Improving Automatic Text Classification by Discriminant Analysis Techniques

Lazaro S. P. Busagala

A Dissertation Submitted to Mie University in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Engineering

Division of Systems Engineering
Graduate School of Engineering

Mie University
Japan

September 2008

Contents

Chapter

Chapter

Chapter

Chapter

Chapter

List of Figures

Figure

List of Tables

Table

Acknowledgments

The author would like to express his sincere gratitude to various individuals and organizations for making this work possible. Profound thanks go to Professor Fumitaka Kimura, Associate Professor Tetsushi Wakabayashi and Dr. Wataru Ohyama for their constructive advice and contributions to this research work. Working with these outstanding individuals led to a successful completion of the preparations and implementation of the research tasks. Since I joined the human interface research laboratory in the department of information engineering, Mie University, I have received kind treatment and cooperation, making it possible for me to do this research project.

Thanks to the other staff of Mie University as well as students of the Department of Information Engineering for their moral support and willingness to share their knowledge in Information Technology field as a whole. My knowledge in machine learning was shaped by various discussions we held on research matters.

I would like to thank the members of the examining committee of this dissertation. They are Professor Fumitaka Kimura, Professor Naoki Isu, Professor Hiroshi Naruse and associate Professor Tetsushi Wakabayashi. Their constructive suggestions contributed to the improvement of this work.

Furthermore I am indebted to the Government of Japan through the Ministry of Education, Culture, Sports, Science and Technology (MEXT) for its support in my graduate studies.

Moreover I extend my sincere thanks to the management of Sokoine University of Agriculture (SUA) for granting me study leave so that it was possible for me to pursue my doctoral studies at Mie University.

Also I acknowledge my wife Elida Simon Busagala for the moral support and care. She devoted time and effort in taking care of our two boys: Ezekiel and Justice enabling me to concentrate on academic work. Her loving heart and kindness always strengthened me mentally and physically. Without her support it would have been more difficult to accomplish this work.

My acknowledgments can not end without expressing my sincere gratitude to my parents, sisters and brothers who have continually supported me in various ways. My parents supported me in my whole life and caused to become interested in learning. They also laid the educational foundations in my life through moral and economical support.

My sisters and brothers also supported me morally and in other aspects of life. Without these individuals this story of accomplishments would have been different.

Furthermore I would like to express my sincere gratitude to Mr. Kari Kostiainen and Mrs. Laurie Mizukami for their excellent work of proof reading.

I also thank all the individuals that were involved in one way or another to make my studies a success.

Last, but most importantly, I would like to thank God for his provision of life, health, peace of mind, salvation through Jesus Christ and all other spiritual blessings. My thanks cannot end without mentioning the body of Christ, the church. The church of Japan under the leadership of Rev. Sachio Fujimoto have always prayed for my studies. The church in Tanzania under the leadership of Dr. Barnabas Mtokambali was similarly devoted to prayers so that I could successfully complete my doctoral studies.

Executive Summary

The development of information technology has increased the availability of documents in digital format. These documents can be stored in databases and can be accessed through the Web or offline. Since the information is always disorganized, users are always overwhelmed by a lot of information, most of which is irrelevant. In order to mitigate this problem electronic information need to be indexed and organized in databases. Text classification (TC) is one of the powerful tools for organizing electronic documents. There are numerous other applications for TC. Examples include spam filtering, automated cataloging, word sense disambiguation, and automated survey coding.

While achievement in text classification has been reported, the performance of classification systems is far from satisfactory. In the context of TC a researcher may wish to address various research problems in the process of modeling texts for improved classification performance. Examples of problems include variation in text length, asymmetry sample distribution, high-dimensional feature space and under-sampled problems. All these hinder the process of machine learning in text classification.

Generally, the objective of this research was to find out ways that would improve classification performance using various statistical techniques and representation of documents. This should go along with extracting more discriminative information from textual data to compose the features. Furthermore, since automatic text classification faces a problem of high-dimensional feature space, it was important therefore to provide solutions to reduce the dimensionality along with improving the performance of classifiers. Another objective was to improve learning ability of the classifiers by applying discriminant analysis methods which simultaneously reduce the dimensionality and extract more informative features to improve the separability of textual data. In short, Chapter 1 of this dissertation describes my research motivation and the objectives. Based on the above objectives the author presents the scientific contributions that lead to improved machine learning in text classification.

Chapter 2 starts by reviewing the literature on machine learning methods that have been proposed and applied to TC. Then the proposed methods for improving machine learning methods are described. In particular, the normalized-weighted metric (NWM) for k nearest neighbor method (kNN) and a posteriori probability (PPD) obtained from distance based learning (DBL) methods are proposed. Experimental results show that

NWM improved the performance of kNN. Empirical results also show that PPD is better than distance based classifiers in text classification. The performance measures for the learning methods are also described.

Chapter 3 presents the details of the contributions based on feature transformation. In particular, it shows that relative frequency with power transformation (RFPT) is better than classical features such as term frequency weighted by inverse document frequency (TFIDF). The transformation of TFIDF also was studied. Experiments were performed using the benchmark text collections. It turns out that the feature transformation proposed in this chapter outperforms its counterparts. The improvements are statistically significant. Furthermore, the contribution of feature transformation on OCR-based document classification is presented. Although OCR texts are noisy, RFPT proved to be very robust such that classification performance is enormously improved.

Chapter 4 deals with feature selection and reduction methods. First, the conventional methods for feature reduction are briefly described. Second, the contributions of this study on feature reduction are presented. There is further discussions on theoretical and experimental studies proposed on dimensionality reduction. Specifically the author studied the combination of the principal component analysis (PCA) and canonical discriminant analysis (CDA). Since text classification involves multi-label data, a solution that extends the PCA+CDA algorithm is provided. Also an integrated discriminant analysis (IDA) technique is proposed. Discriminant analysis methods simultaneously reduce the dimensionality and maximize discriminating power for classification. Multi-label learning tasks are tackled by IDA in a similar way as in the PCA+CDA algorithm. Based on classification effectiveness and the statistical analysis of significance, it was found that IDA outperformed its counterparts in the comparative study.

Chapter 5 proposes feature integration (FI) which further improves the classification performance. The improvements come due to the increased learning ability by the classifiers emanating from the discriminating power of the integrated features. The improved classification effectiveness is validated by statistical analysis. Chapter 5 also introduces a contribution based on multiple features and multi-classifier combination (MCC). Unlike the conventional methods of MCC, we used various features which were separately fed to various classifiers then combined their decisions by majority vote rule. Since this process involves the combination of the features and classifiers, it can be called multiple feature-classifier combination (MFC).

In Chapter 6, the author draws conclusions derived from the empirical results. He then summarizes the main contributions of this work. Based on performance evaluation using classification effectiveness and statistical analysis of improvements, it is argued that the proposed methods are suitable to use in machine learning for automated text classification. Future research possibilities are given in the same chapter. These mainly outline the open research problems which require further research to improve the effectiveness of

automated text classification.

Chapter 1

Introduction

1.1 Background

The development of information technology has increased the availability of documents in digital format [82]. These documents can be stored in databases and be accessed on the Web. Since the information is always disorganized, users are always overwhelmed by a lot of information most of which is irrelevant [9]. In order to mitigate this problem electronic information needs to be indexed and organized in databases. Text classification (TC) is one of the powerful tools for organizing electronic documents.

TC tasks date back to the early 60's. Until the early 90s, it became a major subfield of information systems. Recently it is being applied in different tasks ranging from automated survey coding, document indexing to organizing web pages [41, 82].

The earlier approach of TC is known as *knowledge engineering* (KE) which involves defining a set of classification rules manually. The disadvantages of this approach include the following:

(i). it needs intervention from knowledge engineers or domain experts.

(ii). it is time consuming.

(iii). Third, it can be very tedious and inconsistency can arise as the set of rules gets large.

The recent approach is to use the *machine learning* (ML) techniques. This approach has two main advantages:

(i). the classifiers are automatically learned by providing examples to the classification systems; and

(ii). it is less costly in the long run.

In general text classification can be considered as a task of categorizing documents according to predefined categories. Automated Text Classification (ATC) is the task of automatically assigning a set of documents to appropriate categories (or classes, or topics) [82, 49]. Other names for ATC include Automatic Text Categorization or Automatic Topic Spotting. ATC can simply be called Text Classification. Therefore it is noted that text classification (TC) also means ATC in this dissertation. In general terms, ATC is at the crossroad of information retrieval (IR) and machine learning (ML) disciplines.

Sebastiani [82] argues that researchers on the side of IR, are interested in one particular aspect of a general movement towards leveraging user data for controlling the inherent subjectivity of the IR task, i.e. establishing the fact that it is the user, and only the user, who can say whether a given item of information is relevant to a query s/he has issued to a Web search engine, or to her/his private folder in which documents are filed.

Wherever there are predefined classes, documents that are manually classified by the user are often available. As a consequence, this data can be exploited for automatically learning the (extensional) meaning that the user attributes to the classes, thereby reaching levels of classification accuracy that would be unthinkable if this data were unavailable.

ML researchers are interested in ATC due to the fact that IR applications are challenging benchmarks for their techniques and methodologies. This is because IR applications usually feature extremely high dimensional feature spaces and involve truckloads of data. ML researchers therefore are adopting TC as one of their benchmark application. This means that cutting-edge ML techniques are being incorporated into TC from their original purpose.

Various machine learning techniques have been proposed and applied to TC. These include probabilistic methods such as Bayesian classifiers [26, 40, 61]; decision tree methods such as C4.5 in [26, 40, 41]; regression methods [94]; instance-based methods such as k nearest neighbor (kNN) [41, 85, 96]; support vector machines (SVMs) [26, 40, 41, 61]; and classifier committees (ensembles) [5, 33, 52]. This work studies distance based learning (DBL) methods. Furthermore it proposes methods to improve kNN and DBL methods.

Most of the ML methods have used the conventional way of representing documents. The classic way of representing texts by the use of term frequency weighted by inverse document frequency (TFIDF). Furthermore there are various methods for feature selection and reduction. State of the art in feature selection and reduction is discussed in Chapter 4.

While a practical achievement has been reported in the literature, the performance of text classification is still far from satisfactory. This work therefore proposes various methods to improve automated text classification. In contrast to the way of representing documents this work proposes the use of relative frequency with power transformation (RFPT). Furthermore this work proposes various methods for feature reduction and proposes methods which prove to be effective in improving the learning process that lead to

better classification performance. Also, we propose a novel technique for feature integration. An overview of the contributions of this work are presented in Section 1.5

1.2 Overview of Text Classification

In this section the overview of the text classification process is provided. In general there are four steps in automated text classification. These include: feature generation, feature reduction, learning or training the classifier and classification. Each step is briefly introduced below.

(i). Feature Generation

The importance of generating features is to represent textual data in such a way that the learning algorithm can recognize them easily. There are various ways of generating features from textual data. The vector model approach is commonly used. The most commonly used vector model method involves the use of words or terms. The literature shows that terms are the most used and probably have proved to be the most effective way to generate features. In other words every document is converted into word tokens which form the feature vector. More complicated representations do not lead to better classification effectiveness [2, 26] which confirms similar results in information retrieval [80].

The following example illustrates document representation using the feature vector model. Assume that the feature vectors are defined as $\mathbf{x} = [x_1, \ldots, x_n]^T$, where x_i is the term frequency in every vector formed from each document. Assume the following items represent a text collection.

(a) Artificial intelligence subfields

(b) Pattern recognition

(c) There are other examples. There are many examples

The following is the vocabulary list (also known as lexicon) for this example.

are, artificial, examples, intelligence, many, other, pattern, subfields, recognition, there

From this vocabulary list we can obtain feature vectors as follows:

(a) $[0, 1, 0, 1, 0, 0, 0, 1, 0, 0]^T$

(b) $[0, 0, 0, 0, 0, 0, 1, 0, 1, 0]^T$

(c) $[2, 0, 2, 0, 1, 1, 0, 0, 0, 2]^T$

Conventionally, after generating features, term weighting is performed. In this work, instead of term weighting, feature transformation is carried out. A comparative

study shows that feature transformation is better than conventional term weighting. Chapter 3 gives further details on feature generation and transformation.

(ii). Feature Reduction

It is common practice not to use all the terms found in the collection. The reasons for feature reduction include improved computational speed and sometimes improved classification performance. In the literature there are various methods for feature reduction. Examples include document frequency [97], mutual information [82, 97, 93] and principal component analysis (PCA) [7, 24, 32]. Chapter 4 gives more details on these methods.

(iii). Learning

The objective of learning is to extract information from available document examples so that the classifier can later classify unseen documents. There are various methods for learning. Chapter 2 reviews and introduces the learning methods.

(iv). Classification

The task of assigning the unseen documents to various labels is what is referred to as classification or categorization. After learning the system is expected to be able to classify unseen textual data automatically. Depending on the need and on the system classification can be offline or on-line. In the classification experiments the offline approach was used.

1.3 The Organization of This Dissertation

The organization of this dissertation is as follows. The proceeding section 1.4 presents the motivation for the research in general. The importance of the findings are given in Section 1.4.1. The author briefly describes the application of text classification in Section 1.4.2. The description of the research problem is presented in Section 1.4.3. The objectives are stated in Section 1.4.4.

Chapter 2 starts by reviewing the literature on machine learning methods that have been proposed and applied to text classification (TC). Then the proposed methods for improving machine learning methods are described. In particular, normalized-weight metric (NWM) for k nearest neighbor method (kNN) and *a posteriori* probability based on distance learning (DBL) methods are proposed. The performance measures for the learning methods are described in 2.5.1.

Chapter 3 goes into the details of the contributions based on feature transformation. Particularly it shows that relative frequency with power transformation (RFPT) is better than the classical method of representing documents using term frequency weighted by

inverse document frequency (TFIDF). The transformation of TFIDF was also studied. It is apparent that the feature transformation proposed in this chapter outperformed its counterparts. Section 3.6 presents the impact of feature transformation on OCR-based document classification.

Chapter 4 deals with feature selection and reduction methods. First, the conventional methods for feature reduction are briefly described. Second, the contributions of this study on feature reduction are presented. Chapter 4 also includes a discussion on theoretical and experimental studies for proposed dimensionality reduction methods. Specifically we studied the combination of principal component analysis (PCA) and canonical discriminant analysis (CDA). Also an integrated discriminant analysis (IDA) technique is proposed in 4.3.2. It was found that IDA outperforms its counterparts in the comparative study.

Chapter 5 proposes feature integration (FI) which further improves the classification performance. The improvements come due to the increased learning ability by the classifiers emanating from the discriminating power of the integrated features. Chapter 5 also proposes a technique namely feature-classifier combination which achieved the highest classification performance.

In Chapter 6 the author draws conclusions derived from the empirical results. He then summarizes the main contributions of this work. Future research possibilities are given in the same chapter. These mainly outline the open research problems which call for further research to improve the classification effectiveness.

1.4 Motivation for the Research

1.4.1 Importance of Text Classification Research

In recent years there has been increased availability of digital documents stored in-house or on line. For example, some researchers have argued that a lot of scientific literature is available on the Internet but it is disorganized [9, 72]. Increased availability of the information creates a need for flexible and convenient access [82]. Automated text classification that involves the activity of labeling natural language texts with thematic categories is important in developing retrieval systems. Much of the work of organizing documents therefore can be automated through text classification.

Also, the importance of automated text categorization lies in the fact that it frees organizations from the need to manually organize document databases, which can be expensive and not feasible in terms of time constraints.

In addition, the findings have a role of increasing the literature base and information access for various purposes such as for research and other developmental programs. It is argued that information usage increases when access is more convenient. Maximizing the

usage of scientific literature benefits the whole society [55].

Another viewpoint of the importance of this study is the applications of the findings in general. We devote Section 1.4.2 to describe various applications of ATC. It is clear that a number of fields and daily activities use ATC.

1.4.2 Applications of Text Classification

The importance of this research work can be viewed in terms of its applications. In this Section various applications of automated text classification are introduced. There are number of applications of text categorization. It is importance to have techniques that give the best possible performance. The following are a few examples.

1.4.2.1 Information Retrieval

Efficient and effective information retrieval (IR) is very important in information systems. Similar information will always tend to be relevant to similar queries [18]. Once similar information is grouped together the retrieval process can be improved in terms of efficiency and effectiveness. The activity of grouping textual information for retrieval can be carried out using TC techniques. ATC improves the performance of the process of information retrieval [50].

1.4.2.2 Automatic Indexing

Automatic indexing can be defined as the assignment of content *identifiers* by the aid of modern computing equipment. The advantages of automatic indexing include:

- the high possibility of maintenance of consistent indexes;
- index entries are generated at lower cost in the long run;
- a reduction of indexing time can be reduced; and
- achievement of an improved retrieval effectiveness [18, 79].

Automatic indexing is essential in tools such as search engines. Web search engines collect information from the web and automatically index such information and store it in databases to facilitate fast information retrieval (IR). The indexed information is used in the search process when the user supplies a query [4].

Automatic document indexing for IR systems can rely on a controlled dictionary [8, 82]. Each document is assigned one or more keywords or phrases describing its content. The keywords and phrases form a finite set called controlled dictionary or vocabulary. A good example of this can be the medical subject heading (MeSH) in the discipline of medicine. One can view the controlled vocabulary or *identifiers* as categories. Therefore TC can be applied in the process of indexing.

1.4.2.3 Document Organization

Document organization can be considered to be a general area of ATC application. Text classification can be used in organizing documents whether online or offline. If there is a set of categories, incoming documents can be filed to an appropriate category automatically. For example organizing patents into related disciplines [51]. Automatic grouping of newspaper articles can be seen as part of the document organization process, too.

1.4.2.4 Automated Cataloging and Metadata Generation

A catalog is a key to a library's collection as each catalog entry contains the bibliographic details of a particular document. It is an organized list of documents in a library with entries representing the documents arranged for access in some systematic order. The catalog is important since it enables a user to find a book or any form of a document in a library [18].

In traditional librarianship cataloging and classification of library materials is done by trained individuals manually. This can be tedious and laborious. By employing text classification techniques these tasks can be done automatically [37]. In digital libraries, documents can be tagged by metadata such as creation date, document type or format and availability. Manual cataloging of Internet resources is not feasible.

1.4.2.5 Information Filtering

With the development of the web, filtering electronic information has become an important task. Examples are the filtering of pornography and junk e-mail (spam). Internet users are facing these problems as they use the Web. This task can be tackled as an ATC problem.

The problem of junk email is growing daily. Siefkes [17, 84] argues that spam is ubiquitous and is one of the most annoying things on the Web. Spam filtering is one of the important applications of text classification. Text classification can play a big role in solving the problem. The learning system can be provided with the information incrementally which can be learned to filter any spam information.

1.4.2.6 Hierarchical Categorization of Web Pages

Categorized Web pages may be useful to users. One can find it easier to navigate to a hierarchy of categories and restrict the search to a particular category of interest. The advantages of automatically categorizing/classifying Web pages has advantages such as removing the infeasibility of doing it manually. A number of authors have shown that ATC can be used in this task [14, 25, 68, 82].

1.4.2.7 Word Sense Disambiguation

In natural language many words can have several meaning or *senses* [66]. Such words lead to ambiguity if interpreted out of context. For example the word 'light' may mean not very heavy or not very dark.

Word sense disambiguation (WSD) is the task of determining which of the senses of an ambiguous word is invoked in a particular context. WSD can be framed as a multi-class text classification task. A number of authors have explored this task [27, 39, 73]. According to Escudero et al. [27], we can view word occurrence contexts as documents and word senses as categories.

