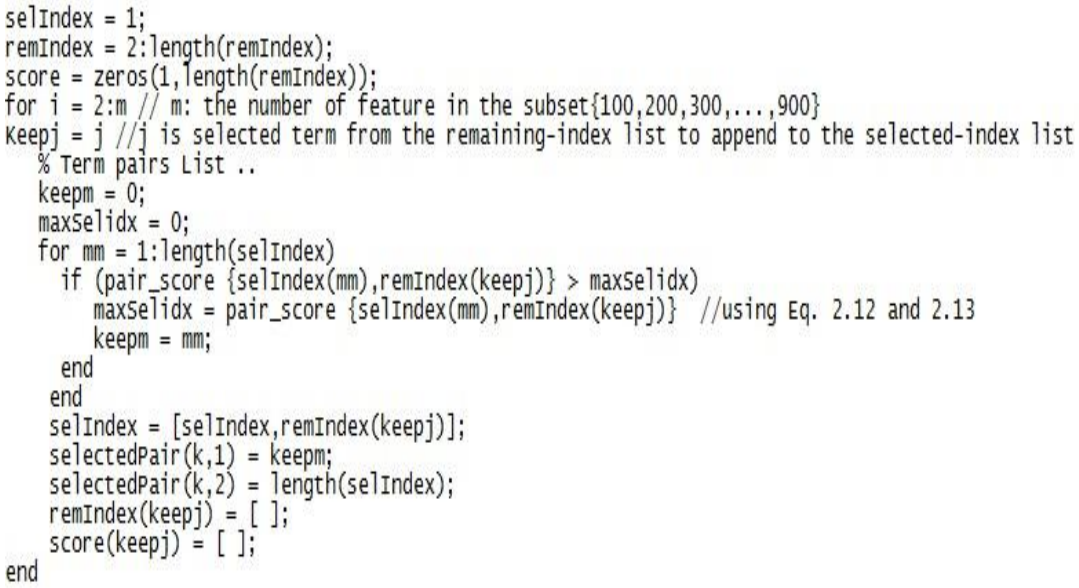
1. The score is defined using the 01 score as:

$$pair_{score}(t_i, t_j) = \omega_{01}(t_i, t_j) \qquad (3.11)$$

2. The score is defined using the sum of the individual scores and the 01 score as:

$$pair_{score}(t_i, t_j) = \omega_{01}(t_i, t_j) + INDScore(t_i) + INDScore(t_j) \qquad (3.12)$$

In order to determine the term pairs, the selected-index list terms determined by the first phase are employed. Assume that t_j denotes the top-ranked term in the remaining-index list that should be appended to the selected-index. The pair$_{\text{score}}$ value is computed for all t_i in the selected-index list. The pair having the highest pair$_{\text{score}}$ value will be. Then, considering the next ranked term, the procedure described above is repeated. The term $\omega_{01}(t_i, t_j)$ is computed using the same term selection scheme employed during ranking the individual terms. The pseudo code below presents the steps of the proposed selection scheme (Figure 3.3).



```
selIndex = 1;
remIndex = 2:length(remIndex);
score = zeros(1,length(remIndex));
for i = 2:m // m: the number of feature in the subset{100,200,300,...,900}
Keepj = j //j is selected term from the remaining-index list to append to the selected-index list
   % Term pairs List ..
   keepm = 0;
   maxSelidx = 0;
   for mm = 1:length(selIndex)
     if (pair_score {selIndex(mm),remIndex(keepj)} > maxSelidx)
        maxSelidx = pair_score {selIndex(mm),remIndex(keepj)}  //using Eq. 2.12 and 2.13
        keepm = mm;
     end
    end
    selIndex = [selIndex,remIndex(keepj)];
    selectedPair(k,1) = keepm;
    selectedPair(k,2) = length(selIndex);
    remIndex(keepj) = [ ];
    score(keepj) = [ ];
end
```

Figure 3.3: The pseudo code that presents how proposed selection scheme select the individual terms and term pairs.

After the discriminative term pairs are identified, document vectors are defined using the union of terms and term pairs are features (Figure 3.2). The numbers of individual terms and term pairs to be utilized are fixed a priori in terms of percentages of the desired number of features. Top ranked term and term pairs are used for this purpose. In computing the term frequency factors of term weights, the cosine normalized frequencies of the terms are used. In the case of term pairs, the sums of normalized term frequency factors of t_i and t_j are used.

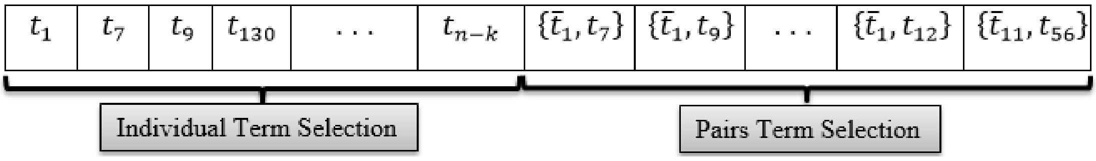


Figure 3.4: Documents are represented in terms of individual terms and term pairs.

The collection frequency factors of individual terms are computed using RF. For computing the collection frequency factors of term pairs, RF is modified based on the informative elements in Table 3.1. In particular, the modified RF is defined as:

$$\widehat{RF}\left(\{t_i, \mathrm{t_j}\}\right) = \log_2(2 + \frac{A_{01}}{\max\{1, C_{01}\}}) \qquad (3.13)$$

After the overall weights are computed as the product of term frequency and collection frequency factors (Erenel & Altınçay, 2012), the document vectors are employed for classifier generation and its evaluation.

Chapter 4

EXPERIMENTAL RESULTS

4.1 Experimental Setup

In the simulations, Reuters-21578 is employed in this study. This dataset contains 10 categories, namely acquisition, corn, crude, earn, grain, interest, money-fx, ship, trade and wheat. The numbers of documents in each category are different. In some categories such as earn, there are thousands of training documents whereas there are less than two hundred in some others such as corn. Therefore, Reuters is known as an imbalance dataset. For each category, the training set contains 6491 documents and the test data contains 2545 documents that were represented using 17008 features in BOW representation. The experimental study is focused on representing the documents using top 900 terms using either various selection schemes. The experiments were done on different subsets of the features in {100, 200, 300, 400, 500, 600, 700, 800, 900}. The details of experimental setups are represented in Table 4.1.

Table 4.1: Experimental setup

Dataset	*Reuters- 21578*
Training data	*6491 documents*
Test data	*2545 documents*
Number of terms (in dataset)	*17008 terms*
Number of terms in selected subsets	*{100, 200, 300, 400, 500, 600, 700, 800, 900}*
Document length normalization method	*Cosine normalization*
Term selection schemes	*Chi-square (χ^2) , $Gini\_index$, DPM*
Collection frequency factor	*Relevance Frequency(RF)*
Classifier	*SVM^{Light} with the linear Kernel and default settings*

After removing the stop words and applying the Porter stemmer algorithm (Porter, 1980), cosine normalization is applied to normalize the length of the documents. We used SVM in our simulations since it is observed to achieve better performance in high dimensional problems (Man L., Tan, Jian, & Yue, 2009) including text categorization. SVM^{light} toolbox with linear kernel and the default cost-factor value ($C = 1/avg(\bar{x}^T \bar{x})$) which is the inverse of the average of inner product of the training data's values is used for this purpose (Joachims, 1998). As the performance metric, micro and macro $\mathcal{F}_1 score$ are considered.

4.2 Simulations

In the first set of experiments, the use of document frequency factor in term selection is evaluated. The terms are ranked using χ^2, $Gini\_index$ and DPM schemes. In the second phase, the use of term frequency in term selection is studied. χ^2_{TF},

$Gini\_index_{TF}$ and DPM_{TF} are utilized for this purpose. Figures 4.1 to 4.6 present the performance of the term frequency and document frequency based selection schemes.

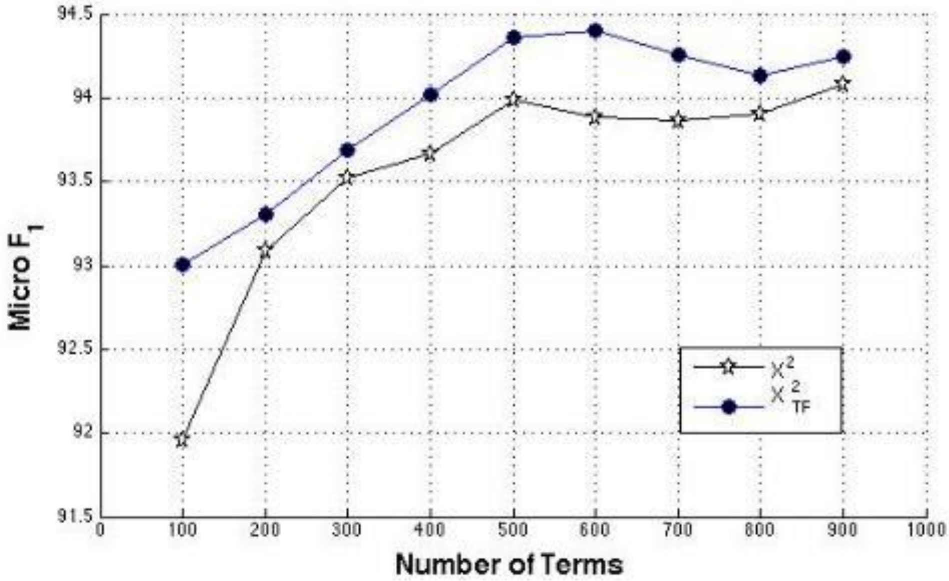


Figure 4.1: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2 and χ^2_{TF} using RF as the CFF and SVM^{light} as the classifier.