It is argued that WSD can be treated as a single-label TC case. One can consider that WSD is just one example of solving natural language ambiguities. Other examples include context-sensitive spelling correction, prepositional phrase attachment, speech tagging and word choice selection in machine translation [78, 82].

1.4.2.8 Automated Survey Coding

Survey coding can be defined as the task of assigning a symbolic code to an answer given by respondents. The codes are taken from a set of predefined codes and questionnaires are usually used in getting responses from the participants. Open ended questions can be used in questionnaires. The applications of survey coding include classification of respondents in order to extract statistics on politics, customer satisfaction and life style habits.

Survey coding is one of the good applications of text classification [35]. The task of automating survey coding can be done using text classification techniques. The set of all answers to a given question are represented as documents. The set of all possible codes that may be attributed to an answer to a question represent the set of categories. Therefore the task corresponds to that of TC. Classifiers are automatically built using machine learning techniques and association between answers and codes (categories) is done. This can be easily done when precoded answers used in the training set are available. Then new answers can be automatically associated (coded) by using the examples provided to the system in the training phase.

1.4.3 Research Problem

Although achievement in text classification has been reported, the performance of classification systems is far from satisfactory. Text classification tasks are characterized by natural languages (NL). This means TC is closely linked to natural language processing (NLP) which needs knowledge on its subject matter. In general NL reveals a lot of syntactic and semantic ambiguities as well as complexities [66]. In the context of TC a

researcher may wish to address various problems arising from document properties in the process of modeling texts; or else problems emanating from the learning algorithms. The following sections provide ideas on research problems.

1.4.3.1 Problems Emanating From Textual Data

(i). Variation in Text Length

In practice, textual data vary in content and length. It is common to have very short, medium length or very long documents in a collection. This can have a negative impact in representing textual information for classification. This is because the words contained in the documents are usually used in document representation. Short documents have fewer words and long ones have more words.

Term or word frequency are conventionally used to represent documents in classification systems. Absolute word frequency has the drawback of depending on text length leading into lower classification performance. This is because text length may differ within the same class of documents leading to difficulties in the learning process [12, 10, 82].

The problem of variation in text length has been conventionally tackled by normalizing the feature vectors using the concept of the unit length of a vector. This has been applied to term frequency (TF) weighted by inverse document frequency (IDF) which is abbreviated as TFIDF. Examples can be found in [40, 41, 56, 82, 85, 95, 100].

While this kind of normalization can partially solve the problem, theoretical analysis have pointed out that TFIDF is basically suited for information retrieval problems [43, 77](see Section 3.1).

Therefore we propose the use of feature transformation techniques which include normalizing absolute word frequency to relative word frequency (RF) and power transformation (PT). When RF is transformed using PT we abbreviate the name of these features as RFPT. This was essentially studied in comparison with conventional methods.

(ii). Asymmetric Sample Distribution

Sample distribution of real world data may be skewed. This is especially true when one deals with texts. This is undesirable particularly for parametric classifiers such as linear or quadratic classifiers which are typically designed for Gaussian distributions. TFIDF do not take care of sample distribution leaving it at a risk of being skewed.

This skewness can lead to classification errors by the classification systems. Conventional methods of representing texts always ignore the fact of skewness. We provide a remedy in Section 3.3. Results of the experiments show that RFPT improves the symmetry of sample distribution, leading to higher classification effectiveness.

(iii). High-Dimensional Feature Space

Extremely high-dimensional feature spaces and large volumes of data problems occur in automatic text classification. High dimensionality problems arise because the number of words used in the classification process increases along with dimensionality of the feature vectors [12, 10, 41, 82]. Practical examples show that the number of features consisting the dimensionality could amount to thousands.

In order to solve this problem the feature vector dimensionality is required to be reduced without degradation of classification performance. It was important to extract the features with high discriminating power using various techniques. Chapter 4 treats this problem in detail.

(iv). Frequency Distribution of Words

The frequency distribution of words in text collection can give an insight on what kind of words should be included in the classification process. It has been argued that frequency distribution of words follow what is called Zipf's law [101]. His law states that given a (large) corpus of natural language, the frequency of occurrence of any word is inversely proportional to its rank/position in the word list. More formally, let x_i denote word frequency of a word i and r be the rank of that word (position of the word in the list). Zipf's law gives the reciprocal relationship as

$$x_i \propto \frac{1}{r} \quad \text{or if there exists a constant } k \text{ then } x_i.r = k. \tag{1.1}$$

Mandelbrot [65] however noted that Zipf's law is not good in reflecting details. He therefore provides a "generalized" relationship between rank and word frequency

$$x_i = c(r+s)^{-b} \quad \text{or} \quad \log x_i = \log c - b\log(r+s), \tag{1.2}$$

where c, s and b are parameters of a text that measures the richness of the text's use of words. Experimental study to verify these formulae have been done by a number of authors [41, 66]. It is apparent that the Mandelbrot formula provides the better fit than Zipf's law.

The implication of these laws is that a small number of words occurs very frequently, while most words occur very infrequently. This concept is important in feature selection. Words that occur very frequently, such as articles, can occur in

every document – hindering separability of documents according to their categories. Furthermore the use of all words in the text collection may increase the complexities in machine learning due to the curse of dimensionality. Therefore only words that can provide high performance are preferable. In the experiments therefore words that occur very frequently are usually removed. This was achieved by the use of a stop word list.

(v). Under-sampled Data

Due to high-dimensional feature space, the size of the training sample may always be smaller than its dimensionality. This hinders the application of the classical discriminant analysis (DA) [12, 13, 11, 24, 32]. In chapter 4 we provide a remedy to this problem.

(vi). Imbalanced Sample Size

In text classification, binary classification tasks are common. This involves the class of interest (positive class) and the rest of training data (negative class). The TC learning tasks are always featured with smaller training sample sizes particularly in the positive class. The performance of learning systems is usually influenced by the class sample size imbalance in which samples in the data belonging to one class heavily outnumber the examples in the other [3, 47, 53, 54]. Consequently, the number of errors can be high.

1.4.3.2 Problems Emanating From the Learning Algorithms

(i). Weakness of learning algorithms can be in obtaining enough information for decision making. For example the Euclidean distance algorithm uses only the vector mean of a class in obtaining information for making classification decisions. The class mean vector may easily be affected by outliers. As such it can make a lot of errors in classification. Furthermore the class mean vector alone does not give enough information of the distribution of a sample. Another example is when one considers the classical k nearest neighbor (kNN) method. The classical kNN gives equal weight to all examples regardless of distance or similarity value. This can lead to errors especially when furthest data points outnumber the closest.

(ii). Learning algorithms can be numerically unstable. For example, the classical linear discriminant function can usually face the singularity problem when the sample size is smaller than the dimensionality. This is because it requires the inverse of the within-class matrix which may not exist in practice.

With the aim of solving these problems above, various techniques are proposed and experiments were carried out to verify their effectiveness.

1.4.4 Objectives

In general, the objective of this research was to find out ways that would improve classification performance using various statistical techniques and representation of documents. This should go along with improving separability of textual data.

Specifically, problems presented in Section 1.4.3 were tackled to improve classification performance. In other words, we had to provide a way of avoiding variation in text length per document. Therefore, normalizing absolute word frequency to relative word frequency and power transformation techniques are proposed in this research work.

In particularly, improvement in text classification was found by the use relative frequency with power transformation (RFPT). The goal of using RFPT is to avoid variation in text length and to improve the sample distribution by removing the skewness and normalizing the kurtosis. In other words, RFPT seeks to represent documents such that their distributions are in a Gaussian-like form.

Furthermore, since automatic text classification faces a high dimensional feature space, it was important therefore to provide solutions to reduce the dimensionality along with improving the performance of classifiers.

Another objective was to apply discriminant analysis methods which simultaneously reduce the dimensionality and improve the separability of textual data. Finally, it was an aim to improve learning algorithms in regard to the problems presented in this work.

1.5 Scientific Contributions

In this Section, we outline the scientific contributions as a consequence of this study. This work proposes various techniques for improving automated text classification. The following items summarize the general contributions that lead to improved machine learning.

(i). Text Representation by Transformed Features

The traditional way of representing textual data is by the use of term frequency weighted by inverse document frequency (TFIDF). In contrast, we propose the relative term frequency with power transformation RFPT in Chapter 3. Empirical results and statistical analysis show that RFPT is better than TFIDF. Furthermore we employed power transformation on TFIDF, and we refer to it as TFIDF+PT. Empirical results show that TFIDF+PT outperforms conventional TFIDF.

(ii). Empirical Study with OCR-based Texts

Optical character recognition (OCR) is applied in various areas in daily life. OCR-based texts are full of errors and so the conventional methods are not effective enough. In this work the proposed RFPT was applied to these kinds of data and

empirical results show that RFPT is suitable to use in this context. Section 3.6 is the topic of this contribution. The experimental study with noisy data shows that RFPT is robust regardless of the use of error-prone OCR texts.

(iii). Feature Reduction

Chapter 4 proposes various methods for dimensionality reduction. In the first place we experimentally studied principal component analysis (PCA) which is not common in ATC. We noted setbacks in the PCA method and experimentally studied the canonical discriminant analysis.

However we noted that due to the reason described in Section 1.4.3(v), CDA can't be a good choice for ATC. Therefore we studied the PCA+CDA algorithm. The classical CDA can't handle multi-label problems. Since ATC can involve multi-label data, we extended CDA and the PCA+CDA algorithms to handle multi-label learning tasks.

(iv). Integrated Discriminant Analysis (IDA)

In Chapter 4, we further propose an integrated discriminant analysis (IDA) which outperforms its counterparts. The multi-label setting is tackled in a similar way as with the PCA+CDA algorithm. A comparative study was also carried out. Finally, we conclude that IDA increased the learning ability of various methods. This conclusion is based on the improved classification effectiveness and the statistical significance of the improvements.

(v). Normalized-weighted metric for k Nearest Neighbor (kNN)

This work proposes a method called normalized-weighted metric (NWM) for the kNN learning method or simply the kNN classifier. As it will become clear in the following chapters, NWM improves the performance of kNN. We describe NWM in 2.4.1.

(vi). Distance Based Learning Methods (DBL)

To the best of our knowledge DBL methods studied in this work are not seen in the TC literature. Therefore DBL were experimentally evaluated as a preliminary study in 2.3.1.

Furthermore we propose the use of *a posteriori* probability (PPD) based on DBL methods in 2.4.2. An empirical study shows that PPD is far better than the use of distance classifiers in text classification.

(vii). Feature Integration

Chapter 5 presents another contribution. It proposes a feature integration (FI) technique that generates composite features with higher discriminating power. Experimental results show that FI improves the performance of ATC.

(viii). Multiple Feature-Classifier Combination (MFC)

Chapter 5 also introduces a contribution based on multiple features and multi-classifier combination (MCC). Unlike the conventional methods of MCC, we used various features which were separately fed to various classifiers and then combined their decisions by majority vote rule.

Based on these contributions, Chapter 6 provides concluding remarks. Furthermore it outlines the open research problems which provide further areas for investigation to achieve further ATC improvements.

1.6 Summary of the Introduction

In this chapter, we have seen that in TC, there are knowledge engineering (KE) and machine learning (ML) approaches. The ML approach is the subject of this study and it is also called automated text classification (ATC). However both TC and ATC will be used inter-changeably in this document.

In Section 1.2, the author gives a brief overview of the text classification process. The motivation for the research in 1.4 is based on the elements of research problem, objectives and the importance of TC in general. The importance includes the numerous applications of TC presented in Section 1.4.2. The organization of this dissertation is presented in 1.3 which briefly describes the topics for each chapter.

We have also outlined a number of contributions by this work in Section 1.5. These contributions are discussed in detail in the chapters to follow. We describe these techniques in detail meanwhile introducing the contributions in the rest of the chapters.

Chapter 2

Learning Methods

2.1 Introduction

In general there are two approaches to text classification. These are the knowledge engineering (KE) and machine learning (ML) [82]. The knowledge engineering approach involves defining a set of rules encoding from expert's knowledge on how to classify documents under a given set of categories. This approach was popular until late in the 80s. Manually defined sets of rules are common in KE. For example one rule can be defined as:

$$\text{If}(DNF formula) \text{ then } (category).$$

A DNF (disjunctive normal form) formula is a disjunction of conjunctive clauses; the document is classified under *category* if and only if it satisfies the formula, that is, if and only if it satisfies at least one of the clauses. The most popular example is the CONSTRUE system [19], developed by the Carnegie Group for Reuters news agency.

The main disadvantage of the KE approach is the need for intervention from knowledge engineers or domain experts. If the set of classes is updated professionals should intervene again, and if these rules are taken to different domains the work must again begin from scratch. So it does not free organizations from the need for human expert interventions. Therefore, it can be quite constraining in terms of cost and time [82].

The ML approach has been popular since the early 90s. It involves building a classifier through learning a classification scheme from labeled training examples [24]. This refers to supervised learning since the learning happens from examples of categorized datasets. Unsupervised learning or clustering does not need manually labeled examples. The system forms clusters or groups or classes that are dependent on the algorithm in use.

The advantages of the ML approach are higher effectiveness comparable to those achieved by human experts; it saves expert labor power and time, because there are no human expert interventions every time the classification task has to be carried out; and

it is portable to different domain applications. It is portable in the sense that if it is given examples from different domains, it automatically learns from those examples and easily classifies unseen documents [82]. Unsupervised machine learning seems to be rare in automatic text categorization research. Possibly this area can be a challenging task to be carried out in the near future leading to organizations being freed in making examples.

In this chapter, machine learning methods are introduced. The first aim of the chapter is to present the state of the art in the literature on machine learning methods. This is mainly the objective of Section 2.2. The second aim is presented in Section 2.3. In particular we introduce learning methods which are not common in TC literature. In the same section we shall further describe proposed learning methods which improve the classification performance of the conventional ones.

2.2 Popular Learning Methods

2.2.1 Multinomial Naive Bayesian Classifier

In this section we present Bayesian classifiers which are popular in the TC literature [26, 40, 41, 45, 61, 59]. In general there are two event probabilistic models which are common. The first one is the multivariate Bernoulli event model. The second is the multinomial event model [45].

The multinomial model is usually referred to as multinomial naive Bayes (MNB). This has been reported to outperform the multivariate one [45, 67]. We give a brief review of MNB below. Further details of the multivariate model can be found in references provided.

MNB classifier finds *a posteriori* probability $P(\omega_j|\mathbf{x}_k)$ that document $\mathbf{x}_k$ belongs to class ω_j. Let

$$P(\omega_j) = \frac{N_j}{N} \tag{2.1}$$

be *a priori* probability that a document $\mathbf{x}_k$ belongs to class ω_j such that N_j is the number of documents in class ω_j and N is the total number of all documents in a collection.

The multinomial naive Bayes assigns a test document $\mathbf{x}_k$ to a class that has the maximum $P(\omega_j|\mathbf{x}_k)$ using Bayes' rule which is given as

$$P(\omega_j|\mathbf{x}_k) = \frac{P(\omega_j)P(\mathbf{x}_k|\omega_j)}{\sum_{j=1}^{C} P(\omega_j)P(\mathbf{x}_k|\omega_j)}. \tag{2.2}$$

$P(\mathbf{x}_k|\omega_j)$ is the probability of observing a document $\mathbf{x}_k$ in class ω_j.

The estimation of $P(\mathbf{x}_k|\omega_j)$ in (2.2) is difficult in practice, because the the number of possible document vectors $\mathbf{x}_k$ can be too large. Also it is almost impossible to collect enough training samples without prior knowledge or assumptions. Therefore it is common

to assume that any two coordinates of the document vector are statistically independent of each other [41, 82]. This assumption leads us to an equation written as

$$P(\mathbf{x}_k|\omega_j) \approx \prod_{i=1}^{n} P(x_i|\omega_j), \tag{2.3}$$

where x_i ranges over the sequence of words in document $\mathbf{x}_k$ and n is the number of words in document $\mathbf{x}_k$. The approximation of $P(\mathbf{x}_k|\omega)$ is reduced to estimating each $P(x_i|\omega_j)$. The probability

$$\hat{P}(x_i|\omega_j) = \frac{1 + TF(x_i, \omega_j)}{n + \sum_{i=1}^{n} TF(x_i, \omega_j)} \tag{2.4}$$

is estimated from the training documents. Where $TF(x_i, \omega_j)$ is the count of the word x_i in all training documents belonging to class ω_j. The Laplace estimator is employed to add one to each word frequency to avoid the zero-frequency problem [40, 41, 45].

2.2.2 Decision Tree Learning Methods

Decision tree learners (DTL) are symbolic algorithms which use a tree in which internal nodes are labeled with terms. The branches attached to the nodes are labeled by tests on the weight that the term has in the unseen document $\mathbf{x}_k$. The leaves have category labels. DTLs classify a test document $\mathbf{x}_k$ by recursively testing for the weights that the terms labeling the internal nodes have in vector $\mathbf{x}_k$, until a leaf node is reached. Then the label of such a node is assigned to $\mathbf{x}_k$. Most of DTLs use binary document representations and as a result they consist of binary trees [82].

One of the most common DTL algorithms is the C4.5 [19, 26, 40, 41, 60, 75, 82]. It outputs confidence values when classifying new examples. The output values are used to obtain recall/precision tables. Further details can be found in references provided.

2.2.3 Support Vector Machines

The support vector machine is a machine learning method which has attracted attention since the second half of the 1990s. The SVM is one of the linear classifiers that use linear boundaries with margins such that the data from two categories are separated by the hyperplane with the largest margins. In other words, a support vector machine consists of finding the optimal hyperplane which is the one with maximum distance from the nearest training data. The support vectors are those patterns nearest from the hyperplane [24].

Fig. 2.1 illustrates the decision boundary of two categories. SVM can be extended to nonlinear functions when combined with kernel functions. For further reading, references [20, 41] have more detailed explanations about the theory of SVMs.

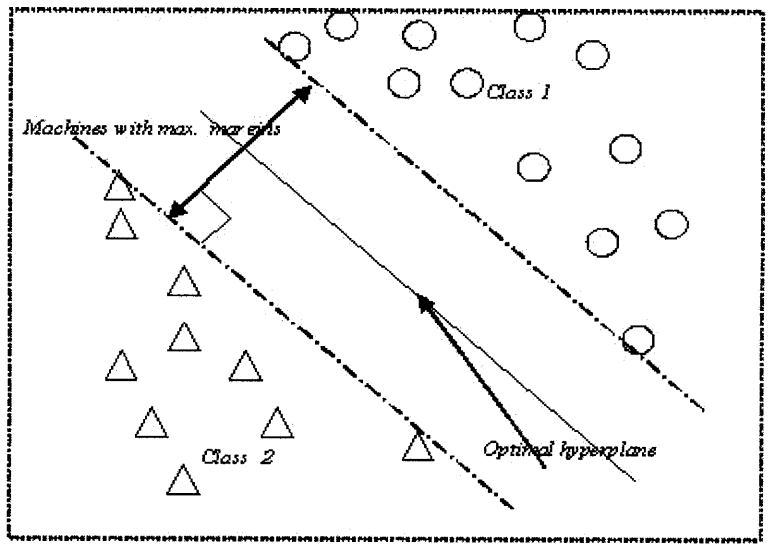


Figure 2.1: SVM margins of two categories

In the classification experiments, we used the LIBSVM package*. The linear kernel (Linear SVM) and radial basis function (RBF SVM) were adopted[15]. We further used the SVM^{light} package† [41] in experiments that need to give out values for recall, precision and F-measure (see Section 2.5). In this case we used the linear SVM and polynomial SVM.

2.2.4 k Nearest Neighbor (kNN)

Although various classification methods have been proposed in the literature, k nearest neighbor (kNN) is one of the best performers [48, 76, 82, 96, 100]. The kNN algorithm relies on the concept that given unseen document $\mathbf{x}$, the learning system finds the k nearest neighbors in the training document set D to predict its category [96].

The system assigns $\mathbf{x}$ to the class that appears most frequently within the subset $D_k \subseteq D$. In short this method requires:

- an integer k preferably an odd number to avoid ties in the decision.
- a set of labeled examples which are referred to as training data set D. These examples are called instances. Therefore the kNN method is grouped under instance based method.
- a metric to measure or determine the nearest or closest examples. A distance metric such as those described in Section 2.3.1 can be used. One of the popular metric is

*The software can be freely obtained at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/` Much appreciation to Lin's research team for availing the software and their support.

†This package can be freely obtained at `http://svmlight.joachims.org/`. I am grateful to acknowledge Thorsten Joachims for availing the software and his support.

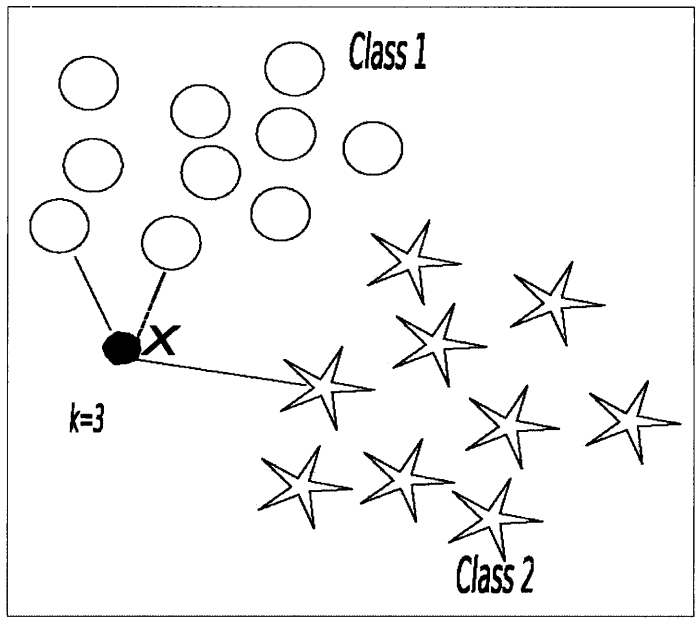


Figure 2.2: kNN learning method for two categories

the Euclidean distance defined by equation (2.8). Furthermore similarity measures such as the cosine function

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.5}$$

can be used.

- a threshold t, particularly when binary classification decisions are used. Conventionally, binary classification facilitates the use of performance measures such as recall, precision and the F_β-measure described in Section 2.5.1.4.

Fig. 2.2 illustrates the concept of the kNN learning method. In this Figure, $k = 3$. Since there are two closest examples in class 1 and one example in class 2, the decision would be that sample $\mathbf{x}$ belongs to class 1. This way of predicting the unseen data is called majority vote rule (MVR). This can be defined as

$$k_j = \max\{k_1, \cdots, k_C\} \rightarrow \mathbf{x} \in \omega_j, \quad k = k_1 + \cdots + k_C, \tag{2.6}$$

where k_j is the number of neighbors from class ω_j, $(j = 1, \cdots, C)$ among kNN examples. [32]. This can also be called discrete metric function (DMF) in [64].

Instead of using expression (2.6) directly, *a posterior* probability $P(\omega_j|\mathbf{x})$ can be preferably estimated as

$$P(\omega_j|\mathbf{x}) = \frac{k_j}{k}. \tag{2.7}$$

Then we have the property $0 \leq P(\omega_j|\mathbf{x}) \leq 1$. With this property, the task of determining

the threshold is simplified. The threshold is correlated to the number of neighbors k. We used equation (2.7) in the experiments.

Furthermore, kNN can easily handle both multi-class and multi-label problems simultaneously as opposed to other classification methods. This is because it can use class local information flexibly. Figure 2.3 illustrates how kNN can deal with these problems

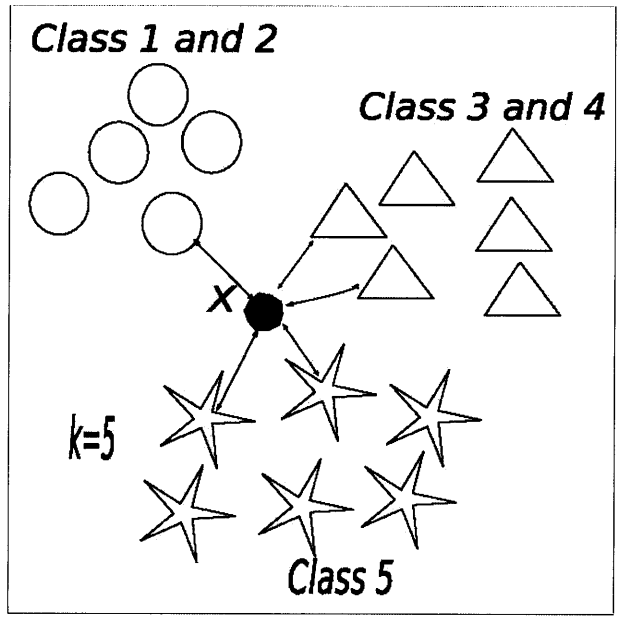


Figure 2.3: kNN learning method for multi-class multi-label case

simultaneously in the vector space. In this figure it can be seen that there are 3 groups of data belonging to 5 categories. The first group belongs to class 1 and 2; the second group belongs to class 3 and 4; and the third one belongs to class 5. Suppose the threshold $t = 0.4$. Assume the decision rule for classification is $\mathbf{x} \in \omega_j$ when $P(\omega_j|\mathbf{x}) \geq t$, then an incoming document $\mathbf{x}$ would be classified into categories 3, 4 and 5. This flexibility is a great advantage of the kNN method. Since the Reuters-21578 and OHSUMED collections are both multi-class and multi-label problems, kNN was used in the classification experiments.

Motivated by this flexibility, the binary category assignments were obtained by specifying a threshold in experiments. The k value and the threshold can be determined by using a validation set or by use of cross-validation techniques. The k value was experimentally varied from 1 until when the classifier could give more errors rather than improving the performance.

The kNN method can be considered to be a lazy learning algorithm. This is because it defers learning and processing the training data until it receives a request to classify unseen data. After classifying the data it discards the constructed prediction and any immediate results.

This is in contrast to other methods which can be referred to as eager learning algo-

rithms. They process the training data and store it in a compressed description or model such as in SVMs. They discard the training data after the training phase. They then classify the incoming sample using the constructed model.

The trade-offs can be realized between the eager learning and the lazy learning algorithms. Lazy algorithms have lower computational costs than eager algorithms during the training phase. In contrast, lazy algorithms have higher storage requirements and higher computational cost during classification.

In summary kNN have various advantages such as simple implementation, and use of local information which can lead to highly adaptive behavior. The disadvantages include large storage requirements and highly susceptibility to the curse of dimensionality.

2.3 Unpopular Learning Methods

2.3.1 Distance Based Learning Methods (DBL)

In this section, distance based methods are introduced. It is noted that most of these methods, except for the Euclidean Distance, are not common in text classification literature.

2.3.1.1 Euclidean Distance (ED)

The Euclidean Distance between the incoming pattern or document $\mathbf{x}$ and mean vector of the training data set is defined as

$$g_\omega(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_j)^T(\mathbf{x} - \mathbf{m}_j) = \|\mathbf{x} - \mathbf{m}_j\|^2, \tag{2.8}$$

where $\mathbf{x}$ is a feature vector of the incoming text, $\mathbf{m}_j$ is the mean vector of category ω_j. The learning process by this classifier needs to compute the mean vector of each class. In the expressions below the subscript j is omitted for the sake of simplicity.

2.3.1.2 Projection Distance (PD)

Projection distance function gives the distance from pattern $\mathbf{x}$ to the minimum mean square error in the hyperplane which approximates the distribution of the sample and is defined as

$$g(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}\|^2 - \sum_{i=1}^{k}\{(\mathbf{x} - \mathbf{m})^T\Phi_i\}^2, \tag{2.9}$$

where Φ_i denotes the i^{th} eigenvector of the covariance matrix for each category, k denotes the dimensionality of the hyperplane. In the case of two dimensional feature spaces, the decision boundary of projection distance becomes a pair of straight lines (i.e. asymptotic

lines of hyperbola as shown in Fig. 2.4) in which the hyperbola degenerates, and in case of 3-dimensional space or more the decision boundary is of the form of quadratic hypersurfaces. Training this classifier requires the computation of covariance matrices and the eigenvectors for each class.

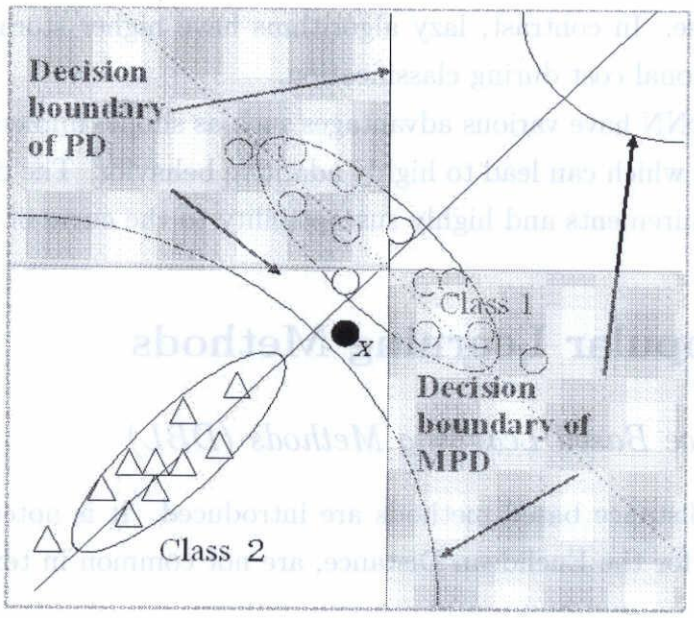


Figure 2.4: Decision boundaries of PD and MPD

2.3.1.3 Modified Projection Distance (MPD)

Classification error rates in and near the intersecting point of the hyperplane (i.e. shared subspace) may be high when projection distance is used. If expression (2.9) is modified to expression (2.10) this problem can be eliminated [31].

$$g(\mathbf{x}) = \|\mathbf{x}-\mathbf{m}\|^2 - \sum_{i=1}^{k} \frac{(1-\alpha)\lambda_i}{(1-\alpha)\lambda_i + \alpha\sigma^2}\{(\mathbf{x}-\mathbf{m})^T\Phi_i\}, \tag{2.10}$$

whereby λ_i is the i^{th} eigenvalue of the covariance matrix for each category, and σ^2 is the average of all eigenvalues of all categories. α represents a parameter of a value [0, 1]. In expression (2.10) when $\alpha = 0$ it is equivalent to the projection distance and when $\alpha = 1$, it is equal to the Euclidean Distance. Fig. 2.4 illustrates the decision boundary of modified projection distance. This figure shows that an incoming document $\mathbf{x}$ would be correctly classified by MPD as opposed to PD. Training this classifier requires the computation of the covariance matrices and the eigenvectors for each class.

2.3.2 Linear Discriminant Function (LDF)

The linear discriminant function's decision boundary is a straight line, a plane, a hyperplane for two, three, or higher dimensional spaces respectively. The linear discriminant function can be defined as

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0, \tag{2.11}$$

where $\mathbf{w}$ is the weight vector computed as

$$\mathbf{w} = -\Sigma_W^{-1}\mathbf{m}, \tag{2.12}$$

and w_0 is the bias or threshold weight expressed as

$$w_0 = \frac{1}{2}\mathbf{m}^T\Sigma_W^{-1}\mathbf{m}. \tag{2.13}$$

Σ_W denotes the pooled within covariance matrix (the mean covariance matrix for all categories). Expressions (2.12) and (2.13) have to be computed during the training of this classifier. The reader may note that this classifier assumes that Σ_W^{-1} exists. This is not necessarily true in practice.

2.3.3 Regularized Linear Discriminant Function (RLD)

The drawback of the LDF is that when the sample size is smaller than the dimensionality, the inverse matrix, Σ_W^{-1} does not exist. To overcome this problem a regularized linear discriminant function (RLD) can be used. It is also argued that RLD is for high-dimensional data to overcome performance degradation [92]. Since automatic text classification (ATC) involves higher dimensional space than sample size, it was appropriate to study RLD in ATC.

Instead of using Σ_W in equation (2.12), we regularize it to Σ_W^r which can be defined as

$$\Sigma_W^r = (1-\alpha)\Sigma_W + \alpha\frac{tr(\Sigma_W)}{n}I, \quad (0 \leq \alpha \leq 1), \tag{2.14}$$

where $tr(\Sigma_W)$ is the trace of Σ_W, I is the identity matrix and n is the dimensionality. The value of α can be determined by use of cross-validation techniques [30, 36]. RLD tends to be equivalent to the linear discriminant function and the Euclidean distance when $\alpha = 0$ and $\alpha = 1$ respectively.

2.3.4 Logistic Discrimination Classifier

The early studies on logistic discrimination can be found in [1, 21]. The author also applied the logistic discrimination classifier in text classification. We adopted the maximum likelihood estimation of parameters which uses iterative optimization. This uses

the likelihood function and its derivatives as described in [92].

Since the binary classification approach was adopted the two category case was taken into consideration. The basic assumption is that the difference between the logarithms of the conditional density function is linear in the variables $\mathbf{x}$ shown here as

$$\log\left(\frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)}\right) = \beta_0 + \boldsymbol{\beta}^T\mathbf{x}, \tag{2.15}$$

where β_0 and $\boldsymbol{\beta}$ are constant and the vector parameters, respectively. These parameters are estimated during training as described below.

From 2.15 it can be shown that the assumption is equivalent to

$$P(\omega_1|\mathbf{x}) = \frac{\exp(\acute{\beta}_0 + \boldsymbol{\beta}^T\mathbf{x})}{1+\exp(\acute{\beta}_0 + \boldsymbol{\beta}^T\mathbf{x})}, \quad P(\omega_2|\mathbf{x}) = \frac{1}{1+\exp(\acute{\beta}_0 + \boldsymbol{\beta}^T\mathbf{x})}, \tag{2.16}$$

where $\acute{\beta}_0 = \beta_0 + \log(P(\omega_1)/P(\omega_2))$. The maximum likelihood estimation can be used to estimate this model of logistic discrimination.

An iterative nonlinear optimization procedure can be used which involves likelihood function and its derivatives. The likelihood of observed documents $\mathbf{x}$ with two categories can be written as

$$L = \prod_{r=1}^{N_1} P(\mathbf{x}_{1r}|\omega_1) \prod_{r=1}^{N_2} P(\mathbf{x}_{2r}|\omega_2) \quad r = 1,\ldots,N_s; \quad s = 1,2, \tag{2.17}$$

where $\mathbf{x}_{sr}$ are the documents in category ω_s. Rewriting equation (2.17) we have

$$L = \prod_{r=1}^{N_1} P(\omega_1|\mathbf{x}_{1r})\frac{P(\mathbf{x}_{1r})}{P(\omega_1)} \prod_{r=1}^{N_2} P(\omega_2|\mathbf{x}_{2r})\frac{P(\mathbf{x}_{2r})}{P(\omega_2)} \tag{2.18}$$

$$= \frac{1}{P(\omega_1)^{N_1}P(\omega_2)^{N_2}} \prod_{\text{all } \mathbf{x}} P(\mathbf{x}) \prod_{r=1}^{N_1} P(\omega_1|\mathbf{x}_{1r}) \prod_{r=1}^{N_2} P(\omega_2|\mathbf{x}_{2r}). \tag{2.19}$$

Since the factor

$$\frac{1}{P(\omega_1)^{N_1}P(\omega_2)^{N_2}} \prod_{\text{all } \mathbf{x}} P(\mathbf{x})$$

is independent of the parameters of the model, it can be eliminated, under the assumption that we are free to choose $P(\mathbf{x})$ as in [22]. In this regard the likelihood L can be maximized by using

$$\acute{L} = \prod_{r=1}^{N_1} P(\omega_1|\mathbf{x}_{1r}) \prod_{r=1}^{N_2} P(\omega_2|\mathbf{x}_{2r}). \tag{2.20}$$

For simplifying the computation, we can introduce logarithms to this equation obtaining

$$\log(\acute{L}) = \sum_{r=1}^{N_1} \log(P(\omega_1|\mathbf{x}_{1r})) + \sum_{r=1}^{N_2} \log(P(\omega_2|\mathbf{x}_{2r})), \tag{2.21}$$

which can be written by using (2.16) assuming that

$$\log(\acute{(L)}) = \sum_{r=1}^{N_1} (\acute{\beta_0} + \boldsymbol{\beta}^T \mathbf{x}_{1r}) - \sum_{\text{all } \mathbf{x}} \log\{1 + \exp(\acute{\beta_0} + \boldsymbol{\beta}^T \mathbf{x})\}. \tag{2.22}$$

The gradient ascent algorithm was used to estimate parameters $\acute{\beta_0}$ and β. The gradient of (2.22) with respect to β_j is

$$\frac{\partial \log \acute{L}}{\partial \acute{\beta_0}} = N_1 - \sum_{\text{all } \mathbf{x}} P(\omega_1|\mathbf{x}) \tag{2.23}$$

$$\frac{\partial \log \acute{L}}{\partial \beta_j} = \sum_{r=1}^{N_1} (\mathbf{x}_{1r})_j - \sum_{\text{all } \mathbf{x}} P(\omega_1|\mathbf{x}) x_j, \quad j = 1, \ldots, n \tag{2.24}$$

Algorithm 2.1 Gradient Ascent Algorithm for One Category

1: **input:** document $\mathbf{x}$
2: **output:** optimized β_0 and $\boldsymbol{\beta}$
3: **initialize:** β_0, $\boldsymbol{\beta}$, threshold ϵ, learning rate r, $k = 0$
4: **repeat**
5: $k \leftarrow k + 1$
6: $\beta_0^{k+1} \leftarrow \beta_0^k + r\left(N_1 - \sum_{\text{all } \mathbf{x}} P(\omega_1|\mathbf{x})\right)$
7: **for** $j = 1$ to n **do**
8: $\beta_j^{k+1} \leftarrow \beta_j^k + r\left(\sum_{r=1}^{N_1} (\mathbf{x}_{1r})_j - \sum_{\text{all } \mathbf{x}} P(\omega_1|\mathbf{x}) x_j\right)$
9: **end for**
10: **until** $\left(\sqrt{\sum_{i=1}^{n} \beta_i^{k+1} - \beta_i^k}\right) < \epsilon$

In this regard algorithm 2.1 was particularly used in the iterative optimization process.

2.4 Proposed Improvements of Learning Methods

2.4.1 Normalized-weighted Metric (NWM) for kNN

We propose an improvement to the kNN learning method. The conventional kNN learning method has drawbacks. For example, it assumes that all examples in the subset $D_k \subseteq D$ have equal importance in predicting the class of the incoming document. Thus it results in giving equal weight even to those instances that are far from the incoming pattern.

Consequently, local-category information for correct prediction of a class can be distorted.

In an attempt to remove this drawback Lim [64] proposed a technique for weighting document similarities defined by

$$Z(\mathbf{x}, \omega_j) = \sum_{D_i \in D_k}^{k} sim(\mathbf{x}, D_i) y(D_i, \omega_j), \tag{2.25}$$

where $y(D_i, \omega_j) \in \{0, 1\}$ is a discrete function that refers to the classification of training document D_i belonging to a specific category such that $y(D_i, \omega_j) = 1$ for YES and $y(D_i, \omega_j) = 0$ for NO. The $sim(\mathbf{x}, D_i)$ is the similarity between the test document and the training document. This can be called the similarity weighted function (SWF). In general terms it can be called weighted metric function (WMF).