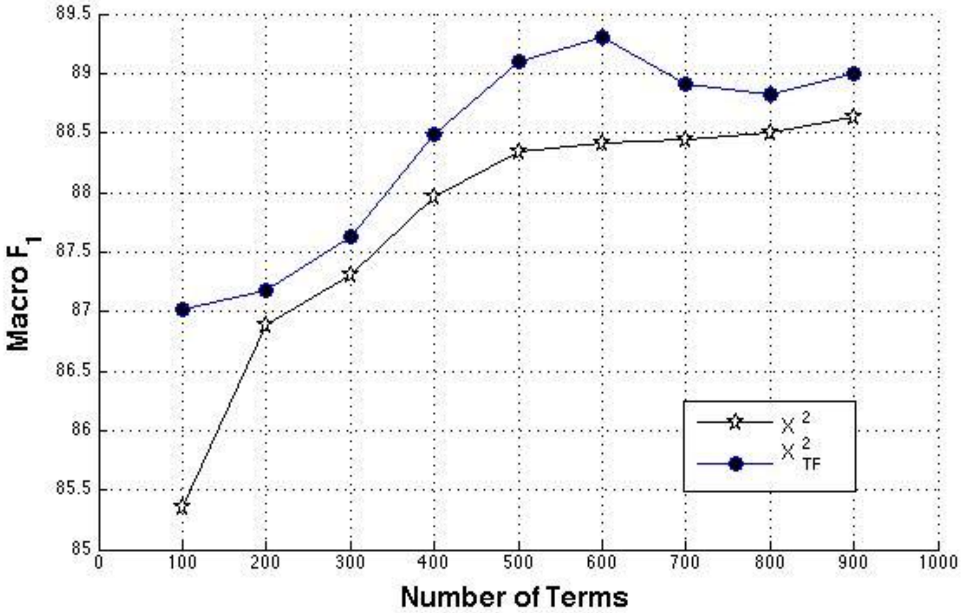


Figure 4.2: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2 and χ^2_{TF} using RF as the CFF and SVM^{light} as the classifier.

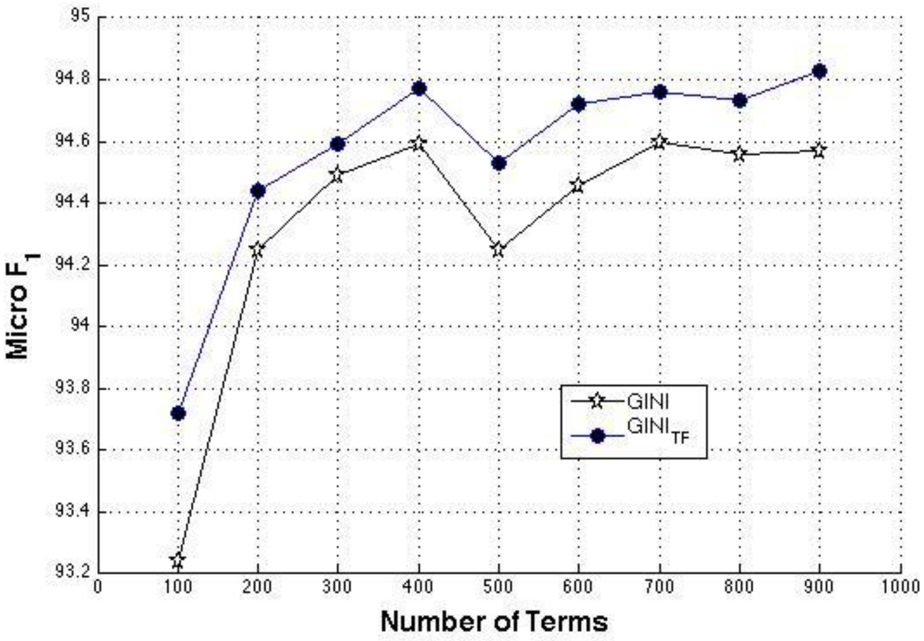


Figure 4.3: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$ and $Gini\_index_{TF}$ using RF as the CFF and SVM^{light} as the classifier.

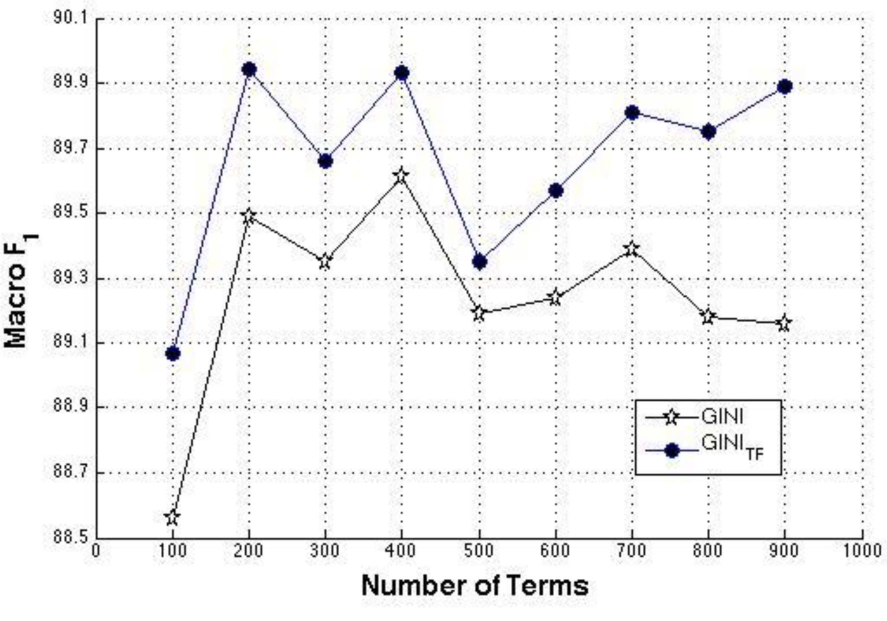


Figure 4.4: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$ and $Gini\_index_{TF}$ using RF as the CFF and SVM^{light} as the classifier.

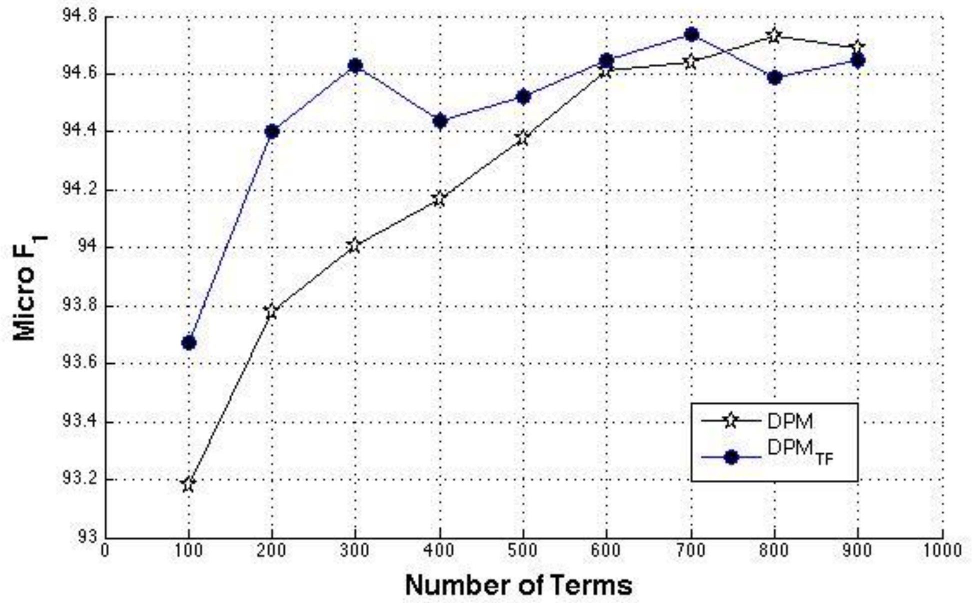


Figure 4.5: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM and DPM_{TF} using RF as the CFF and SVM^{light} as the classifier.