However, one can note that expression (2.25) can still result in noises and difficulties in determining the threshold. To solve these problems, we propose a normalized-weighted metric (NWM) function. The function can use a distance or similarity measure. Let our metric be similarity measure, $sim(.)$ as in (2.25). NWM can be defined as

$$Z'(\mathbf{x}, \omega_j) = \frac{Z(\mathbf{x}, \omega_j)}{\sum_{D_i \in D_k} sim(\mathbf{x}, D_i)}. \tag{2.26}$$

This can be understood as the normalization of (2.25). This was used instead of the conventional voting strategy. Expression (2.26) has a property such that $(0 \leq Z'(\mathbf{x}, \omega_j) \leq 1)$. In other words probabilistic value will always be obtained. In doing so, the threshold will always be in the range of 0 to 1.

In general terms, $sim(\mathbf{x}, D_i)$ can be replaced with other metrics such as distance metrics. For example Euclidean distance can be used instead of cosine similarity. However the reader should note that the cosine similarity function was used in the experiments.

2.4.2 A Posterior Probability by Distance Classifiers (PPD)

PPD is one of the proposed improvements for distance classifiers. It generally works better than the distance based learning methods. Distance based learning methods may not catch enough discriminatory information from training data. This can lead to errors in classification decisions of unlabeled data points. In order to solve this problem and enhance learning by these methods, we introduce a function which provides more information in a probabilistic approach.

Assume there are C classes in the text collection. Instead of using distances, $g_{\omega_j}(\mathbf{x})$ in the classification process, *a posteriori* probability, $P(\omega_j|\mathbf{x})$ of class ω_j can be computed as

$$P(\omega_j|\mathbf{x}) = \frac{P(\omega_j) exp(-\frac{1}{2} g_{\omega_j}(\mathbf{x}))}{\sum_{i=1}^{C} P(\omega_j) exp(-\frac{1}{2} g_{\omega_j}(\mathbf{x}))}, \tag{2.27}$$

where a priori probability $P(\omega_j)$ is defined in (2.1). $P(\omega_j|\mathbf{x})$ has a property that $0 \leq P(\omega_j|\mathbf{x}) \leq 1$. This simplifies the process of setting the threshold.

Training this classifier depends on the distance metric used. Section 2.3.1 describes various classifiers based on distance metrics. Using such distances involve training the classifier as described accordingly.

The general form of PPD can use other discriminant functions such as the linear discriminant function described in 2.3.2. Empirical results show that PPD is better than the simple use of distance metric in text categorization.

2.5 Performance Evaluation

Although TC is at the cross-road of information retrieval and machine learning, evaluating text classification is always done using measures from information retrieval. The popular ones are recall, precision, break-even point and the F_β-measure.

Other measures adopted from machine learning techniques have been reported in the literature. These include classification rates [10, 70, 62], accuracy and error rates [94]. The list is not exhaustive here. We discuss these measures in Section 2.5.1.

Note that the measures such as the recall, precision and F_β-measure for evaluation of classification effectiveness were mostly adopted. These measures are regarded as the standard evaluation methods for classification systems in automatic text classification.

It is also of interest to test the significance of the proposed methods in this work. We adopted the commonly used methods to test statistical significance in Section 2.5.2.

2.5.1 Measuring Classification Performance

Classification performance can be measured using effectiveness and efficiency criteria. Effectiveness is the most popular criterion in measuring the performance of classifiers. Most of the measures in this section will deal with effectiveness. Furthermore, the efficiency criterion is explained very briefly in Section 2.5.1.5.

Binary classification tasks are common in text classification (TC). These involve defining the class of interest or positive class and the negative class. To measure effectiveness, information retrieval (IR) metrics are usually used [57, 58, 82, 94]. Most of the measures for binary classification tasks involve defining a contingency table for classification decisions. Table 2.1 illustrates the idea of the contingency table. It is noted that the classification system is evaluated against a human expert's decisions.

When the decision of a human expert and that of the classification system (i.e., machine/classifier decision) is YES (i.e., the document truly belongs to class ω_j), the decision is called a *true positive* (TP). When the decision of the classifier is NO while human expert decision is YES (i.e., the machine falsely rejects the document in category

ω_j), the decision is called a *false negative (FN)* or an *error of omission.* When the classifier's decision is YES while the human expert's decision is NO (i.e., the machine falsely accepts the document classifying it under category ω_j), the decision is called a *false positive (FP)* or an *error of commission.* When both classifier and human expert's decisions are NO (i.e., the document truly does not belong to class ω_j), this judgment is referred to as a *true negative (TN)*.

Table 2.1: Contingency table for classification decisions

(a) Macro-average strategy

Category ω_j	Human Expert Decisions		
		YES	**NO**
Machine Decisions	**YES**	TP_j	FP_j
	NO	FN_j	TN_j

(b) Micro-average strategy

Category Set $\Omega = \{\omega_1, \cdots, \omega_C\}$	Human Expert Decisions		
		YES	**NO**
Machine Decisions	**YES**	$TP = \sum_{j=1}^{C} TP_j$	$FP = \sum_{j=1}^{C} FP_j$
	NO	$FN = \sum_{j=1}^{C} FN_j$	$TN = \sum_{j=1}^{C} TN_j$

2.5.1.1 Recall

Recall is one of the popular measures for text classification effectiveness. Recall can be regarded as the proportion of class members that the machine assigns to class ω_j.

In general, there are two methods of computing recall. These are the macro averaging and the micro averaging methods. These methods can give quite different scores depending on the generality of the data sets. Categories with few positive training examples are always emphasized by macro-averaging – since this approach gives equal weight to every category. Micro-averaging gives equal weight to every document, and it can be considered to be a per-document average strategy.

In the macro-averaging method recall R_j for each class is first computed as

$$R_j^M = \frac{TP_j}{TP_j + FN_j}. \tag{2.28}$$

The illustration is given in Table 2.1(a). The averaged recall R of all categories is computed as

$$R^M = \frac{\sum_{j=1}^{C} R_j}{C}. \tag{2.29}$$

The micro-averaging method requires the calculations of recall as

$$R^{\mu} = \frac{TP}{TP + FN} = \frac{\sum_{j=1}^{C} TP_j}{\sum_{j=1}^{C}(TP_j + FN_j)}. \tag{2.30}$$

This concept is summarized in Table 2.1(b). To enable comparisons with previous works in the literature we mostly adopted the micro-averaging and reported the F-measure scores. For complementing the results, macro-averaged results are also reported.

2.5.1.2 Precision

Precision is another popular measure for text classification effectiveness. We adopted precision for measuring classification effectiveness. This is one of the measures that are regarded as standard evaluation methods for classification systems in automatic text classification.

Precision can be regarded as the proportion of documents that the classification system assigns to a class that really belong to the class in question. As described in Section 2.5.1.1, macro- and micro-averaging methods were usually adopted. The macro-averaged precision needs to calculate the precision P_j^M for each category in the first place as

$$P_j^M = \frac{TP_j}{TP_j + FP_j}. \tag{2.31}$$

Then macro-averaging is done using the formula

$$P^M = \frac{\sum_{j=1}^{C} P_j}{C}. \tag{2.32}$$

The micro-averaging is similarly carried out as

$$P^{\mu} = \frac{TP}{TP + FP} = \frac{\sum_{j=1}^{C} TP_j}{\sum_{j=1}^{C}(TP_j + FP_j)}. \tag{2.33}$$

This concept is summarized in Table 2.1(b). To enable comparisons with previous works in the literature we mostly adopted micro-averaging and complimented the results with macro averaging.

Fig. 2.5 illustrates the relationship between recall and precision. It can be observed that recall decreases with an increase in precision and vice versa. This means that the use of recall only or precision only can not reflect the real behavior of the classification system. A trade-off should be sought by setting up the parameters appropriately.

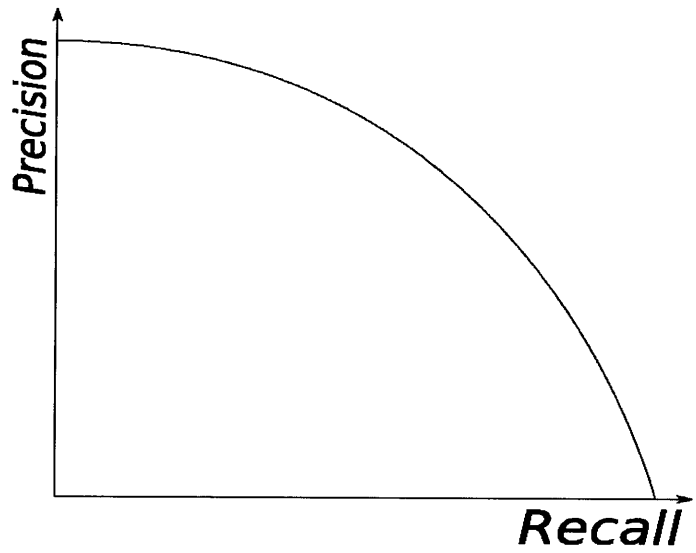


Figure 2.5: Recall versus precision

2.5.1.3 Break-even point

Break-even point (BEP) is one of the popular measure of TC effectiveness. It seeks to provide a single score from recall and precision. If recall and precision of a classifier can be tuned to have an equal value (i.e., $R = P$), then this is called the BEP of the classification system.

If the recall and precision values can not be made exactly equal, the average of the closest recall and precision scores is used as an *interpolation* of the BEP. The interpolation is always undesirable since it can give misleading systems' performance [94].

2.5.1.4 F_β-measure

The F_β-measure was first defined by C. J. van Rijsbergen [88]. It is a harmonic mean of recall and precision. The harmonic nature can easily be seen by considering the harmonic mean formula. The harmonic mean H of n numbers $p_1, \ldots, p_n$ can be defined as

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{p_i}}. \tag{2.34}$$

By plugging the recall R and precision P in (2.34), we have

$$H = \frac{2}{(\frac{1}{R} + \frac{1}{P})} = \frac{2RP}{R+P} = F_1. \tag{2.35}$$

This gives recall and precision equal weight. The generalized case of F_1 in (2.35) can be defined as

$$F_\beta(R, P) = \frac{(\beta^2 + 1)RP}{\beta^2 R + P}, \quad (0 \leq \beta \leq \infty), \tag{2.36}$$

where β is a parameter which allows weighting of precision and recall.

F_1 in (2.35) is the special form of (2.36) when $\beta = 1$. F_β is maximized when the values of recall and precision are equal or close. Otherwise the value of either one can dominate F_β. It can therefore be noted that the BEP in Section 2.5.1.3 is a specific variable of the F_β-measure. According to [94], the F_1 is the most suitable choice among the measures of TC effectiveness. However it is notable that, in general terms, a variety of scores to measure classification performance are desirable.

2.5.1.5 Other Performance Measures

- Classification Rate

  This measure of effectiveness is used when the binary classification tasks are not considered [70]. This is defined in terms of the number of documents which are correctly classified ζ and number of documents which are wrongly classified η. The formula is

  $$CR = \frac{\zeta}{\zeta + \eta} \times 100. \tag{2.37}$$

  Note that the contingency table is not used here. This measure is common in machine learning and pattern recognition fields. We used this measure in some of the experiments.

- Accuracy

  Accuracy is another measure of text classification effectiveness and it can be defined as

  $$A = \frac{TP + TN}{TP + FP + FN + TN}. \tag{2.38}$$

  This measure was adopted from machine learning literature, even though it is not widely used in TC literature. This is because it has a large denominator which leads to insensitivity to more variations in the number of correct decisions than with recall and precision [94].

- Error Rate

  The error rate takes care of both errors of commission and errors of omission [57] and it can be defined as

  $$E = \frac{FP + FN}{TP + FP + FN + TN} = 1 - A. \tag{2.39}$$

  This measure is closely considered in statistical analysis of improvements as will be indicated in Section 2.5.2. It usually takes care of text classification effectiveness.

- Efficiency

  The efficiency measure is very important in the case of applications that require time and speed considerations [26]. In our study we report how the instance based classifier like the kNN's efficiency was improved. The feature reduction methods proposed in chapter 4 improved the efficiency of the classifiers. Efficiency is also very important in selecting among classifiers with similar effectiveness.

2.5.2 Statistical Analysis of Improvements

Statistical significance testing gives an insight into any apparent improvement in the performance of algorithms or methods. This gives any researcher an insight for drawing conclusions about the performance of a proposed method. It is therefore desirable to perform statistical analysis to show whether proposed methods really have an impact on the performance of text categorization or not. Testing for significance of improvements can be done using the following general steps:

- Formulate the null hypothesis (H_0, which assumes that "no difference in ability of methods") and the alternative hypothesis (H_A – the opposite statement of H_0).

- Decide on the value of significance level α. This can be considered as the fixed probability of wrongly rejecting H_0 when it is in fact true. It is the probability of a type I error set by the researcher. Commonly used values include 0.05 and 0.01.

- Calculate the critical value of the test statistic by using the classification score that shows the error rate of both methods (e.g. see Table 2.2). This depends on the statistical test in use. Examples of statistical tests are described below.

- Compare the test statistic with p-value obtained either from a table or computed based on the normal distributions' assumption, or compare with a critical value from a statistical table

- Draw conclusions.

  – If the calculated test statistic is higher than the critical value from the table (i.e., if the p-value is less than α), then reject H_0. This implies that, the probability that the difference happened by chance is small. In other words the improvement is statistically significant.

  – If the calculated test statistic is lower than the the critical value from the statistical table (i.e., if the p-value is greater than α, then accept H_0). The implication is that the probability that the difference happened by chance is high. In other words the probability that the improvements happened by chance is high. In this case there is no statistical evidence of improvement.

There are various statistical tests used in the literature. These include McNemar's test, Z-test of comparing two proportions and the χ^2 test. We describe these statistical tests below.

2.5.2.1 McNemar's Test

In statistics, McNermar's test was first introduced by Q. McNermar in 1947 [69]. In the field of machine learning and other related fields, McNemar's test has been recommended to be a powerful statistical test [23, 34]. It is powerful in the sense that it has a low probability of making a type I error. Indeed various authors on text categorization have applied this statistical test successfully [87, 91].

Table 2.2: 2x2 Contingency Table for two methods' performances

		Method δ_2	
		Correct	Wrong
Method δ_1	Correct	$\hat{a}$	$\hat{b}$
	Wrong	$\hat{c}$	$\hat{d}$

McNermar's test is a statistical analysis tool that can validate the significance of the differences between two methods. Let δ_1 and δ_2 be the first and second method, respectively. The number of documents which are correctly and wrongly classified can be defined using a contingency table as illustrated in Table 2.2. In this table the number of data that are correctly classified by both methods is represented by $\hat{a}$. The number of data that are wrongly classified by both methods is represented by $\hat{d}$. The number of data that are incorrectly classified by first method δ_1 is denoted by $\hat{c}$. The number of data that are wrongly classified by the second method δ_2 is denoted by $\hat{b}$.

Testing the statistical significance can be done under a null hypothesis (H_0) that the two methods δ_1 and δ_2, would have the same error rate E, which is reflected by $\hat{b}$ and $\hat{c}$ of Table 2.2. The alternative hypothesis (H_A) is that the two methods have different error rates. The chi-squared statistic can be approximately obtained using

$$\hat{\chi}^2 = \frac{(|b-c|-1)^2}{b+c}, \tag{2.40}$$

where minus one in the numerator caters for the Yates correction of continuity [98, 99]. This approximation is carried out due to the fact that McNemar's test is based on a χ^2 test for goodness-of-fit that compares the distributions of counts expected under the null hypothesis [23]. If the test statistic $\hat{\chi}^2$ is greater than $\chi^2_{1,0.95} = 3.841459$, there is a probability that the difference in performance by the two methods is less than the significance level $\alpha = 0.05$. Therefore we reject H_0 in favor of H_A. In other words, given the fact that the degree of freedom is 1, if the statistical value $\hat{\chi}^2 > 3.84149$, it would

give a p-value that is less than the significance level $\alpha = 0.05$. This suggests that the there is statistical evidence that the two methods in question perform differently.

Note that various values for significance level α can be used as is usual in statistical analysis procedures. For example the popular values include $\alpha = 0.01$, $\alpha = 0.001$ or even smaller values. In these cases, standard statistical tables or software can be used to obtain the p-value that is always compared in order to make the decision to reject or accept H_0. The smaller the p-value in comparison with the α significance level, the more the significance of the difference among the methods under the statistical analysis.

2.5.2.2 Comparing Two Proportions by Z-Test

Another statistical test is that of comparing two proportions [23] to find out whether there is a difference among methods δ_1 and δ_2. This is based on the comparison of the error rates of methods δ_1 and δ_2. Using Table 2.2, let

$$p_1 = \frac{(\hat{c} + \hat{d})}{N} \tag{2.41}$$

be the proportion of the test instances that are misclassified by method δ_1, and

$$p_2 = \frac{(\hat{b} + \hat{d})}{N} \tag{2.42}$$

be the proportion of the test instances that are misclassified by method δ_2. This statistical test assumes that when method δ_1 classifies an instance $\mathbf{x}_k$ from test set T, the probability of misclassification is p_1. Therefore the number of misclassification of N test instances is a binomial random variable with a mean Np_1 and the variance $p_1(1 - p_1)N$.

Another assumption is that given a good representation of N, the binomial distribution can sufficiently be approximated by a normal distribution. The difference between two *independent* normally distributed random variables result in normally distributed variables [23]. Therefore the quantity of $p_1 - p_2$ is normally distributed by assuming that the measured error rates p_1 and p_2 are independent. If we let $\bar{p} = (p_1 + p_2)/2$ be the average of the two error probabilities, then we have a mean of zero and a standard error defined as

$$SE = \sqrt{\frac{2p(1 - \bar{p})}{N}}. \tag{2.43}$$

Therefore the test statistic Z can be given as

$$Z = \frac{|p_1 - p_2|}{SE}, \tag{2.44}$$

which has approximately a standard normal distribution. It is known that the critical value $\alpha(-1.96 \leq Z \leq 1.96) = 0.05$ [91, 99]. The null hypothesis can be rejected if

$Z > 1.96$ or $Z < 1.96$. This is true if the significance level $\alpha = 0.05$ for the two-sided statistical test. A similar approach using various values of significance level α can be followed as discussed in Section 2.5.2.1.

This method of comparing two proportions in machine learning has been criticized [23] due to the following problems. First, the assumption of independence of p_1 and p_2 is violated because methods δ_1 and δ_2 are always measured using the same test set T. Second, the internal variation of methods δ_1 and δ_2 are not measured.

The lack of independence can be corrected by changing the estimation of the standard error and the resulting Z' statistic can be calculated as

$$Z' = \frac{|\hat{b} - \hat{c}| - 1}{\sqrt{\hat{b} + \hat{c}}}. \tag{2.45}$$

The test statistic in (2.45) can be seen to be the square root of the χ^2 statistic in McNemar's test.

2.5.2.3 The Binomial Comparative Trial Using the χ^2 Test

The test statistics described in Sections 2.5.2.1 and 2.5.2.2 have been implemented successfully in machine learning as seen in the literature (see respective sections for details). The χ^2 test also can be a good method to use in testing the significance of improvements. It was therefore adopted in the current study and described below.

Table 2.3: The 2x2 contingency table for binomial comparative Trial

	Method δ_1	Method δ_2	Total
Correct	$\hat{a}$	$\hat{b}$	$\hat{e}$
Wrong	$\hat{c}$	$\hat{d}$	$\hat{f}$
Total	$\hat{g}$	$\hat{h}$	$\hat{N}$

Using Table 2.3 we can redefine the two proportions defined in expressions (2.41) and (2.42) as follows:

$$\acute{p}_1 = \frac{\hat{a}}{\hat{g}}, \tag{2.46}$$

and

$$\acute{p}_2 = \frac{\hat{b}}{\hat{h}}. \tag{2.47}$$

The assumption with this method is that if the columns or the rows of a contingency table represent random samples from independent populations, then the null hypothesis can be phrased as a comparison of proportions such that $H_0 : \acute{p}_1 = \acute{p}_2$. Another assumption of these test statistics is that the proportions follow the chi-square goodness of fit property. Therefore given a good representation of $\hat{N}$, the binomial distribution can be sufficiently

approximated by a normal distribution. χ^2 test is recommendable when the sample size is considerably large. Otherwise the Fisher exact test[‡] is recommended.

This is a two-tailed null hypothesis [99] which can be tested using the statistic

$$\chi_I^2 = \frac{\hat{N}(|\hat{a}\hat{d} - \hat{b}\hat{c}| - \frac{\hat{N}}{2})^2}{\hat{e}\hat{f}\hat{g}\hat{h}}. \tag{2.48}$$

One can find that the degree of freedom $df = 1$ with the significance level $\alpha = 0.05$, the critical value of the statistic $\chi^2_{0.05,1} = 3.841$. If the test statistic $\chi_I^2 > 3.841$ at this significance level, then H_0 can be rejected in favor of $H_A : \acute{p}_1 \neq \acute{p}_2$.

This test also faces similar discrepancy as seen in Section 2.5.2.2, because it assumes that the proportions $\acute{p}_1$ and $\acute{p}_2$ are independent while they are actually obtained from methods δ_1 and δ_2 measured on the same test set T.

2.6 Experiments to Evaluate PPD and NWM

The experiments in this section focused on demonstrating the effectiveness of the proposed learning methods. Therefore a lot of details of the experimental setup are given in proceeding chapters.

2.6.1 Experimental setup

Experiments were conducted to evaluate the effectiveness of the proposed PPD and NWM. Since the focus is on these learning methods we follow the experimental setup which is described in 3.5 of Chapter 3. The PCA+CDA algorithm was also used in extracting the features. Further details of this algorithm can be found in Chapter 4.

2.6.2 Data for experiments

The ModApte Split of Reuters data set was used in the experiments. The details of these data can be found in 3.5.1.1.

2.6.3 Empirical Results

2.6.3.1 PPD Results

Table 2.4 summarizes the results of PPD. In this table results from various features are presented. These include absolute term frequency (AF), relative term frequency (RF), and power transformed relative frequency (RFPT). Other features include the use of term frequency weighted by document frequency (TFIDF) and TFIDF with the power

[‡]Fisher exact test is preferable when the sample size is small such as below 30 [99]

transformation (TFIDF+PT). The definitions of all these features are found in chapter 3.

Table 2.4: Micro-averaged F_1 measure (%) of the proposed PPD versus Euclidean Distance (ED). 115 categories of ModApte Split.

	Type Features Used					
Classifier	AF	AFPT	RF	RFPT	TFIDF	TFIDF+PT
PPD	75.3	76.8	78.2	80.5	72.6	74.2
ED	32.8	36.6	46.8	64.4	64.7	63.6

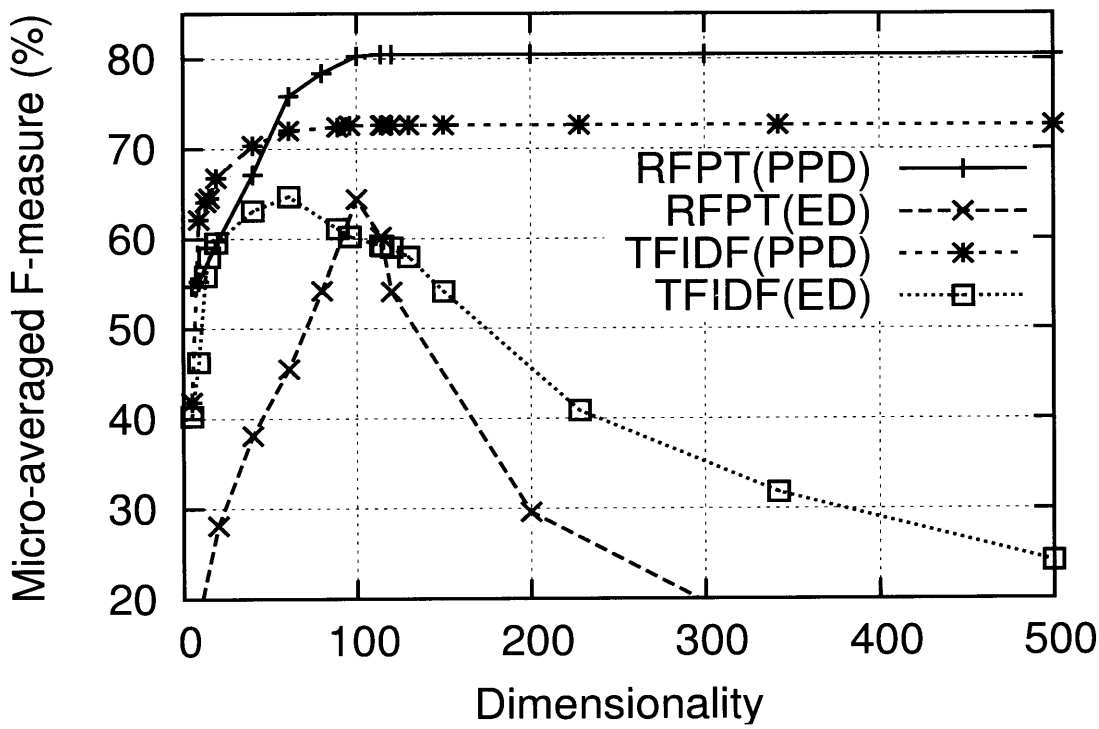


Figure 2.6: PPD in comparison with ED

Note that PPD in this case used the Euclidean distance (ED) to compute the *posteriori* probability. As it was pointed out earlier any distance based classifier can be used. Since the ED is efficient, it was chosen to illustrate the effectiveness of PPD.

With a simple perusal of Table 2.4, it can be seen that the PPD outperformed the Euclidean distance. The improvements range from about 20% to 40% of the averaged F_1-measure. Given the fact that the implementation was efficient these can be considered to be significant improvements.

Fig. 2.6 presents the relationship between dimensionality and micro-averaged F_1-measure. The significant improvements in all features used can be easily seen. RFPT leads to the highest performance of learning methods.

Results for the use of various popular learning methods in [41] were outperformed by the PPD method. The outperformed classifiers included the multinomial Naive Bayes

(MNB), Rochio and C4.5 algorithms. This is particularly true when the comparison is done using RFPT (i.e., in Table 2.4, PPD F_1 =80.5%, while in [41]: MNB's break even point (BEP) = 72.3%; Rochio BEP = 79.9% and C4.5 BEP = 79.4%). The encouraging thing is that the improvement is obtained with less computational costs. Note that the BEP is a special variable of F_1-measure as discussed in 2.5.1.4.

2.6.3.2 NWM results

Figure 2.7 illustrates the effect of NWM. It presents empirical results using various features which have been mentioned in Section 2.6.3.1. It is observable that NWM improves the performance of kNN learning methods.

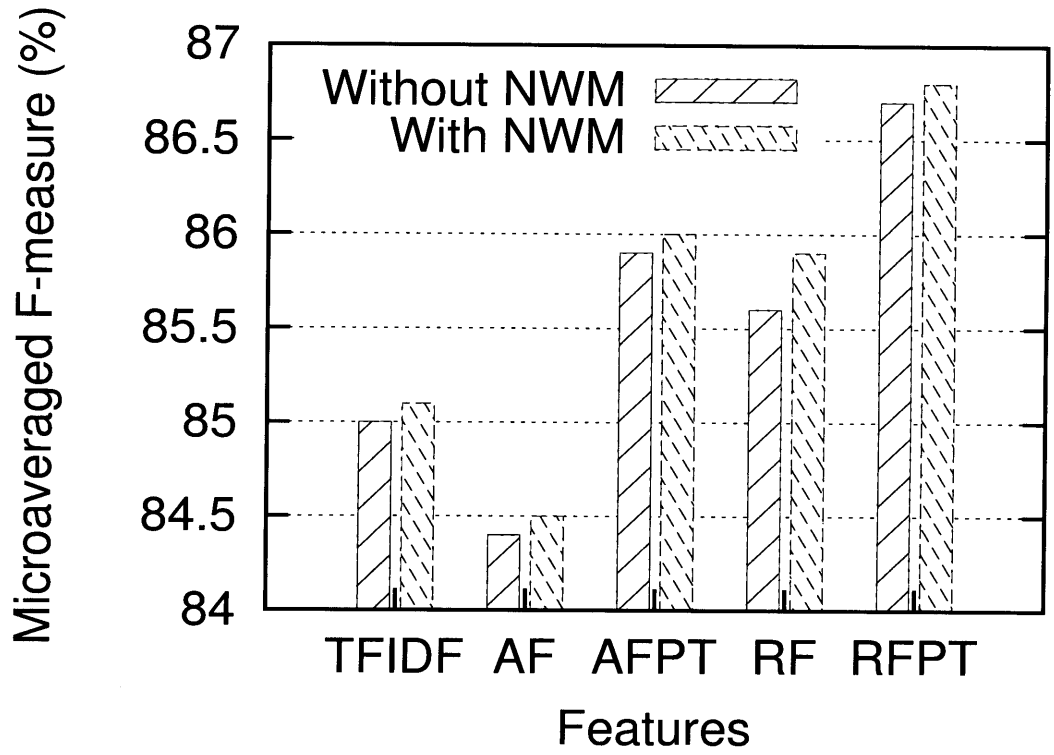


Figure 2.7: The effect of normalized-weighted metric (NWM)

The improvements are observed in all types of features used in the classification process. These results shows the weakness in the classical kNN of equal weighting of neighbors. It reveals that the use of the NWM method mitigates this weakness and improves the performance of kNN. These improvements are achieved without introducing complications in determining the threshold. This is because the value of the NWM method is always in the range of [0, 1].

The essence of improving the kNN learning method comes due to the fact that it is one of the best performers and it is simpler to implement in comparison with other learning methods such as SVMs. It is also flexible in handling multi-label data as is common in

automated text classification. Details of the advantages and flexibilities can be found in 2.2.4.

2.7 A Summary of the Learning Methods

In this chapter we described the machine learning methods. We first described selected popular learning methods in Section 2.2 to give an idea on the state of the art in the literature of TC. We further introduced the unpopular learning methods in Section 2.3. In particular we experimentally studied the learning methods presented in Sections 2.3.1.1; 2.3.1.2; 2.3.1.3; 2.3.2 and 2.3.3. Methods to improve the learning methods in this study are given in Section 2.4.2 and 2.4.1.

Although we give more details on feature transformation, reduction and integration in the following chapters, we presented the experimental results using the learning methods proposed in this Chapter. In every chapter it is indicated whenever a learning method is used accordingly. It will become clear that the proposed learning methods improve the classification performance in the TC.

Moreover we have seen various methods of performance evaluation described in Section 2.5. We used most of these methods given in Section 2.5.1 to evaluate the classification effectiveness. Then we carried out the statistical evaluation described in Section 2.5.2, which gave insight in drawing conclusions about the improvements.

Chapter 3

Document Representation

Documents are composed of natural language idioms. In order for a machine learning system to recognize a document there should be a way of representing it. This is usually done by the use of feature vectors. The vector elements are known as features. In the context of documents they are the words originating from the documents. They are known as features because not all words are used to represent a document.

This chapter starts by reviewing the state of the art on document representation for machine learning in text classification. Then it proposes a feature transformation that form a better document representation in machine learning. It mainly proposes the use of relative frequency with power transformation (RFPT). Comparative studies show that RFPT is better than the conventional methods for document representation.

3.1 Conventional Features

The vector model representation of textual data is common in automated text classification (ATC). Classically, the components of the feature vectors are the term frequencies weighted by inverse document frequency (TFIDF). This technique has been borrowed from information retrieval (IR) [82]. A recent theoretical analysis has shown that TFIDF is well suited for information retrieval problems. This is because it was actually developed with the idea of ranking documents in terms of relevancy to simplify the retrieval process [43, 77]. Although TFIDF has been shown to work well in IR, it might not be the best choice for text classification problems. For the convenience of the reader we review the theoretical analysis briefly.

In 1972 Jone Spark [43] proposed the term weighting technique in IR which later came to be known as inverse document frequency (IDF). He was working specifically with information retrieval problems for indexing documents. The theory has shown that the combination of TF (term frequency) and IDF is basically suited for IR. Without going into the details of IR-style, suppose we represent the documents as a feature vector

$\mathbf{x} = [x_1, \ldots, x_n]^T$, where n is dimensionality (lexicon size or vocabulary size), x_i is the frequency value of i^{th} word which are also known as term frequency (TF), and T refers to the transpose of a vector. Then the term weighting is done as

$$w_i = x_i * log\frac{N}{N_i}, \tag{3.1}$$

where N refers to the total number of documents in the collection and N_i is the *document frequency* which is the number of documents in which term i occurs. In other words the *log* part of the equation (3.1) denotes the *inverse document frequency* (IDF). The intuition here was that a *query term* which occurs in many documents is not a good discriminator for retrieving desired documents. Therefore it should be given less weight than the one which occurs in few documents [43, 77]. In order to avoid text length variation within documents, a normalization to vector unit length is classically carried out using

$$\acute{w}_i = \frac{w_i}{\sqrt{\sum_{j=1}^{n}(w_j)^2}}, \tag{3.2}$$

which is also called cosine normalization.

The IDF part of equation (3.1) consists of the logarithm of inverted probability. We can therefore estimate the probability that a random document d can consist of the term t_i. This probability can be computed as

$$P(t_i) = P(t_i \text{ occurs in } d) \approx \frac{N_i}{N}. \tag{3.3}$$

We can therefore redefine the IDF in terms of probability as

$$idf(t_i) = -logP(t_i), \{ \text{ by recalling from } -logx = log\frac{1}{x}\}. \tag{3.4}$$

In terms of IR, the document scoring functions are assumed to be *additive*. In this context, if two query terms are denoted as t_1 and t_2, and the weights for each term are w_1 and w_2. Therefore a document containing both terms would score $w_1 + w_2$. Let us use the symbol $\wedge$ to denote logical operator 'and'. Assuming that the occurrences of different terms in documents are statistically independent then the *idf* can be computed as

$$\begin{aligned} idf(t_1 \wedge t_2) &= -logP(t_1 \wedge t_2) && (3.5)\\ &= -log\left(P(t_1)P(t_2)\right) && (3.6)\\ &= -(logP(t_1) + logP(t_2)) && (3.7)\\ &= idf(t_1) + idf(t_2) && (3.8) \end{aligned}$$

From this point of view and from the probabilistic model of IR in general, we can say that TFIDF can be seen to fit well with IR. A detailed theoretical analysis can be found in [77]. The work in [77] also explains how document relevancy ranking in IR relates to TFIDF which consists of IDF, consequently working well with IR problems.

Contrary to the use of TFIDF in ATC, we propose transforming relative frequency using power transformation (RFPT) in text classification problems. RFPT simultaneously takes care of both document length and sample distribution problems. We performed extensive experiments to verify the effectiveness of RFPT. We find RFPT to be empirically superior to TFIDF. In section 3.3 we shall see how RFPT is suited to text classification problems.

3.2 Related Works

In this section we present some relevant works that may relate to ours. Where similarities with our work are found, we briefly describe the differences.

First of all it is worth mentioning that there are some works that use the unit length normalization and power law concepts [76]. However, it may be noted that the approach in [76] and ours are different. First, they used weighted vectors commonly called TFIDF while we used relative term frequency and its transformation. Second the unit length normalization they used is common in the literature and is different from ours.

Rennie et al. [76] furthermore used the concept of power law distribution on weighted vectors - based on choosing a value of a parameter d, which is added to weighted vector components and then the result is transformed by computing its *log* value. In other words, their transformation is a special version of the log transformation described in [32]. In contrast we use power transformation (PT) on relative term frequency (RFPT) with which we observe higher classification performance. In addition we apply PT to TFIDF. Similarly we observe higher performance than by simply using TFIDF.

Unlike in previous research we used transformed features. Particularly we improved classification effectiveness by the use of relative term frequency with power transformation (RFPT).

3.3 Feature Transformation

In this Section the author describes the proposed feature transformation in text classification. It will become clear that the proposed way of document representation improves the learning ability of classifiers. In this research, feature transformation (FT) refers to transforming absolute term frequency (AF) to relative term frequency (RF) and power transformation (PT). We discuss these techniques in this subsection.

Let us consider a set of N sample texts, $\chi = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ with n–dimensional text space. Let us assume that every textual document belongs to one of the C classes $\{\omega_1, \omega_2, \ldots, \omega_C\}$. Each text can be represented as a feature vector, $\mathbf{x}_k = [x_1 x_2 \ldots x_n]^T$, whereby, n is dimensionality (lexicon size), x_i is the frequency value of i^{th} word and T refers to the transpose of a vector.

3.3.1 Relative Term Frequency(RF)

It is clear that the feature vectors generated in this way are absolute term frequencies. We transform absolute frequency to relative term frequency (RF) to solve the problem of dependency on text length as follows:

$$y_i = \frac{x_i}{\sum_{j=1}^{n} x_j}. \tag{3.9}$$

This follows a property so that

$$\sum_{i=1}^{n} y_i = 1; \quad (0 < y_i < 1). \tag{3.10}$$

The length variation can now be smaller compared to absolute term frequency. Therefore the problem of dependency on text length is solved. By plotting y_i for all i this leads to frequency distribution of words in a document. In other words RF can be regarded as a probability $P(t_i) = y_i$ of an event t_i in document $\mathbf{x}$.

3.3.2 Power Transformation (PT)

After obtaining the RF, the sample distribution for the documents may still be skewed. This is undesirable particularly for parametric classifiers such as linear or quadratic classifiers which are typically designed for Gaussian distributions.

Therefore, with the purpose of obtaining Gaussian-like distribution, power transformation can be performed using equation (3.11).

$$z_i = y_i^v, \qquad (0 < v < 1\ ). \tag{3.11}$$

This transformation makes the sample distribution of the frequency $y_i \geq 0$ to be Gaussian-like. Note that when RF is transformed using equation (3.11) the resulting features can be abbreviated to RFPT and they have superior properties to simplify the process of learning by a classification system. One can choose to write equations (3.9)

and (3.11) concisely as

$$z_i = \left(\frac{x_i}{\sum_{j=1}^{n} x_j} \right)^v, \quad (0 < v < 1\ ), \tag{3.12}$$

which represents RFPT explicitly.

The process of modeling texts using these transformations is illustrated in Fig. 3.1. In this figure the distribution for absolute term frequency is right-skewed (positive skewness) and document length within-class may vary considerably. This depicts many cases of real world problems. When absolute term frequency is transformed into relative frequency (RF), the lengths of documents are normalized and we can observe the fact that the length generally gives no information about the category. This reduces the learning load of classifiers and improves classification accuracy.

When power transformation (PT) is applied to relative frequency, the shape of the distribution becomes Gaussian-like. Based on the optimality of the linear or quadratic classifiers to the Gaussian distributions, this kind of transformation can be advantageous to text classification systems. Gaussian-like distributions lead to an optimal decision boundary.