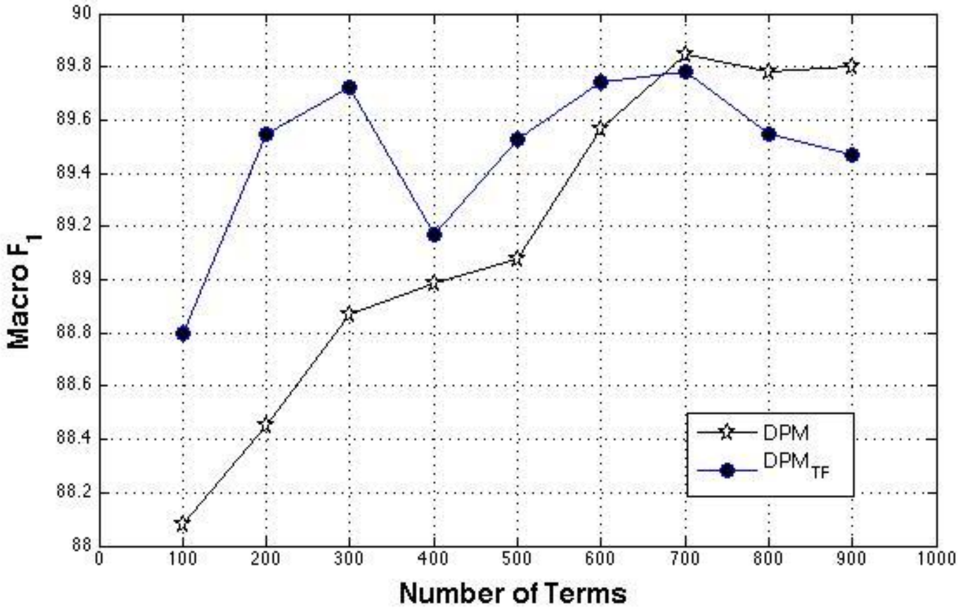


Figure 4.6: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM and DPM_{TF} using RF as the CFF and SVM^{light} as the classifier.

For a comparative evaluation, the micro and macro $\mathcal{F}_1 scores$ achieved using χ^2_{TF}, $Gini\_index_{TF}$ and DPM_{TF} is presented in Figure 4.7 and 4.8.

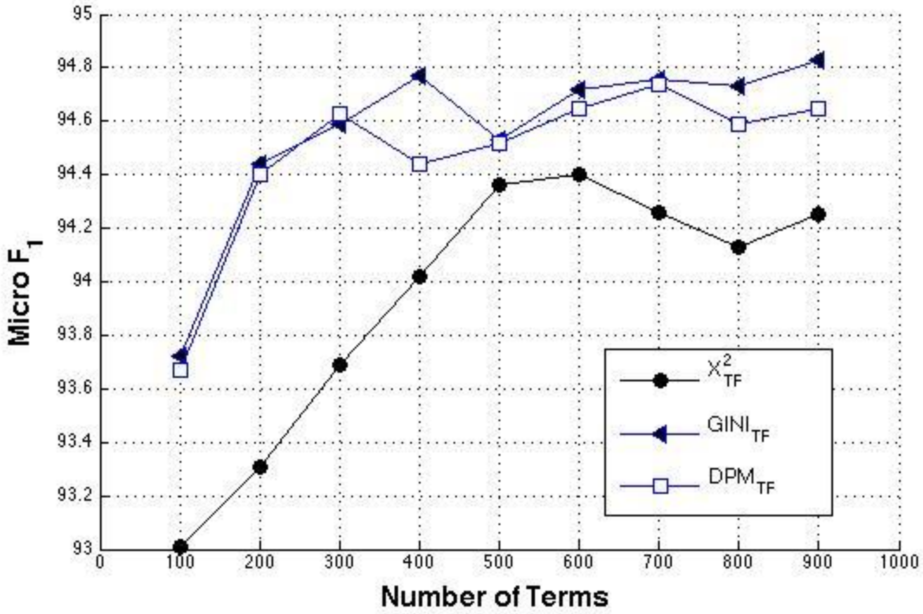


Figure 4.7: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2_{TF}, $Gini\_index_{TF}$ and DPM_{TF} using RF as the CFF and SVM^{light} as the classifier.

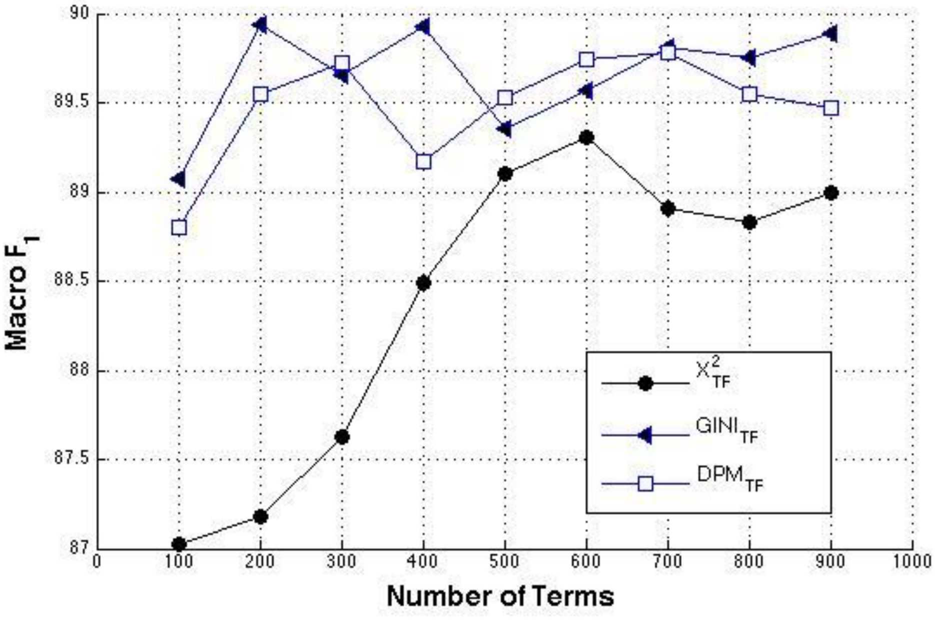


Figure 4.8: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2_{TF}, $Gini\_index_{TF}$ and DPM_{TF} using RF as the CFF and SVM^{light} as the classifier.

It can be seen in the figures that, for all three selection schemes, the use of term frequencies provides improved performance compared to the use of document frequencies. The performances of $Gini\_index_{TF}$ and DPM_{TF} are comparable.

Figures 4.9 to 4.14 present the micro and macro $\mathcal{F}_1 scores$ achieved using the reference and the proposed individual term selection scheme. Considering the large number of the terms in the Reuters dataset, the number of term pairs that must be evaluated is $(17008 \times 17007)/2$. Therefore, the pairwise evaluation is restricted to top 1000 terms. To increase the speed of simulations, the values of the $\mathcal{X}_{01}^2$, $\mathcal{X}_{10}^2$, $Gini\_index_{01}$, $Gini\_index_{10}$, DPM_{01} and DPM_{10} for the termsets were computed before doing experiments.

As it was mentioned in Chapter 3, in document representation using individual terms, three different techniques were considered to compute $TERMSETscore_j$. For example, in the following figures, χ_{01}^2 is the plot of the modified system that $01_{score}(t_j)$ is employed for computing the value of $TERMSETscore_j$. Similarly, $Max\ (\chi_{01}^2, \chi_{10}^2)$ presents the systems that used $Max\ (01{,}10)_{score}(t_j)$ for the same purpose while $Mean\ (01{,}10)_{score}(t_j)$ technique is used in $Mean\ (\chi_{01}^2, \chi_{10}^2)$ system. The same naming methodology is employed for $Gini\_index$ and DPM as well.

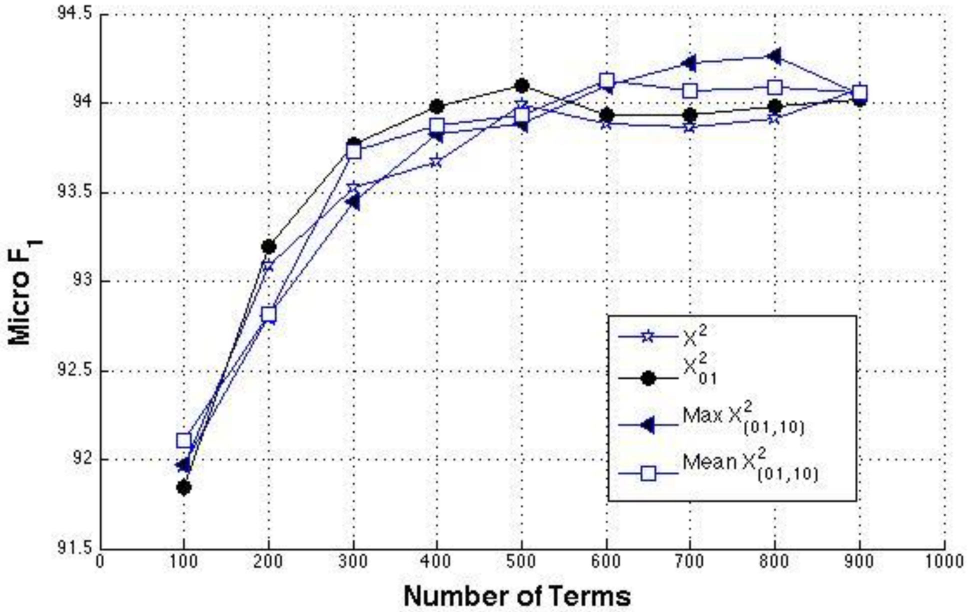


Figure 4.9: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $\chi^2$and the proposed individual term selection schemes using χ^2_{01}, Max $(\chi^2_{01}, \chi^2_{10})$ and $Mean$ $(\chi^2_{01}, \chi^2_{10})$.

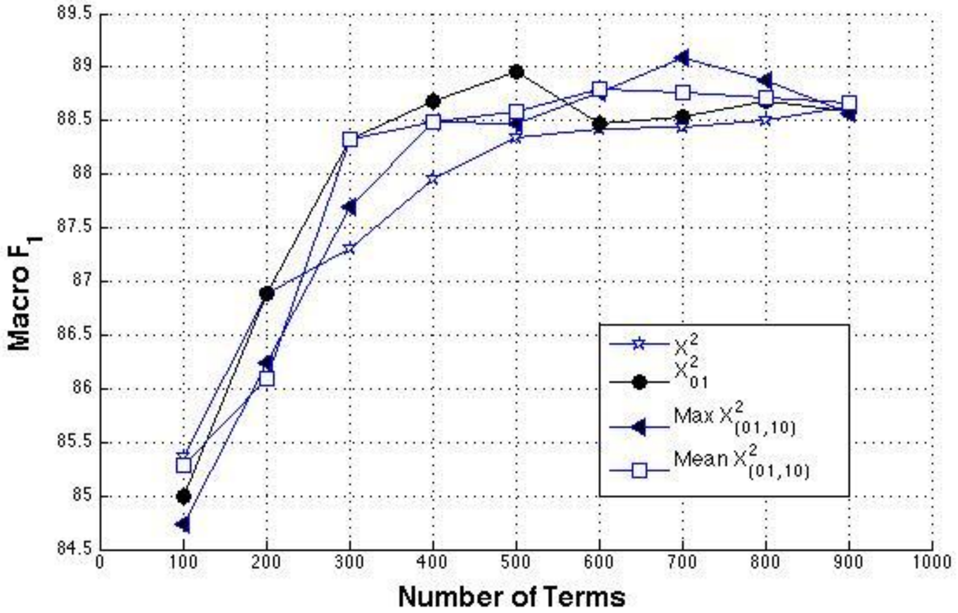


Figure 4.10: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $\chi^2$and the proposed individual term selection schemes using χ^2_{01}, Max $(\chi^2_{01}, \chi^2_{10})$ and $Mean$ $(\chi^2_{01}, \chi^2_{10})$.

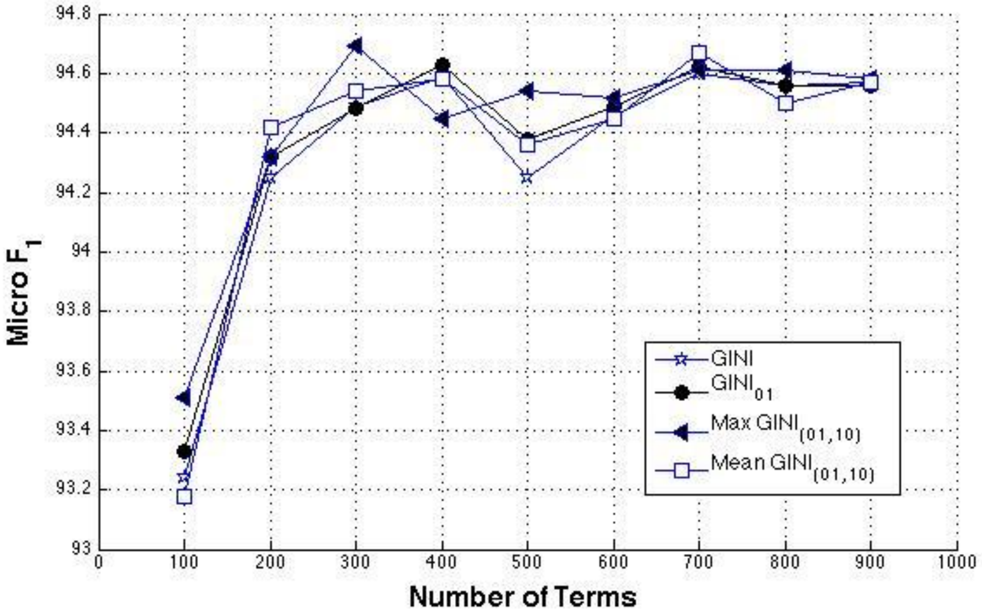


Figure 4.11: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$, and the proposed individual term selection schemes using $Gini\_index_{01}$, $Max\ (Gini\_index_{01}, Gini\_index_{10})$ and $Mean\ (Gini\_index_{01}, Gini\_index_{10})$.

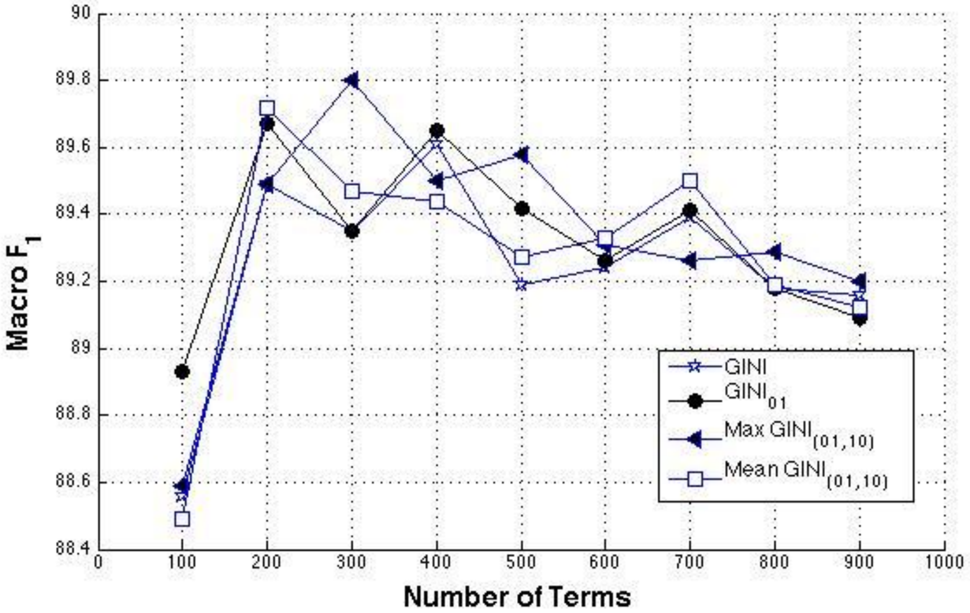


Figure 4.12: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$, and the proposed individual term selection schemes using $Gini\_index_{01}$, $Max\ (Gini\_index_{01}, Gini\_index_{10})$ and $Mean\ (Gini\_index_{01}, Gini\_index_{10})$.

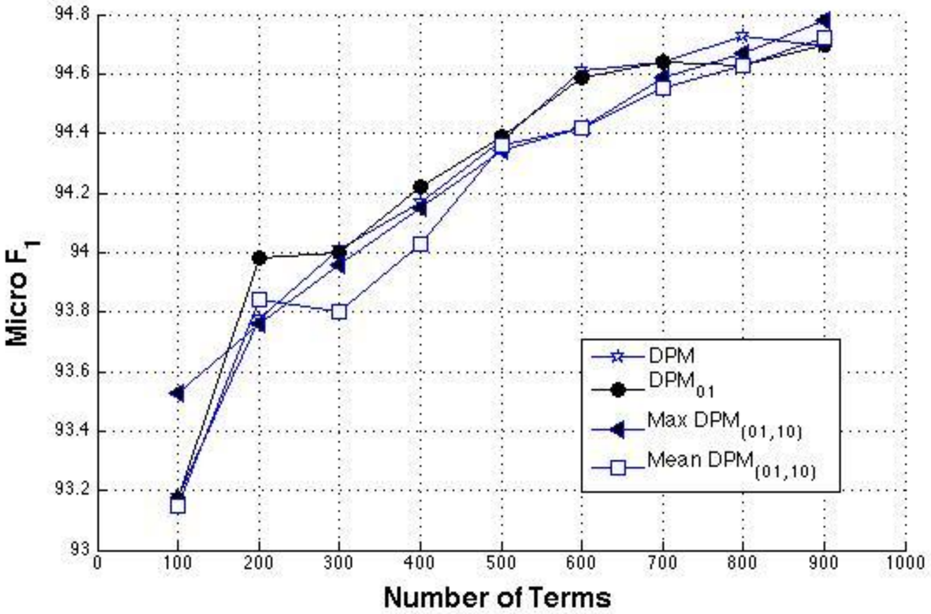


Figure 4.13: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM, and the proposed individual term selection schemes using DPM_{01}, $Max\ (DPM_{01}, DPM_{10})$ and $Mean\ (DPM_{01}, DPM_{10})$.