The proportion of the frequency of the variables that lie at the center of a sample distribution is important in defining the shape of a given distribution. Therefore the shape of the sample distribution can be conveniently discussed in terms of skewness and kurtosis. The discussion on how to obtain a desirable sample distribution can be done with reference to Gaussian distribution as a standard. Let $\bar{z}$ and σ^2 denote the mean and the variance of z_i, respectively. The measure of skewness is based on the third moment about the mean, $E\{(z_i - \bar{z})^3\}$. The index of skewness is a unit measure defined by the ratio

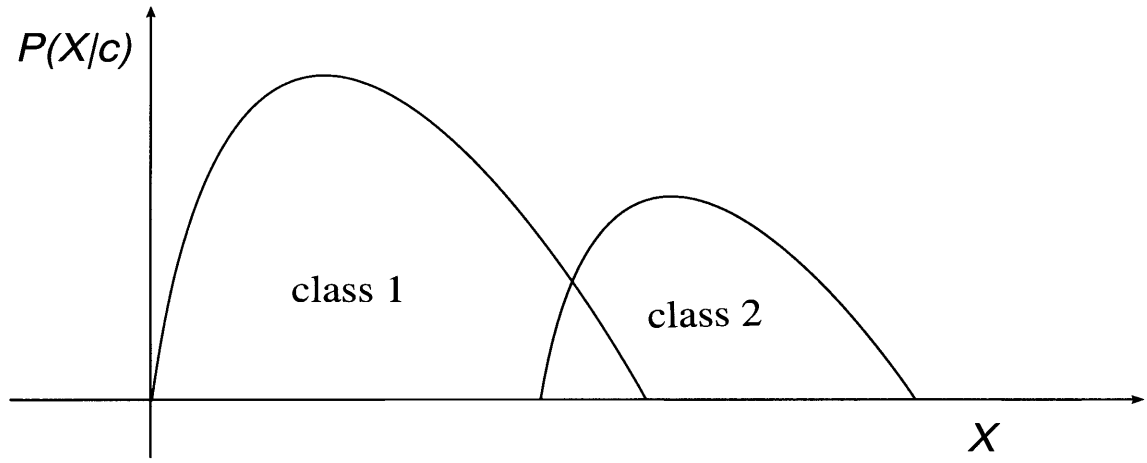
$$\gamma = E\{\left(\frac{z_i - \bar{z}}{\sigma} \right)^3\} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{z_i - \bar{z}}{\sigma} \right)^3. \tag{3.13}$$

Large negative values of γ indicate negative skewness (left skewness). Large positive values show right skewness (positive skewness). For a normal distribution the value of $\gamma = 0$ in (3.13). Therefore, the aim of (3.11) is to get a Gaussian-like distribution with a value of γ that is close to 0. Therefore variables z_i tend to be Gaussian-like when the values of γ tend to be close to 0.

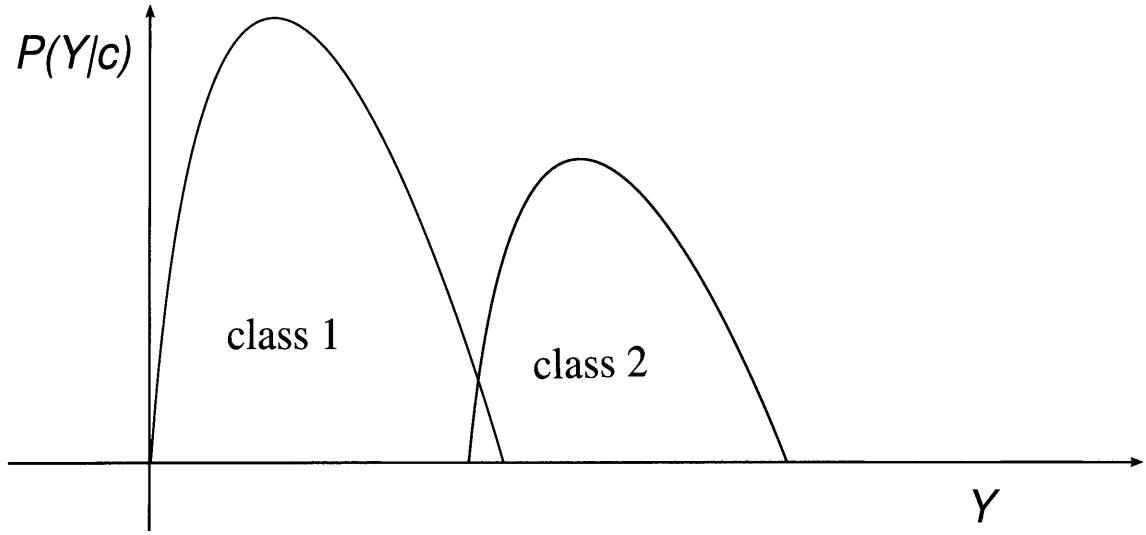
Kurtosis is measured around the 4^th central moment of a distribution. It is also known in [42] that the kurtosis

$$\kappa = E\{\left(\frac{z_i - \bar{z}}{\sigma} \right)^4\} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{z_i - \bar{z}}{\sigma} \right)^4 \tag{3.14}$$

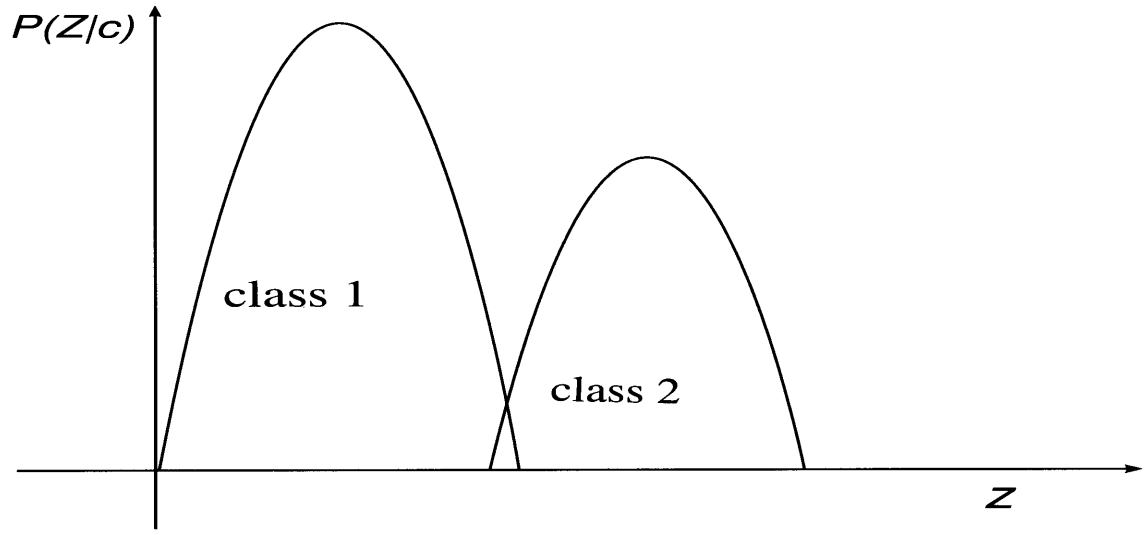
of the Gaussian distribution is 3. Positive kurtosis is always accompanied by a property



(a) Absolute term frequency(AF)



(b) Relative frequency (RF)



(c) RF with power transformation (RFPT)

Figure 3.1: The effect of feature transformation on non-Gaussian sample distribution.

of *peakedness* in a distribution. Peakedness usually suggests a relatively long or fat tail. Positive kurtosis can also be called *leptokurtosis*. Negative kurtosis is sometimes referred to as *platykurtosis*. It shows a presence of a relatively short or thin tail. Fukunaga [32] shows that the kurtosis of variables z_i tends to be 3 when the variables y_i are causal (i.e. positive) and are transformed using equation (3.11). Therefore variables z_i tend to be Gaussian-like.

It is also worth noting that the length of RFPT is normalized to 1 when $v = 0.5$ as follows:

$$\sum_{i=1}^{n} z_i^2 = \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \left(\frac{x_i}{\sum_{j=1}^{n} x_j} \right) = 1. \tag{3.15}$$

In other words RFPT satisfies the normality property. Indeed we find higher classification performance with RFPT when $v = 0.5$ (see section 3.5.5 for the empirical results).

3.4 Experiments Using Small Samples

3.4.1 Data for experiments – randomly selected

For an effective evaluation of feature transformation and classification techniques, a set of text data for category assignment was required. A benchmark collection of text categorization research called Reuters-21578 was used. This collection has been widely employed by other researchers, too [82], [49], [96], [83], [76], [100]. Reuters-21578 is composed of 21,578 articles manually classified into 135 categories.

A total of 1500 articles, *randomly* selected from 10 categories (i.e. acq, crude, earn, grain, interest, money-fx, money-supply, ship, sugar, trade) that is 150 articles per category were used. In order to retain the independence of the data used and to promote the validity of results, 150 articles in each category were divided into three groups of 50 articles each. One of the groups became the evaluation data and the remaining two groups were used as training data. Thus, by alternating the groups the same data could be used 3 times (two times for training, one time for evaluation). The average values of the experimental results were used to evaluate classification techniques.

After extracting or selecting data for experiments, the procedure for automatic text classification could be divided into four general steps. The steps include Feature vector generation, Dimension reduction, Learning or classifier training based on discriminant functions and Classification. The following subsections describe each of the steps.

3.4.2 Term Selection in Generating Vocabulary List

Generally speaking, not all generated words during feature vector generation have a significant contribution to discriminately represent a text. In general, function words

are not useful to represent document features discriminately. Similarly all content words are not necessarily helpful to represent document features. Therefore, before generating feature vectors, functional words and general words were removed with reference to a stop list prepared beforehand to reduce the features, amount of storage and processing time required by a classification system. In doing so over-fitting is also reduced.

In this work, the stop list of 572 words which is used by the typical retrieval system - SMART [49] for retrieving English documents was used. Even when a stop list was used to remove needless words, a lot of words still remained. Hence, words which appeared twice or less in all training data were removed to reduce further the remaining words.

3.4.3 Dimension Reduction by Principal Component Analysis

Principal Component Analysis (PCA) was used to further reduce the dimension of the generated feature vectors. PCA is based on the idea of performing an orthonormal transformation called the Karhunen-Loéve transformation, retaining only significant components. Orthonormal transformation is a linear transformation which allows the derivation of uncorrelated features from a set of correlated features [24, 32]. Details and results reflecting dimensionality reduction are reported in Chapter 4.

3.4.4 Learning Methods Used with Small Samples

Various discriminant functions were used in the classification experiments which included Euclidean distance, projection distance, modified projection distance, linear discriminant function, regularized linear discriminant function and support vector machines. We give brief explanations for each classifier in Chapter 3. Readers who need details are encouraged to consult the indicated references.

The Euclidean distance classifier is trained by computing the mean vector of each class. Training projection and modified projection distance classifiers involve computing covariance matrices and eigenvectors for each class [31].

The Support Vector Machine is one of the linear classifiers that use linear boundaries with margins such that the data from two categories are separated by hyperplanes with the largest margins. In other words, a support vector machine consists of finding the optimal hyperplane which is the one with maximum distance from the nearest training data. SVM can be extended to nonlinear functions when combined with kernel functions. Details for SVM are found in [20]. In the classification experiments the linear SVM function (SVM-Linear) and the radial basis function (SVM-RBF), namely the C-Support Vector Classification were used. Particularly we used the Library for Support Vector Machines (LIBSVM) version 2.33 that was published by Chang and Lin [15].

3.4.5 Empirical Results of Randomly Selected Samples

For the randomly selected texts of 10 category case, the results from the experiments show that:- (1) the regularized linear discriminant function (RLD) classifier outperformed all other classifiers by achieving the best classification rate (92.5%) with minimal computational cost. Table 3.1 shows that the RLD exhibited best classification rates in every kind of feature vector used i.e. absolute frequency, relative frequency and power transformed features. (2) As shown in figure 3.2, it is evident that the Linear Discriminant Function achieved lower results than the regularized linear discriminant function (RLD) for every feature vector used.

Table 3.1: The summary of best classification rates in %. Randomly selected texts of 10 categories

Classifiers	AF	AFPT	RF	RFPT
Euclidean Distance	68.9	83.2	78.7	86
Linear Discriminant	85.1	89.6	90.9	91.1
Regularized Linear Discriminant	88.2	90.9	91.4	92.5
Projection Distance	85.9	90	87.7	90.7
Modified PD	87.6	90.7	87.8	90.7
Linear SVM	86.4	87.7	90.7	90.3
RBF SVM	86.7	89.5	90.5	90.7

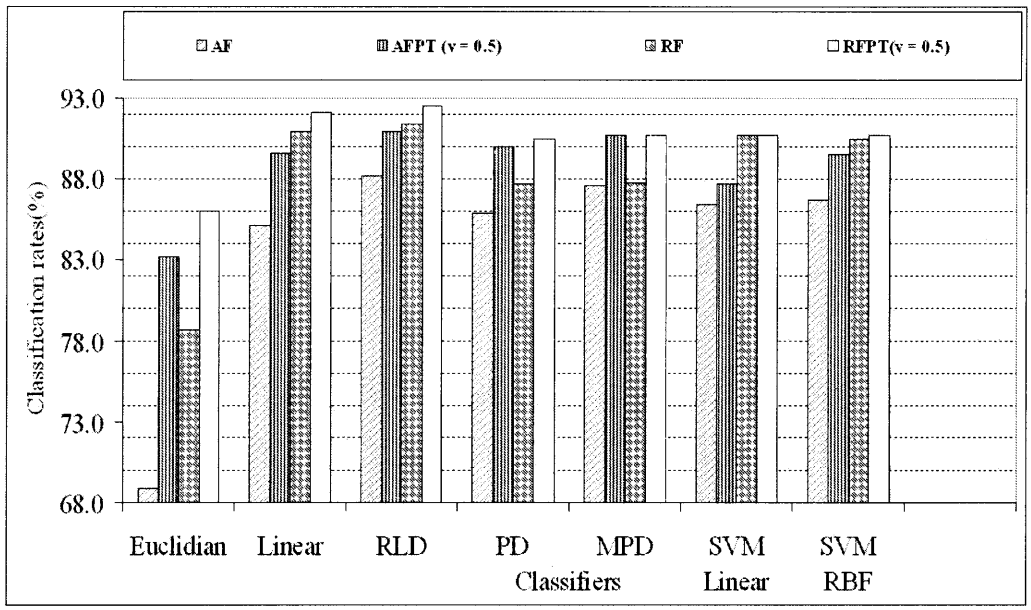


Figure 3.2: Learning methods and classification rates of various features. 10 category set.

(3) As shown in table 3.1 and in figure 3.2, the classification rate was significantly improved by employing the relative frequency instead of the absolute frequency. For instance the accuracy of the Euclidean distance classifier was improved by 9.8% and

that of the linear discriminant was improved by 5.8%. (4) As figure 3.2 shows, power transformation further improved the classification accuracy of each classifier used. The accuracy of the Euclidean distance classifier was improved cumulatively by 17.1% and that of the linear discriminant was similarly improved by 7%. Power transformed features from relative frequency improved classification accuracy such that all classifiers except the Euclidean distance exhibited over 90% accuracy.

3.4.6 Summary of the Small Samples Results

This Section presents a performance evaluation of techniques for feature transformation and classification to improve the accuracy of automatic text classification. The normalization to the relative word frequency, principal component analysis (K-L transformation) and power transformation were applied to feature vectors. Machine Learning Techniques include the Euclidean Distance, the Linear Discriminant Function, the Regularized linear discriminant function (RLD), the Projection distance, the Modified projection distance and the SVMs.

It can therefore be stipulated that:- (1) the highest classification rates by a considerable margin for all kind of features were obtained from the regularized linear discriminant function (RLD) with less computational cost. (2) normalizing the absolute frequency to the relative frequency followed by the power transformation improved the classifiers' performance significantly.

Furthermore, relative frequency and power transformation are techniques that showed considerable improvements in the classification performance. The implication of these results is that, these techniques can take a great role in getting higher performance without unnecessarily employing sophisticated techniques to represent texts for classification purposes even at lower dimensionality.

3.5 Experiments Using Large Samples

3.5.1 The Data For The Experiments – published Splits

We used two popular data sets in our experiments. These were Reuters-21578 and the OHSUMED data collections. The following two subsections describe them.

3.5.1.1 Reuters-21578

A benchmark collection for text categorization research called Reuters-21578 was used. This has been widely employed by other researchers, too [49, 76, 83, 82, 96, 100]. Reuters-21578 is composed of 21,578 articles manually classified into 135 categories. One textual

document may belong to one or more categories. Hence Reuters-21578 poses both multi-class and multi-label problems.

We used the ModApte Split [100] which contains 12,902 articles. In this split the training set contains 9,603 documents and for the test set 3,299 documents, and 8,676 documents are not used. ModApte Split is the most commonly used split among the splits. In total we used 115 categories in the experiments.

3.5.1.2 OHSUMED(HD-119)

The OHSUMED collection was first published as a text retrieval test collection in 1994 [38]. It contains 348,566 MEDLINE references from the years 1987 to 1991. Although all of the references have titles, only 233,445 have abstracts. According to Lewis et al. [56], in text categorization problems, queries and relevance judgments in the collection are ignored. We followed the split used in [56]. There are 183,229 documents from the years 1987 to 1990 which were used as a training set, and 50,216 documents from the year 1991 which were used as a test set. Categories are based on medical subject headings (MeSH categories).

Like in some of the past studies [56, 16], the focus here was on 119 MeSH categories in the heart disease sub-tree (HD-119) of the cardiovascular diseases tree structure. In the experiments, after preprocessing and labeling, we extracted 90 MeSH categories. We used 12739 abstracts of documents as a training data set. And 3742 abstracts of documents as test data set. Therefore results presented here are for 90 categories. The HD-119 subtree is a multi-label problem meaning that one document may belong to one or more categories.

3.5.2 Lexicon Generation

In general, function words are not useful to represent document features while preserving separability. Therefore, before generating feature vectors, stop words and general words were removed with reference to a stop list prepared beforehand to reduce the features. This also reduced the amount of memory required for storage as well as the processing time required by a classification system.

Even when a stop list was used to remove needless words, a lot of words still remained. Hence, words with a frequency value of 5 or less in all training data were removed to reduce further the remaining words. This removal of words reduced the lexicon size from 24868 to 7474 for Reuters and from 32,724 to 11,265 for OHSUMED. According to [82] this word removal does not affect the classification performance. We did not perform any type of stemming.

3.5.3 Implementation of Dimension Reduction

As described in 4.14, PCA was used to reduce the dimensionality and used fewer principal components. We then applied CDA to appropriate amounts of principal components. We choose the dimensionality before application of CDA experimentally, meanwhile observing the classification performance. This was done until no classification improvement occurred, even when more features were added afterward.

3.5.4 Classification and Performance Measures

3.5.4.1 Learning Methods for Classification

In the experiments k nearest neighbor (kNN) and support vector machines were used. kNN can easily handle both multi-class and multi-label problems simultaneously as compared to other classification methods. Since the Reuters-21578 and the OHSUMED collection are both involve multi-class and multi-label problems, Therefore kNN was used in the classification process.

To determine the kNN a cosine similarity function was used in the experiments. The likelihood scores were obtained using similarity scores. Then the binary category assignments were obtained by specifying a threshold. The k value was experimentally varied from 1 until when the classifier gave more errors rather than improving the performance. Detailed explanations of kNN can be found in 2.2.4.

Another machine learning method is the support vector machine described in 2.2.3. We used the SVM^{Light} package [41]. In the experiments we divided each classification task into n binary classification problems. The linear kernel was adopted. This is because a considerable a number of studies have shown that linear classifiers outperform non-linear ones in ATC [96], [100], [41]. Unless otherwise mentioned, we used the default parameters of the package to construct the training model.