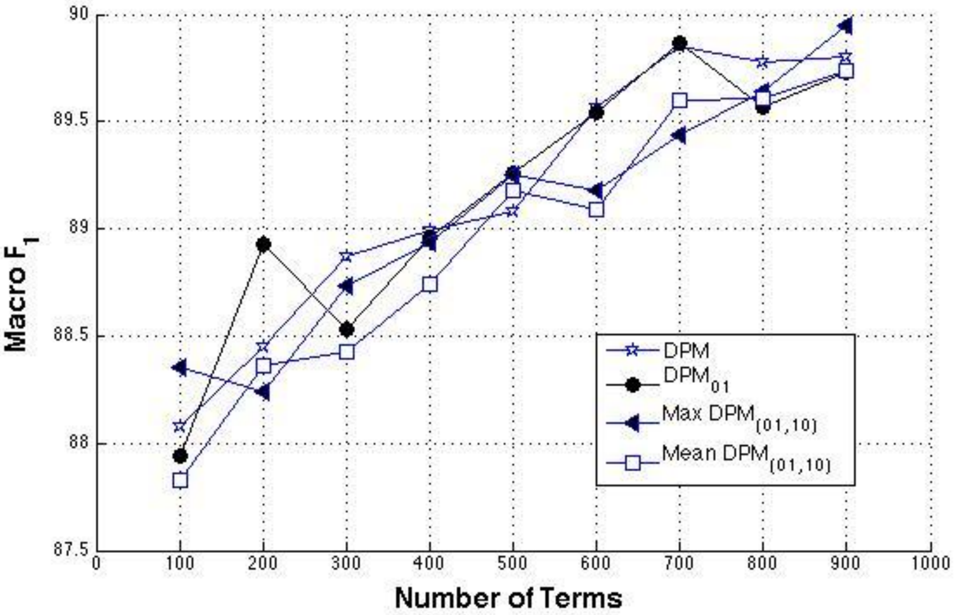


Figure 4.14: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM, and the proposed individual term selection schemes using DPM_{01}, $Max\ (DPM_{01}, DPM_{10})$ and $Mean\ (DPM_{01}, DPM_{10})$.

It can be seen in the figures that better scores are achieved for χ^2. However, the scores are comparable for the other selection schemes. In general, using $01_{score}(t_j)$ provides better scores. For a better visualization, Figures 4.15 to 4.20 present the micro and macro $\mathcal{F}_1 score$ of the reference and proposed individual term selection method where $01_{score}(t_j)$ is used for selection.

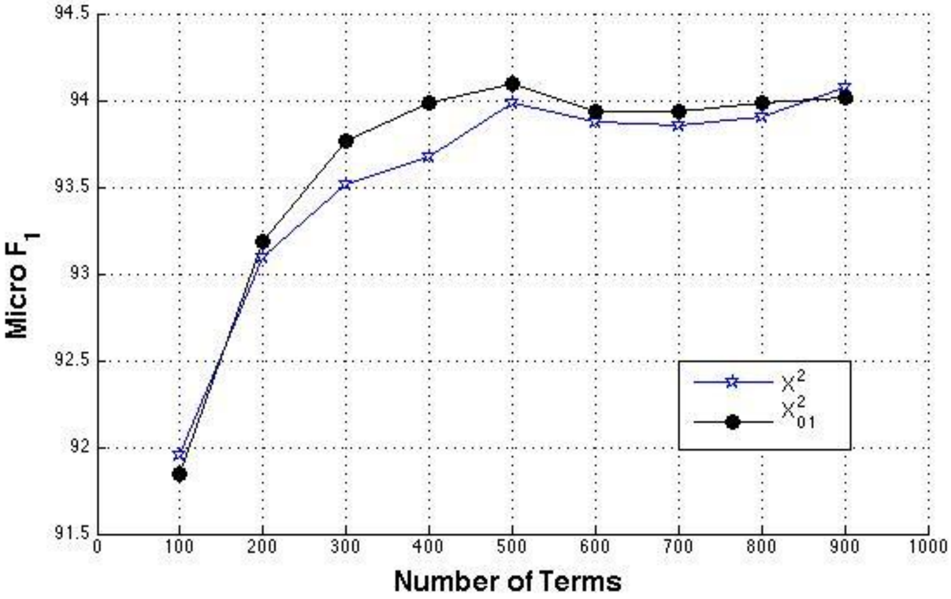


Figure 4.15: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2 and the proposed individual term selection schemes using $01_{score}(t_i)$.

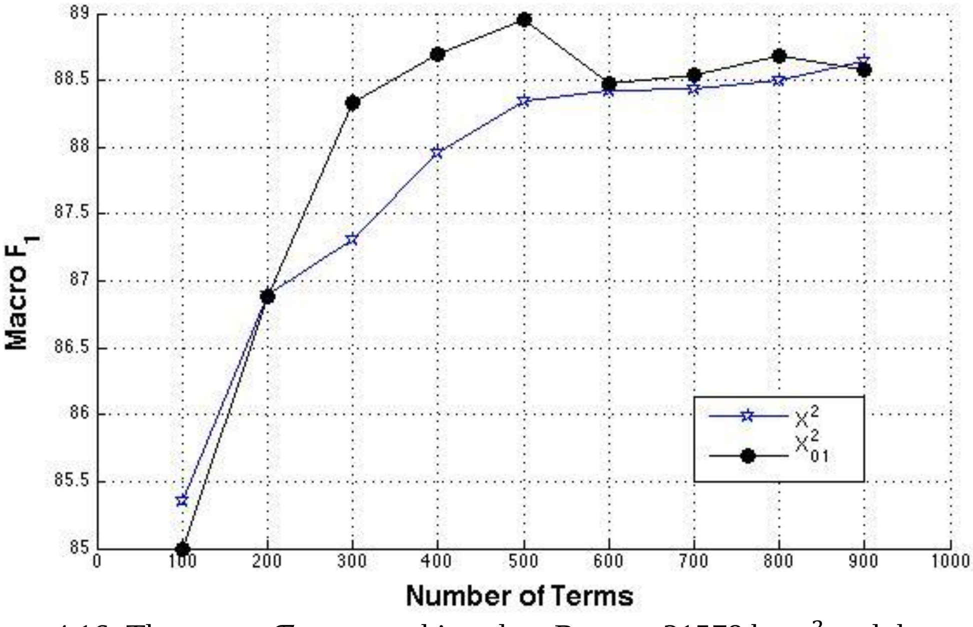


Figure 4.16: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2 and the proposed individual term selection schemes using $01_{score}(t_i)$.

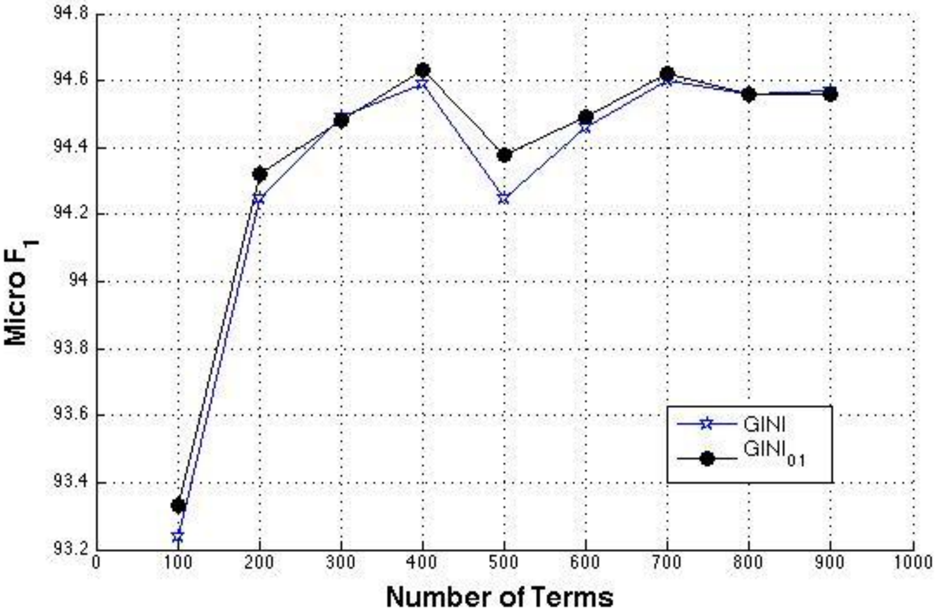


Figure 4.17: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$, and the proposed individual term selection schemes using $01_{score}(t_i)$.

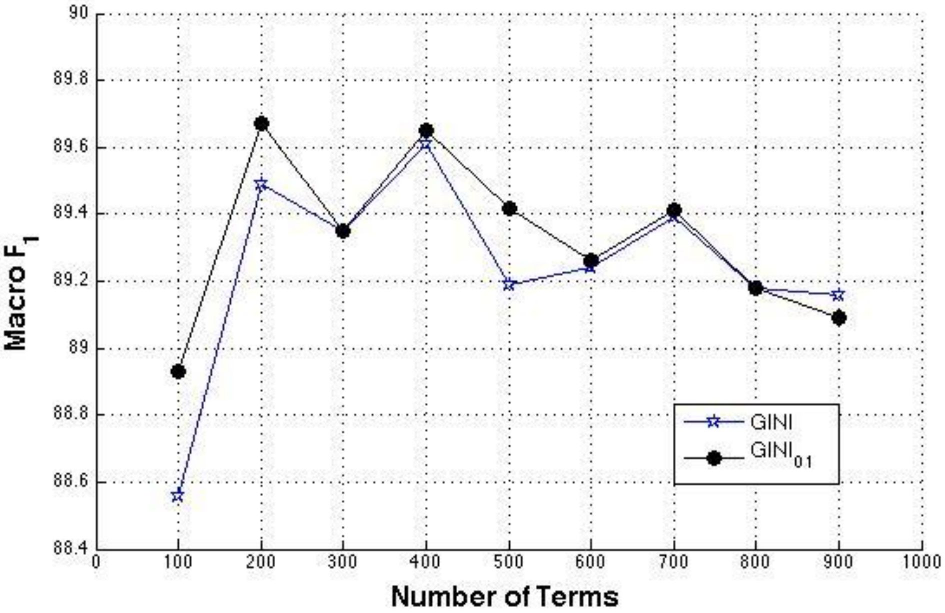


Figure 4.18: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$, and the proposed individual term selection schemes using $01_{score}(t_i)$.

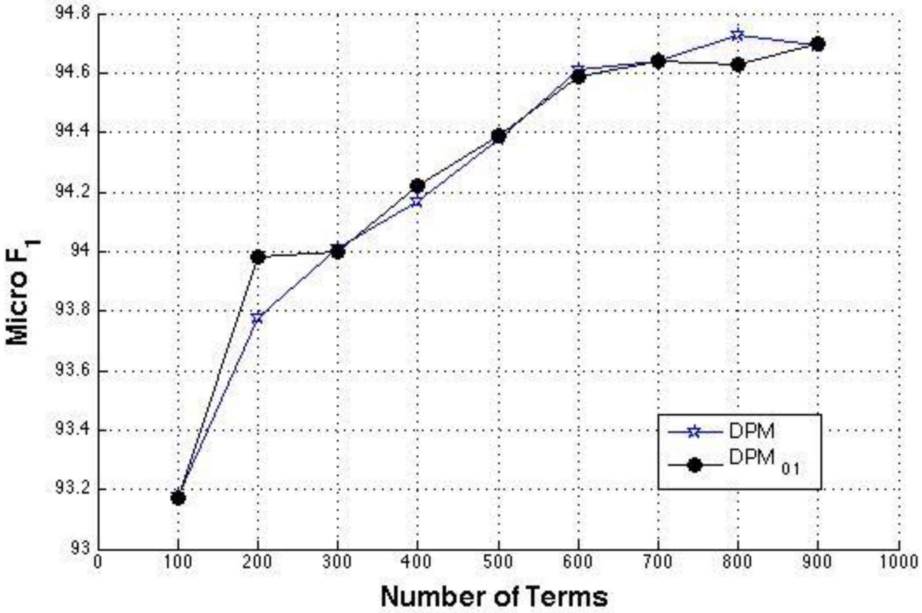


Figure 4.19: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM, and the proposed individual term selection schemes using $01_{score}(t_i)$.

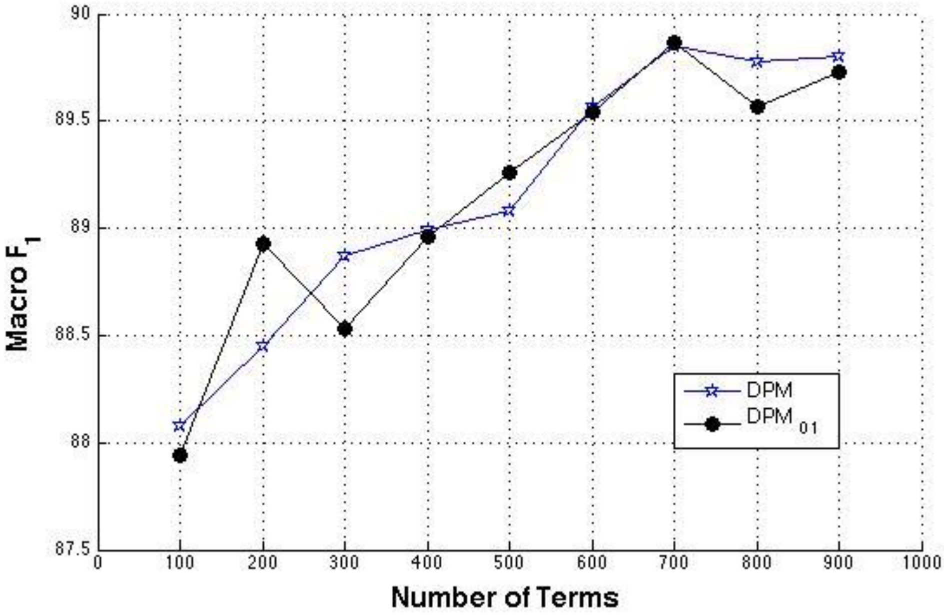


Figure 4.20: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM, and the proposed individual term selection schemes using $01_{score}(t_i)$.

Last sets of experiments are about the performance of the system including term pairs as features. $01_{score}(t_i)$ is used for computing the $TERMSETscore_j$. For term pair selection, the $pair_{score}(t_i, t_j)$ with/without the INDScore of the individual terms are employed.

After sorting all 1000 terms using $score\ (t_i)$ in the selected-index list and 999 term pairs according to their $pair_{score}(t_i, t_j)$ in the pair-list, documents were represented in combination of individual and pair-based features. 90% of features correspond to individual terms and 10% corresponds to term pairs.

Figures 4.21 to 4.26 present the micro and macro $\mathcal{F}_1 score$ of the proposed schemes and reference system. The figures with the “empty star” symbol present the reference systems. The filled bullet present the system using the proposed individual term selection scheme where is used $01_{score}(t_i)$ to compute the $TERMSETscore_j$. The other two plots present the systems where both individual and pair selection schemes are considered. 90% of the selected terms were selected with $01_{score}(t_i)$ technique to compute the $TERMSETscore_j$.

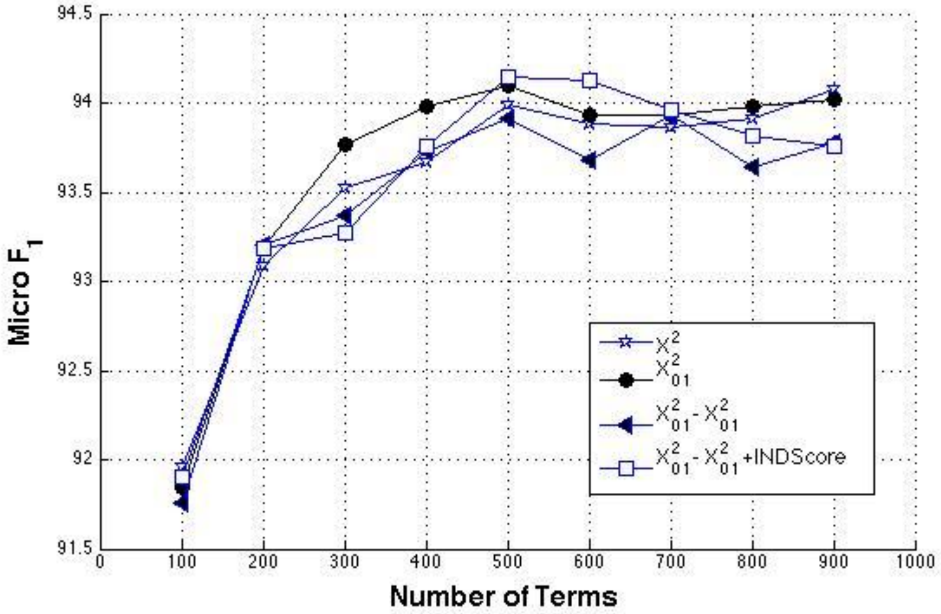


Figure 4.21: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2, χ^2_{01} and the proposed individual and pair selection schemes. $01_{score}(t_i)$ is used for $score\ (t_i)$ and $01_{score}(t_i)$ with and without INDScore for $pair_{score}(t_i, t_j)$.