The third learning method that was used is the logistic discrimination method. This is described in details in Section 2.3.4. Therefore the details are omitted in this section.

3.5.4.2 Measuring Classification Effectiveness

We adopted the recall, precision and F-measure for performance evaluation of classification effectiveness. These measures are regarded as standard evaluation methods for classification systems in automatic text classification. The details can be found in 2.5.1. Micro-averaging and macro-averaging strategies are commonly adopted. To enable comparisons with other works in the literature we adopted micro-averaging and report F-measure scores.

3.5.4.3 Statistical Significance of Improvements

Statistical significance testing gives an insight into any apparent improvement in the performance of algorithms or methods. It is therefore desirable to perform statistical analysis to show whether the proposed methods really have an impact on the performance of text categorization.

We used the methods described in Section 2.5.2 to perform a statistical analysis. The null hypothesis H_0 is that RFPT and TFIDF provide the same error rate when learning and classification is done on the same test set T. Statistical analysis results are presented in Section 3.5.5.2.

3.5.5 Empirical Results of Published *Splits*

3.5.5.1 The Effect of Feature Transformation

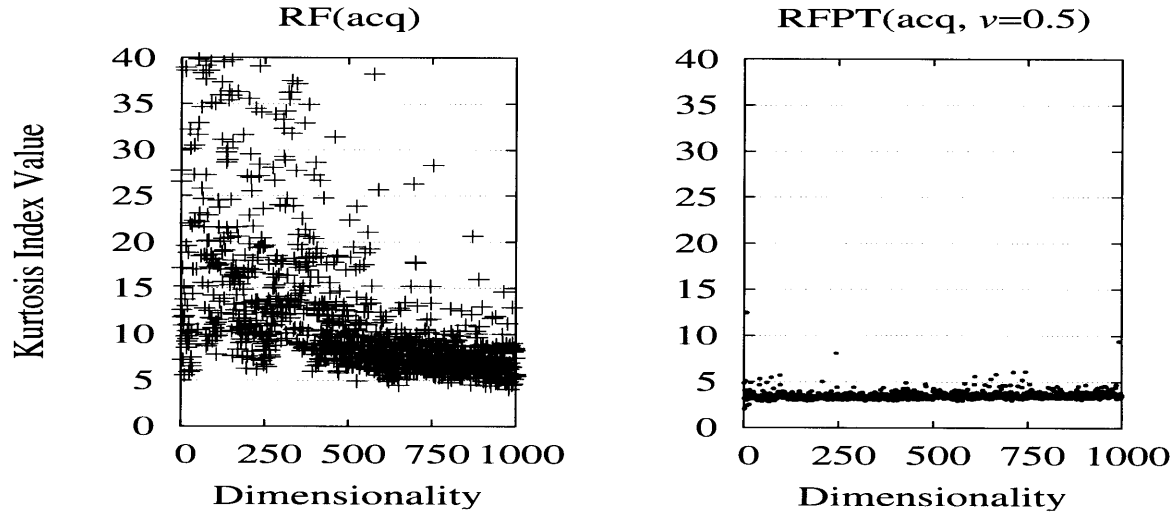
Fig. 3.3 plots the indexes of skewness and kurtosis. This figure illustrates the effect of feature transformation on both the Reuters-21578 and the OHSUMED data sets. For the Reuters data set an example from the acquisition class was used (see 3.5.1.1 for the description of this data set). For the OHSUMED data set the data from heart diseases was taken as an example.

It can be observed that the data before power transformation (i.e., RF) are skewed as shown by the skewness index. In Fig. 3.3(a), it can be observed that the values of κ of (3.14) by the RF take a wider range which is mainly from around 5 to above 40. This indicates that the distributions of the RF are accompanied by a property of peakedness because they show large positive values of the kurtosis index. The problem was mitigated for the RFPT and the kurtosis values closely clustered around 3. Fig. 3.3(b) indicates both positive and negative skewness of the variables in the distribution of the RF. This problem was solved when the proposed transformation was applied as shown by values of the skewness index from the RFPT.

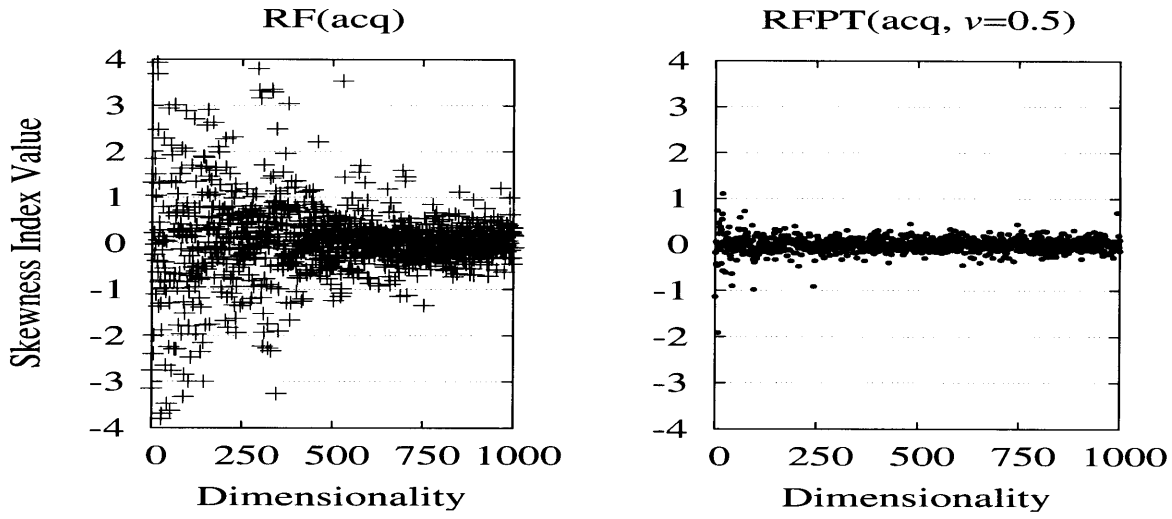
The trend of kurtosis and skewness is similarly observed for the TFIDF in figures 3.3(c), 3.4(a), 3.4(b) and 3.4(c). It is noteworthy that the distributions of the RFPT are more Gaussian-like than that of the TFIDF.

Figure 3.5 gives another view of the impact of feature transformation. The general observation from the experimental results is that the classification performance of transformed features are higher than those of untransformed features. This means it is better to use feature transformation techniques than untransformed features. This is because transformed features do not depend on text length and have better sample distribution – Gaussian-like distribution is advantageous to classification systems. The detailed explanation for this is given in Section 3.3.

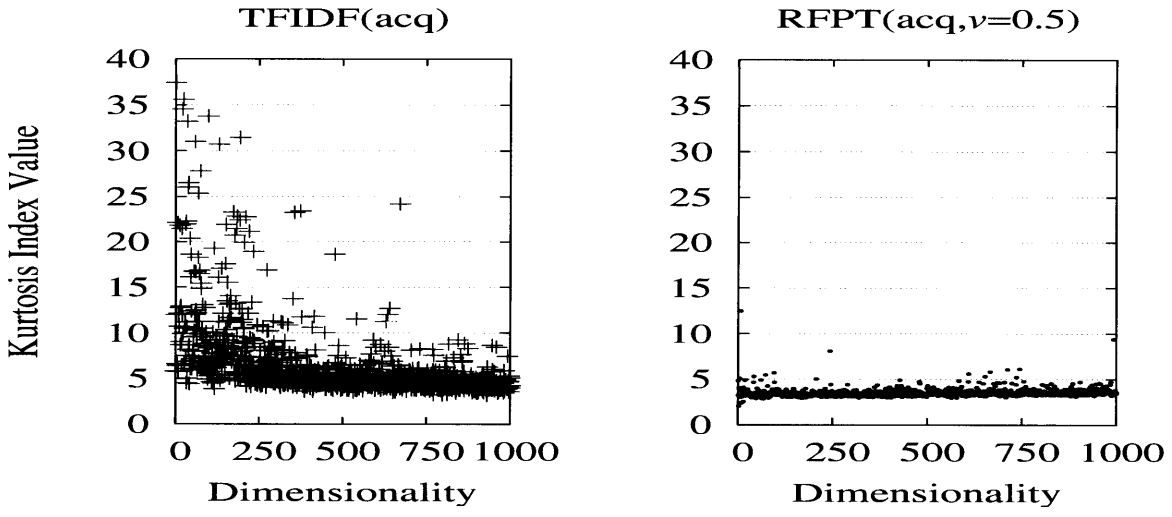
Looking at Fig. 3.5, we also noted a consistent trend which is seen in the ATC



(a) Index of kurtosis for the Reuters-21578 data set

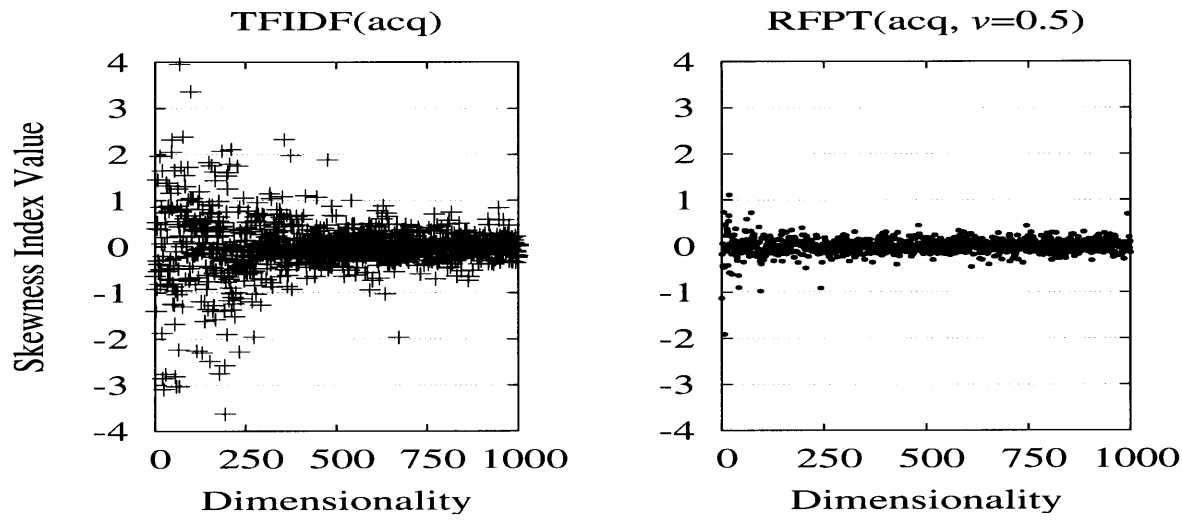


(b) Index of skewness for the Reuters-21578 data set

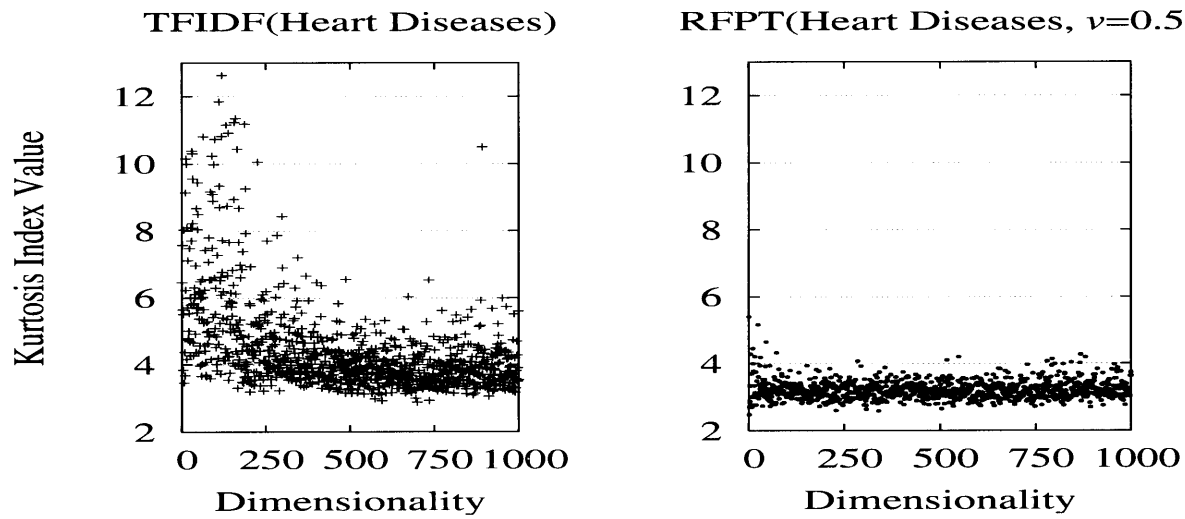


(c) Index of kurtosis for the Reuters-21578 data set

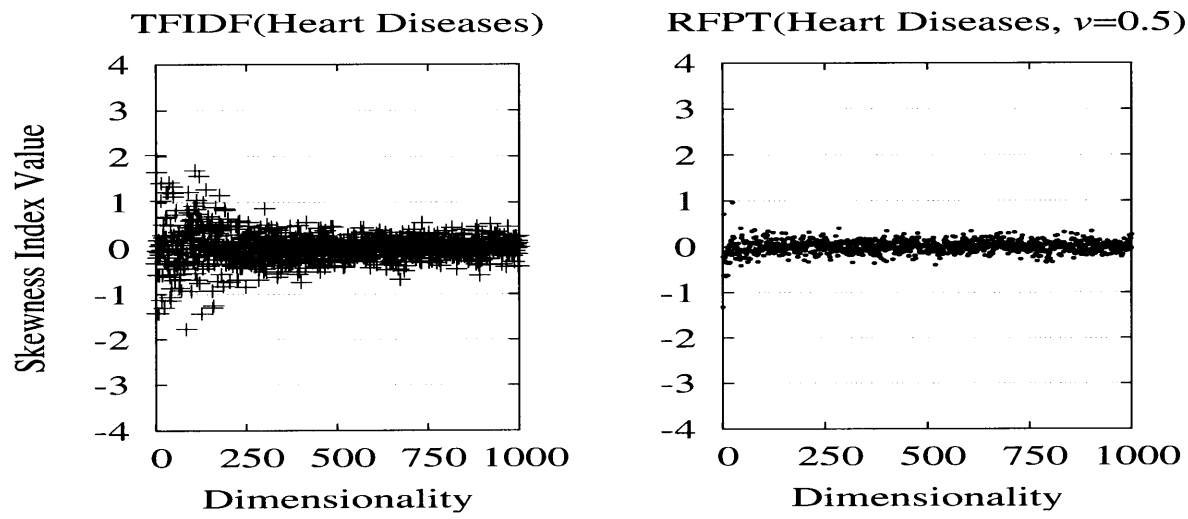
Figure 3.3: The effect of feature transformation based on kurtosis and skewness indexes



(a) Index of skewness for the Reuters-21578 data set

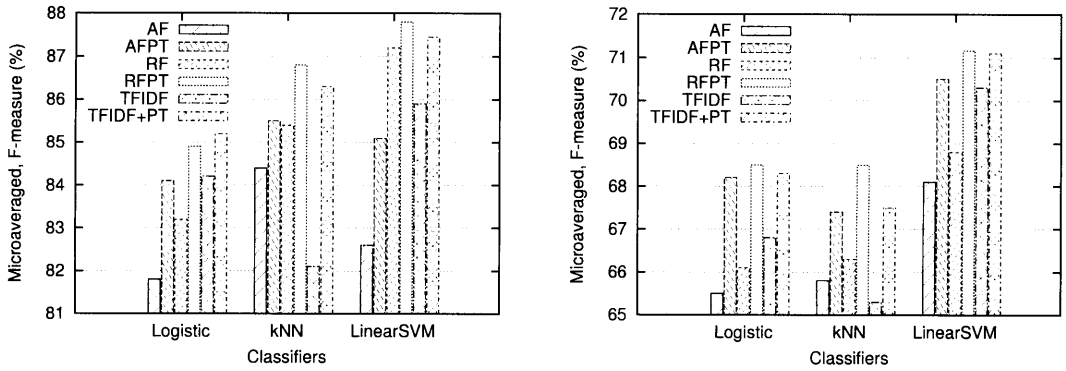


(b) Index of kurtosis for the OHSUMED data set



(c) Index of skewness for the OHSUMED data set

Figure 3.4: The effect of feature transformation based on kurtosis and skewness indexes



(a) The Reuters-21578 data set

(b) The OHSUMED data set

Figure 3.5: The impact of feature transformation on classification performance. Features used include absolute term frequency (AF), AF followed by power transformation(AFPT), relative term frequency(RF), power transformed RF (RFPT), term frequency weighted by inverse document frequency(TFIDF), and power transformed TFIDF (TFIDF+PT)

literature. That is to say classification performances for the Reuters-21578 collection are always higher than those for the OHSUMED data set. This is because the heart disease subtree (HD-119) consisted of closely related documents between classes. It is also useful to note that results from the HD-119 can not be directly compared with results in [41]. This is because the work in [41] uses relatively general medical subject headings (MeSH) from higher level tree structure of 23 Mesh categories. By the term 'general' we mean that the documents from one class to another are not very closely related to each other compared to those in HD-119 subtree. Hence they are relatively easier to classify.

Furthermore, we noted that the RFPT features consistently gave better results than the TFIDF features for the both classifiers and for both of the two data sets. The difference is more obvious with kNN. A similar situation is seen for SVM when Reuters data set was used. We also found classification improvements with the TFIDF+PT. But in most cases, its performances are lower than that with the RFPT features.

Another interesting point can be noted from Fig. 3.5 is that the power transformed TFIDF i.e., TFIDF+PT achieved higher classification performance than the conventional TFIDF. This means power transformation improved the sample distribution as explained earlier.

The results of logistic discrimination are included in Fig. 3.5. However we note that the logistic discrimination classifier can give poorer decision boundaries than support vector machines since it does not use margins. This is due to the fact that it is influenced by regions of high density rather than samples close to the decision boundary.

3.5.5.2 Statistical Analysis of Improvements with RFPT

We carried out statistical analysis to find out whether the improvements were significant. In this Section the analysis was between RFPT and TFIDF on respective classifiers and data sets. Classification scores are considered in terms of the number of true positives (TP), false negatives (FN), true negatives (TN) and false negatives (FN)*.

We were particularly interested to know whether RFPT performed statistically better than TFIDF. The null hypothesis was that RFPT and TFIDF would achieve the same performance on test data.

Table 3.2: Results of the statistical analysis: RFPT versus TFIDF. p-values are indicated as p.

	Reuters		OHSUMED	
	kNN	SVM	kNN	SVM
RFPT (F-measure)	86.8%	87.8%	68.5%	71.16%
TFIDF(F-measure)	82.1%	85.9%	65.3%	70.3%
McNemar	$p <$ 2.2e-16	$p =$ 9.088e-05	$p =$ 0.003497	$p =$ 3.366e-05
χ^2 Test	$p =$ 1.63e-13	$p =$ 0.02563	$p =$ 0.04427	$p =$ 0.03403
Z-test	$p =$ 1.083e-31	$p =$ 9.088e-05	$p =$ 0.003497	$p =$ 3.366e-05

Table 3.2 summarizes the analyzed statistical results. When using kNN on Reuters we found that RFPT performed statistically better ($p < 0.01$) than TFIDF. Similarly, when using SVM on the same data set, RFPT achieved statistically better results ($p < 0.05$).

In the case of OHSUMED data set, we generally found that RFPT was statistically better ($p < 0.05$) than TFIDF. Based on this statistical evidence, we can therefore confirm that the RFPT performs better than TFIDF.