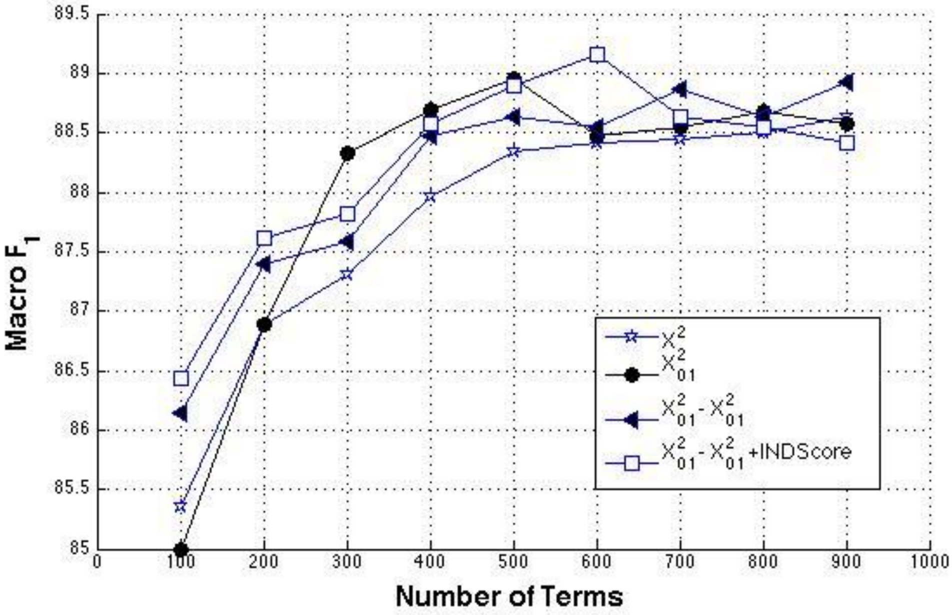


Figure 4.22: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by χ^2, χ^2_{01} and the proposed individual and pair selection schemes. $01_{score}(t_i)$ is used for $score\ (t_i)$ and $01_{score}(t_i)$ with and without INDScore for $pair_{score}(t_i, t_j)$.

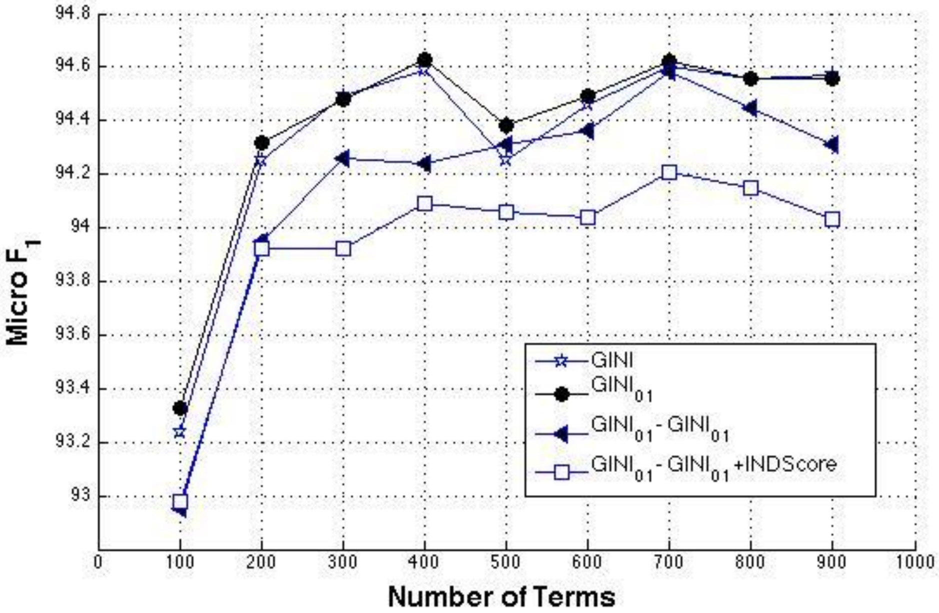


Figure 4.23: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$, $Gini\_index_{01}$ and the proposed individual and pair selection schemes. $01_{score}(t_i)$ is used for $score\ (t_i)$ and $01_{score}(t_i)$ with and without INDScore for $pair_{score}(t_i, t_j)$.

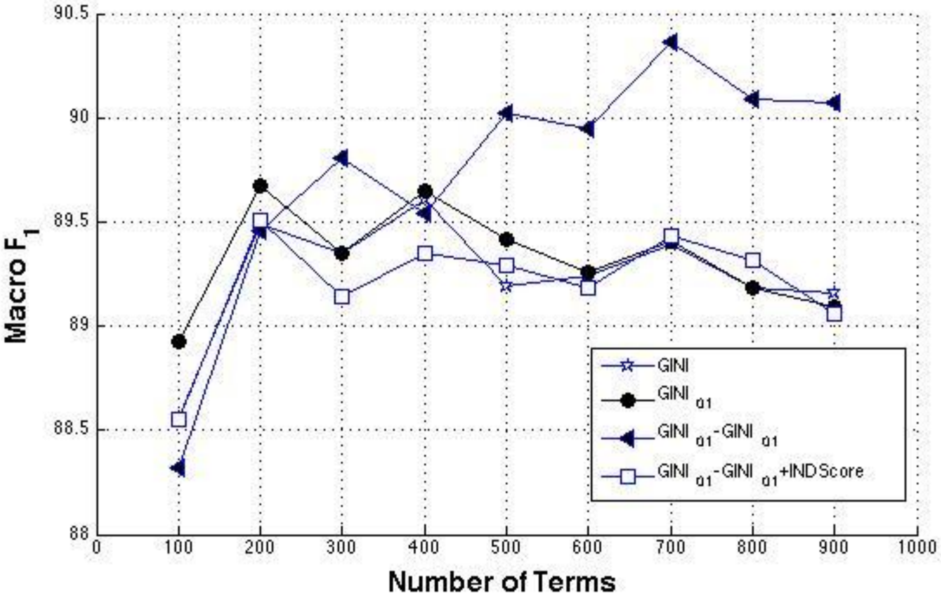


Figure 4.24: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by $Gini\_index$, $Gini\_index_{01}$ and the proposed individual and pair selection schemes. $01_{score}(t_i)$ is used for $score\ (t_i)$ and $01_{score}(t_i)$ with and without INDScore for $pair_{score}(t_i, t_j)$.

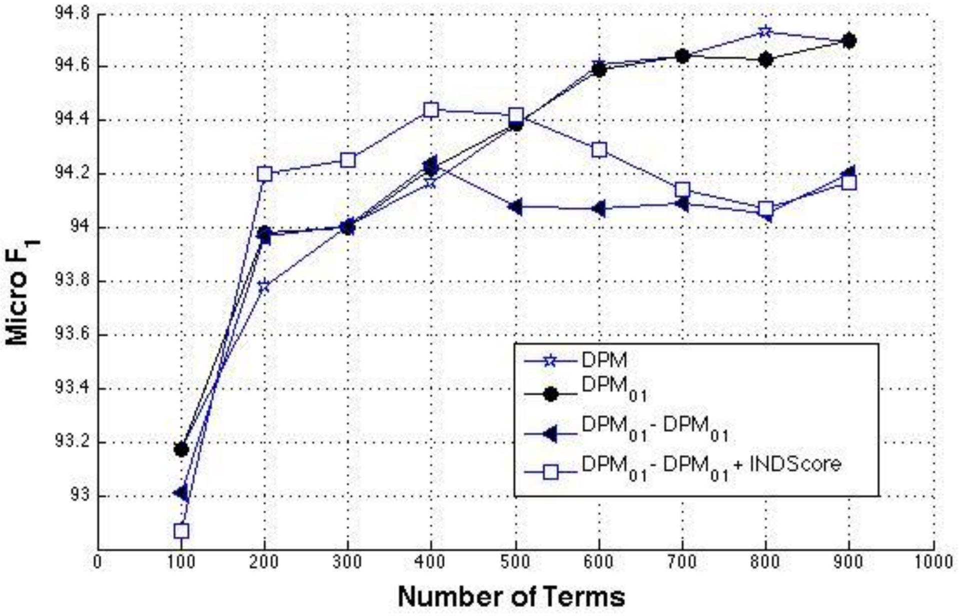


Figure 4.25: The micro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM, DPM_{01} and the proposed individual and pair selection schemes. $01_{score}(t_i)$ is used for $score\ (t_i)$ and $01_{score}(t_i)$ with and without INDScore for $pair_{score}(t_i, t_j)$.

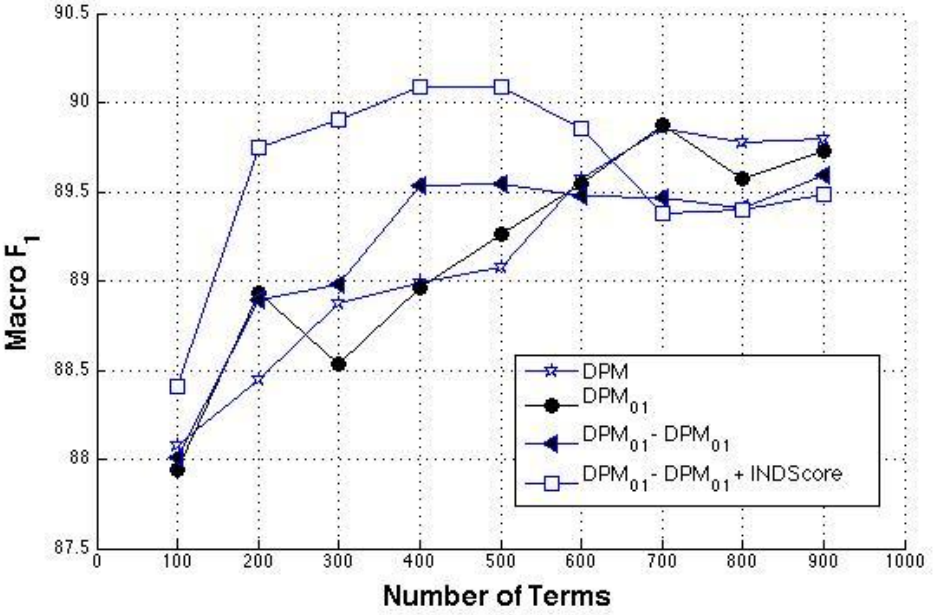


Figure 4.26: The macro $\mathcal{F}_1 score$ achieved on Reuters-21578 by DPM, DPM_{01} and the proposed individual and pair selection schemes. $01_{score}(t_i)$ is used for $score\ (t_i)$ and $01_{score}(t_i)$ with and without INDScore for $pair_{score}(t_i, t_j)$.

As it can be seen in the figures, performance gains are achieved for DPM when the number of features used is less than 500. It can be concluded the use term pairs is not generally within the framework proposed in this thesis.

Chapter 5

CONCLUSION

In this thesis, an iterative scheme is designed for term selection which takes into account the co-occurring statistics of the candidate terms and the previously selected ones. Three term selection schemes are modified for this purpose.

In individual term based approach, the term achieving the highest score that is based on the use of individual and co-occurrence statistics is selected as the next term to be employed. In the alternative scheme, novel features are defined to consider the pairs of different terms as novel features.

Experiments conducted on Reuters-21578 have shown that, $Gini\_index$ and DPM provide better scores than χ^2 when less than 1000 terms are considered. For all three schemes, the use of term frequencies provides higher $\mathcal{F}_1 score$ when compared to the use of document frequencies.

The use of co-occurrence based term selection in an iterative way is observed to provide remarkable improvements for χ^2. For the other selection schemes, better scores are achieved in some cases, especially when small numbers of features are considered.

In this thesis, the proposed approach is evaluated using SVM as the classifier and the experiments are conducted on Reuters-21578 dataset. The experiments should be repeated on more datasets and other classification schemes should be considered. In the case of using pairs of terms as features, the number of pairs is limited to 10% of the total number of features. The effect of different percentages should also be investigated.

REFERENCES

Abe, H., Tsumoto, S., Ohsaki, M., & Yamaguchi, T. (2009). Evaluating learning algorithms composed by a constructive meta-learning scheme for a rule evaluation support method. *Mining Complex Data* , 95-111.

Aggarwal, C., & Zhai, C. (2012). *Mining Text Data* (ebook: document ed). (Springer, Ed.) New York, US.

Altınçay, H. (2013). Feature extraction using single variable classifiers for binary text classification. In Recent Trends in Applied Artificial Intelligence (pp. 332-340). Springer Berlin Heidelberg.

Azam, N., & Yao, J. (2012). Comparison of term frequency and document frequency based feature selection metrics in text categorization. *Expert Systems with Applications* , *39* (5), 4760–4768.

Badawi, D., & Altınçay, H. (2014). A novel framework for termset selection and weighting in binary text classification. *Engineering Applications of Artificial Intelligence* , *35* (0), 38-53.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* , *2* (2), 121-167.

Chen, C.-M., Leeb, H.-M., & Changc, Y.-J. (2009). Two novel feature selection approaches for web page classification. *Expert Systems with Applications , 36* (1), 260–272.

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

Chowdhury, A., Mccabe, M. C., Grossman, D., & Frieder, O. (2002). Document normalization revisited. *Proceeding SIGIR '02 Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 381-382. New York, USA.

Colas, F., & Brazdil, P. (2006). Comparison of SVM and some older classification algorithms in text classification tasks. *Artificial Intelligence in Theory and Practice*, 169-178.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., et al. (1998). Learning to extract symbolic knowledge from the World Wide Web. *Proceeding AAAI '98/IAAI '98 Proceedings of the fifteenth national/tenth conference on artificial intelligence/Innovative applications of artificial intelligence*, 509-516. CA.

Debole, F., & Sebastiani, F. (2004). Supervised term weighting for automated text categorization. In S. Sirmakessis (Ed.), *Text Mining and its Applications*, 81-97. Springer Berlin Heidelberg.

Deng, Z.-H., Tang, S.-W., Yang, D.-Q., Li, M.-Y., & Xie, K.-Q. (2004). A comparative study on feature weight in text categorization. In *Advanced Web Technologies and Applications*, 588-597. Springer Berlin Heidelberg.

Dong, T., Shang, W., & Zhu, H. (2011). An improved algorithm of bayesian text categorization. *Journal of Software , 6* (9), 1837-1843.

Erenel, Z., & Altınçay, H. (2012). Nonlinear transformation of term frequencies for term weighting in text categorization. *Engineering Applications of Artificial Intelligence , 25* (7), 1505–1514.

Erenel, Z., Altınçay, H., & Varoğlu, E. (2011). Explicit use of term occurrence probabilities for term weighting in text categorization. *Journal of Information Science and Engineering , 27* (3), 819-834.

Figueiredo, F., Rocha, L., Couto, T., Salles, T., Gonçalves, M. A., & Meira, W. (2011). Word co-occurrence features for text classification. *Information Systems , 36* (5), 843 - 858.

Fix, E., Hodges, J., & Joseph, L. (1951). Discriminatory analysis-nonparametric discrimination: consistency properties. *CALIFORNIA UNIV BERKELEY*.

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

HaCohen-Kerner, Y., & Yishai Blitz, S. (2010). Initial experiments with extraction of stopwords in hebrew. *KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*. Valencia, Spain.

Hulth, A., & Megyesi, B. (2006). A study on automatically extracted keywords in text categorization. *Proceeding ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics,* 537-544. Stroudsburg, PA, USA.

Ingason, A. K., Helgadóttir, S., Loftsson, H., & Rögnva, E. (2008). A mixed method lemmatization algorithm using a hierarchy of linguistic identities (HOLI). *6th International Conference, GoTAL, August 25-27*, 205-216. Gothenburg, Sweden.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. *Machine Learning , 1398*, 137-142.

Kettunen, K., Kunttu, T., & Järvelin, K. (2005). To stem or lemmatize a highly inflectional language in a probabilistic IR environment *ournal of Documentation , 61,* 476-496.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. *Proceeding SIGIR '92 Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval,* 37-50. NY.

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , 39* (2), 539-550.

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

Man , L., Tan, C., Jian , S., & Yue , L. (2009). Supervised and traditional term weighting methods for automatic text categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on , 31* (4), 721 - 735.

Man, L., Tan, C.-L., Low, H.-B., & Sung, S.-Y. (2005). A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. *Proceeding WWW '05 Special interest tracks and posters of the 14th international conference on World Wide Web*, 1032-1033. New York, USA.

Murphy, K. P. (2006). *Naive Bayes classifiers.* University of British Columbia.

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

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

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

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR) , 34* (1), 1-47.

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

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. *96 Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 21-29. NY,USA.

Tesar, R., Strnad, V., Jezek, K., & Poesio, M. (2006). Extending the single words-based document model: a comparison of bigrams and 2-itemsets. *Proceedings of the 2006 ACM symposium on Document engineering*, 138-146. NY.USA.

Thakur.(2009). Retrieved from ICT Consultants: http://www.thakursahib.com/2009/03/reviews-svm

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 273-314.

Willett, P. (2006). The porter stemming algorithm: then and now. *Program , 40* (3), 219 - 223.

Yang, J., Liu, Y., Zhu, X., Liu, Z., & Zhang, X. (2012). A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. *Information Processing & Management , 48* (4), 741-754.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *Proceeding ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning*, 412-420. San Francisco.

Zhan, J., & Loh, H. (2009). Using redundancy reduction in summarization to improve text classification by SVMs. *Journal of Information Science and Engineering , 25*, 591-601.