3.6 Experiments on OCR Based Texts

The digitization process of various printed documents involves generating texts by an OCR system for different applications including full-text retrieval and document organization. However, OCR-generated texts have errors with regard to the present OCR technology. Moreover, previous studies have revealed that as OCR accuracy decreases the classification performance also decreases. The reason for this is the use of absolute word frequency as a feature vector. Representing OCR texts using absolute word frequency has limitations such as dependency on text length and word recognition rate consequently lower classification performance due to higher within-class variances. We

*The classification scores were fed to a statistical software called R. The software is available at www.r-project.org

present experiments with feature transformation techniques which do not have such limitations and present improved experimental results from all classifiers used.

3.6.1 Background Information on OCR Texts

In recent years, the main means of information exchange has been changing from the traditional printed information to digital data. This is due to the fact that digital data such as text, image, and audio can be transferred and retrieved faster, more flexibly and more easily. Activities such as digital publishing and the digital library might become the main sources of information in the near future. As a matter of fact digitization projects have been taking place [6, 63]. Since there has been a need to make archives accessible through digital information systems, other traditional libraries might be considering converting printed archives into digital data. Digitized materials might need techniques from automatic text classification (ATC) to be applied to different domain applications such as automatic indexing for Boolean information retrieval systems; document organization; information filtering and hierarchical categorization of web pages.

When working with printed documents there might be two ways to generate digital texts which are keying texts into computer system and using optical character recognition (OCR) systems whereby text materials are extracted from digital text images. The LDI project team [63] argues that the Harvard University Library keying process costs approximately 10-13 times more per page than by using uncorrected OCR. They refer to uncorrected OCR due to the fact that OCR-generated texts generally have errors [71, 74]. The authors in [102] showed the impact of OCR accuracy on automatic text classification such that as OCR recognition rates dropped down, the classification performance decreased. In this work, we describe feature transformation techniques for OCR-generated texts and present improved experimental results from all classifiers used.

3.6.2 Related Works on OCR Based Texts

This paper describes techniques for transforming features from OCR-generated documents. The literature shows rare research work done previously on OCR in relation to automatic text classification (ATC). This section gives a brief survey of research that might be relevant.

The work in [44] reports on OCR text representation for learning with a focus on different techniques for automatic construction of relevant features from Germany language documents. Their study considered various features including all words, elimination of stop-words, morphological and composite analysis, and use of n-grams. Some important results are given. The fact that they used different language datasets, means their work is remarkably different in various ways. Not only didn't they perform feature transformation

techniques but also they didn't use the benchmark collection for text categorization from which we generated image text documents to study the impact of transformed features on OCR-generated documents.

Frasconi et al. [28, 29] performed experiments on text categorization for multi-page documents extracted by an OCR system. In contrast they used untransformed word counts i.e., bag of words to represent the texts. They also used information gain techniques for feature selection to reduce the number of features. However we employed principal component analysis (PCA) after using term selection techniques for dimension reduction.

The authors in [102] investigated the impact of OCR accuracy on automatic text classification using absolute word frequency as an OCR text representation technique. Since absolute frequency depends on text length we propose techniques to solve this problem.

Most of the research (if not all) mentioned above and in [86], exhibit notable differences with ours. The biggest difference is that they reported experimental results from OCR texts represented by untransformed features. Hence we focus on transformed features for representing OCR texts. Experimental results reveal improved classification performance.

3.6.3 The Data Used for OCR Text Experiments

In order to study the impact of transformed features on OCR-generated documents in the automatic text classification, a training sample was required. Therefore, we used the Reuters-21578 text benchmark collection for English text classification. The Reuters-21578 is composed of 21578 articles manually classified in 135 categories.

In the experiments, a total of 750 articles i.e., 150 articles per category randomly selected from five categories (acq, crude, earn, grain, trade), were used. Since the sample size was not large enough, the sample was divided into three subsets each of which included 50 articles per category. When a subset was tested, the rest of the two subsets were used as learning samples in order to keep the learning sample size as large as possible while maintaining independence between the samples for learning and testing. Classification tests were repeated for three subsets and the average performance measures were computed.

3.6.4 Experimental Setup of the OCR Texts

There are three general steps to be followed in the experiments. These included text image generation, OCR text generation and automatic text classification. The following are descriptions of these steps.

3.6.4.1 Text Image Generation

Textual documents from the Reuters collection were printed out. Paper texts were digitized using a scanner into images of different resolutions including 300 dpi, 200dpi, 150dpi, 145dpi, 140dpi, 135dpi and 130dpi. Figure 3.6(a) and 3.6(b) show examples of the text images for 300dpi and 140dpi respectively.

3.6.4.2 Text generation by an OCR system

In practice, OCR technology can be used in areas such as in automatic entry of information into a computer and bank check processing. Since printed information can usually be entered into a computer system by scanning, then texts have to be generated from those images. The text images generated above were converted into ASCII texts by OCR software "OKREADER2000". The essence of this step was to simulate a practical example of the use of OCR technology.

Examples are given in Table 3.3(a) and 3.3(b). The obtained texts were compared with the original texts in the Reuters collection to compute the average character recognition rates and the average word recognition rates for each dpi value. The average character recognition rates can be defined as

$$r_c = \frac{(s-e)}{s} \times 100, \tag{3.16}$$

where s and e are the total numbers of characters and the number of miss-recognized characters, respectively. The average word recognition rate can be defined as

$$r_w = \frac{(w-u)}{w} \times 100, \tag{3.17}$$

where w and u are the total number of words and the number of miss-recognized words, respectively.

3.6.4.3 Classification of OCR Texts

After obtaining ASCII texts, learning and classification experiments could be conducted. Automatic Text Classification experiments were carried out. There are four general steps involved in this process.

- Feature Generation: The procedures described in 3.3 were followed in generating features.
- Dimension Reduction: The principal component analysis (PCA) was used in the experiments. Chapter 4 explains the details of this technique.

****

VIEILLE MONTAGNE REPORTS LOSS, DIVIDEND NIL

****

1986 Year

Net loss after exceptional charges 198 mln francs vs profit

250 mln

Exceptional provisions for closure of Viviez electrolysis

Plant 187 mln francs vs exceptional gain 22 mln

Sales and services 16.51 billion francs vs 20.20 billion

Proposed net dividend on ordinary shares nil vs 110 francs

Company's full name is Vieille Montagne SA <VMNB.BR>.

REUTER

(14876)

(a) Text image of 300dpi

****

VIEILLE MONTAGNE REPORTS LOSS, DIVIDEND NIL

****

1986 Year

Net loss after exceptional charges 198 mln francs vs profit

250 mln

Exceptional provisions for closure of Viviez electrolysis

Plant 187 mln francs vs exceptional gain 22 mln

Sales and services 16.51 billion francs vs 20.20 billion

Proposed net dividend on ordinary shares nil vs 110 francs

Company's full name is Vieille Montagne SA <VMNB.BR>.

REUTER

(14876)

(b) Text image of 140dpi

Figure 3.6: Examples of text images

Table 3.3: Examples of ASCII texts converted by OCR software

(a) ASCII text from text image of 300dpi

VIEILLE MONTAGNE REPORTS LOSS, DIVIDEND NIL
1986 Year
Net loss after exceptional charges 198 min francs vs profit
250 min
Exceptional provisions for closure ofViviez electrolysis
Plant 187 min francs vs exceptional gain 22 min
Sales and services 16.51 billion francs vs 20.20 billion
Proposed net dividend on ordinary shares nil vs 110 francs
Company's full name is Vieille Montagne SA <VMNB.BR¿.
REUTER
(14876)

(b) ASCII text from text image of 140dpi

V1FTLLH MONTAGNB RLPORTS LOSS. DIVIDEND NJL
19S6Ye;ir
Net loss after exceptional charges 19S min fnincs vs profil
250 min
Exceptional provisions for closure of Viviez electrolysis
Plant 1S7 ruin fmncs vs exceptional gain 22 min
Sales and services 16,5 b̂illion rr,mcs v2̂0.20 biltion
Proposed net dividend OEI ordinary shares nil vs 110 francs
Company's full name is Vie/lie Monmgne SA &ll:VMNB.BR¿.
REmER
(14876)

- Learning Methods

  Various classifiers were trained accordingly using a learning sample as follows. The Euclidean distance classifier involved computing the mean vector of each class. The linear discriminant function required computation of the weight vector determined by the mean vector of each class and the pooled within covariance matrix of all classes. Training the projection and the modified projection distances needed the computation of the eigenvectors and the eigenvalues of each covariance matrix of the individual category [31]. Support Vector machines (SVMs) are methods that find the optimal hyperplane during training. In the experiments, C-support vector classification methods (C-SVC) with linear and radial basis (RBF) functions were used. Particularly, we used the SVM library (LIBSVM Version 2.33) developed by Chang and Lin [15].

- Measuring Performance: Recall, Precision and Classification rates were adopted.

3.6.5 Empirical Results of OCR Texts

In this section we present experimental results from different features that include absolute word frequency, relative word frequency and their power transformations. Table 3.4 shows the classifiers' classification rates versus character recognition rates and the word recognition rates from absolute word frequency at different resolutions. In this table it can be observed that, as the resolution of text images decreased, the character recognition and word recognition rates by an OCR system also decreased. In other words at relative higher resolutions, it was possible to obtain less recognition errors by using OCR systems. Similarly, classification rates of OCR texts decreased with an increase in OCR errors.

Table 3.4: OCR text classification rates (%) for absolute frequency vs. character recognition rates (%) and word recognition rates (%) by an OCR system at different resolutions (dpi)

Resolution (dpi)	130	135	140	145	150	200	300
Word Recognition Rates	41	53.8	63.7	72.1	84.3	92.9	97.2
Character Recognition Rates	57.7	71.6	82.8	89.8	96	98.4	99.3
Euclidean Distance	44.9	51.9	58.4	62.1	67.2	70.7	74.3
Linear Discriminant	65.7	74.8	80.1	86.0	88.3	89.9	91.1
Projection Distance	75.2	83.3	87.1	88.4	90.1	90.7	91.2
Modified PD	78.1	86.3	89.3	91.6	92.5	92.8	93.1
Linear SVM	76.1	84.9	87.5	89.9	92.0	92.9	93.3
RBF SVM	64.5	76.8	82.0	86.3	89.5	91.3	92.1

With regard to the OCR texts, the summarized best classification rates from all features using different classifiers are given in table 3.5. It is notable that transformed

features improved the performance of all classifiers used. Performing power transformation on relative frequency for example made all classification rates to rise as high as over 91%.

Table 3.5: The summary of best classification rates in % at 300dpi

Classifiers	AF	AFPT	RF	RFPT
Euclidean Distance	74.3	89.6	86.0	91.1
Linear Discriminant	91.1	92.8	93.9	94.5
Projection Distance	91.2	94.9	93.3	95.7
Modified PD	93.1	95.3	94.7	96.1
Linear SVM	93.3	95.1	94.3	95.3
RBF SVM	92.1	93.5	94.4	95.3

Table 3.4, 3.5 and 3.6 also reveal that the modified projection distance (MPD) outperformed all the classifiers used in terms of accuracy and robustness. In other words, this classifier gave the highest classification rates even when there were more OCR errors. For example when OCR word recognition rate was 41%, MPD was accurate achieving 78.1%. This is when absolute frequency was used as feature vectors. This improved to 91.7% † by employing power transformation on the relative frequency (RFPT). And when OCR word recognition rate was 97.2%, the MPD's classification accuracy was improved from 93.1% to 96.1%. This was when RPPT features were used.

Table 3.6: The summary of best recall/precision break even point (BEP) in % at 300dpi

Classifiers	AF	AFPT	RF	RFPT
Euclidean Distance	43.6	48.6	42	64.7
Linear Discriminant	87.3	90.1	87.2	91.3
Projection Distance	42.1	52.9	64.2	90.4
Modified PD	45.5	55.9	65.1	91.8

It was observed that the classification rates were significantly improved using relative frequency instead of the absolute word frequency. For instance, the accuracy of the Euclidean distance classifier was improved by 11.7% at 300dpi and by 36.2% at 130dpi. In addition, power transformation on absolute frequency (AFPT) also improved the performance of all classifiers used. However, it is clear that the use of AFPT gave more classification errors when there were more OCR errors in generating texts.

Power transformation on the relative frequency (RFPT) further improved the classification accuracy of each classifier used. For example the accuracy of the Euclidean distance classifier was improved cumulatively by 16.8% at 300dpi and by 40.4% at 130dpi. RFPT improved classification accuracy such that all classifiers exhibited over 91% classification rates.

†Note that 91.7% is a detail which is not reported in table 3.4 and 3.5. It was obtained at resolution 130dpi.

Not only did the classifiers' performance rise by doing power transformation on relative frequency, but also the robustness of classifiers increased such that even when OCR systems gave a lot of unacceptable huge number of errors, the performance was considerably higher than when using untransformed features in representing OCR texts for classification purposes. For example, at highest level for OCR errors, when RFPT was used, the worst classifier performed with an 85.5% accuracy rate. The best classifier came up with classification rate of 91.7%.

It is also interesting to note that transformed features particularly relative frequency do not heavily depend on word recognition rates by the OCR systems. The differences in accuracy between the absolute frequency and the transformed features increase as the word recognition rates by OCR systems decrease.

3.6.6 Summary of the OCR Text Experiments

In this Section we have shown the impact of using transformed features for OCR-generated documents in automatic text classification. The findings show that using transformed features significantly improved the performance of all classifiers used. Even when OCR systems gave a lot of errors by representing texts with transformed features it was encouraging to obtain as high classification rates as possible. The implications of these results are that, with error-prone OCR texts it is possible to automate the classification tasks and use the automation in different applications such as information retrieval, information filtering and document organization. Future experiments will include increasing the sample size from more categories for real world applications in text classification. Also, error correction of words by spell checking also remains for future study to improve text classification performance.

3.7 A Summary of Document Representation

This chapter proposes the use of relative frequency with power transformation (RFPT) for document representation in text classification. Experimental evaluation has been presented. The author evaluated RFPT mainly in three kinds of experiments. First, experiments were conducted using randomly selected samples. Second, experimental evaluation was carried out using the widely published splits of benchmark text collections. Finally, experiments were performed using noisy texts which are generated using OCR technology.

This technique improved text classification performance. Feature transformation, in particular the use of relative frequency with power transformation, improves the performance of classifiers significantly. Our empirical results show that RFPT is generally superior to conventional features namely the term frequency weighted by inverse docu-

ment frequency (TFIDF).

The implication of these results is that, these techniques can take a great role in getting higher performance without unnecessarily employing sophisticated classification. After all, it is desirable to have better classification performance with efficient and simpler techniques than to have complex methods to be employed in the classification process.

Chapter 4

Feature Reduction

4.1 Introduction

In automatic text classification by machine learning, feature vectors to represent texts are commonly generated by the use of the term frequency, consequently the so called curse of dimensionality arises. Extremely high-dimensional space increases learning complexities which is detrimental to the classification performance of a system. This problem arises because it involves simultaneous increase in dimensionality of the feature vectors with the increase in the number of words (lexicon size). For example, in this research the dimensionality of the word frequency vector of all used articles from Reuters-21578 amounted to 24,868 words. Such a high dimensional feature space needs large calculation resources for processing and memory storage capacity for classification.

In order to solve this problem the dimensionality has to be reduced. This is preferably done while extracting informative features that can improve classification performance. Conventionally, the dimensionality reduction techniques that have been used are latent semantic indexing (LSI) [82] and the mutual information (MI) method [41, 82]. In recent years, some works applied principal component analysis (PCA) to reduce the dimensionality [48]. However PCA has not been experimentally studied in relation to transformed features in automatic text classification. Practically, it is not accurate to assume that it would work without experimental studies in the field of ATC.

It is notable further that PCA [24] and LSI [90] ignore category specific information. For example, PCA maximizes the total scatter across all class resulting into retention of non-discriminative information. To avoid the drawbacks of PCA, it might be desirable to perform canonical discriminant analysis (CDA) [89] - a common statistical method in other fields of research, but not common in ATC. However, there is a singularity problem of the within-class covariance matrix that arises due to higher dimensionality compared to the sample size. Therefore we study the combined dimensionality reduction with PCA. We note that the approach we present in this work is not common in ATC literature.

This combination can be known as the PCA+CDA algorithm.

Contrary to our approach for dimensionality reduction, Kim et al. [46] studied different algorithms for dimensionality reduction which include the Linear Discriminant Analysis/Generalized Singular Value Decomposition (LDA/GSVD) algorithm. However, they could not report some of their experimental results because the algorithm namely LDA/GSVD on the Reuters-21578 data collection ran out of memory while computing the GSVD. This is because the algorithm involves a lot of calculation resources as indicated on page 49 of their paper. This means their algorithm could not handle a larger number of data like those in the OHSUMED data set which we used in the experiments.

The rest of this chapter is organized as follows. Section 4.2 provides the review of the state of the art on feature reduction in text classification (TC). Section 4.3 describes the proposed methods for feature reduction by extraction. Particularly, we study the PCA+CDA algorithm. We also propose an extension of this algorithm to handle multi-label data. Furthermore we propose a technique called integrated discriminant analysis (IDA) with the ability to handle multi-label data as well. Section 4.4 and 4.5 present experimental results. It is observed that the proposed methods in this chapter are effective in extracting fewer features with high discriminating power. Section 4.6 presents the concluding remarks of this chapter.

4.2 Conventional Methods for Feature Reduction

Methods for feature reduction can be grouped into two categories. These include feature selection and feature extraction. In this section we review representative methods for every category.

4.2.1 Document Frequency Thresholding (DF)

The DF is one of the methods classified under the feature/term selection category. Document frequency can be defined as the number of documents in which a term occurs [97]. The document frequency can be computed by using the training sample. Then those terms which are below a predetermined threshold can be removed. This idea is based on Zipf's law with the assumption that very rare terms are not influential in the classification effectiveness. This is true especially when the rare terms introduce false features. Based on the same law most frequent words do not represent documents discriminately. This is one of the simplest methods for reducing the size of the vocabulary list

In our experiments when using the randomly selected sample from Reuters-21578, we removed the words which had document frequency below 3. This was done after removing the functional words with reference to a stop-word list. Similarly, with the published split namely "ModApte" split of the the Reuters, we removed words with values of document

frequency below 6. According to [97], it is possible to reduce the dimensionality (the vocabulary list) by a factor of 10 without noticeable loss in classifier's performance.

The problem with DF is when a category of texts has very few documents. In practice ModApte split of Reuters21578 for example has document categories with only one training document. Therefore one should use this method with great care.

4.2.2 Pointwise Mutual Information (PMI)

The pointwise mutual information method is one of the feature selection methods. Conventionally, it has been reported by many other researchers* in text categorization [82, 97, 93]. Consider a word/term t and a category ω. PMI can be defined by

$$PMI(t,\omega) = \log\frac{P(t,\omega)}{P(t) \times P(\omega)}, \tag{4.1}$$

where $P(t,\omega)$ is the joint probability of t and ω. $P(t)$ is the marginal probability of t and $P(\omega)$ is the marginal probability of category ω. Let τ be the number of times that t and ω co-occur and ν be the number of times that t occurs without ω. Furthermore denote ψ as the the number of times that ω occur without t and denote N as the total number of documents in ω. The estimation of PMI can be given by

$$PMI(t,\omega) \approx \log\frac{\tau \times N}{(\tau+\psi) \times (\tau+\nu)}. \tag{4.2}$$

To get a measure of the goodness of a term in all features, category specific scores of a term can be combined in one of two ways:

$$PMI_{avg}(t) = \sum_{j=1}^{C} P(\omega_j) PMI(t,\omega_j), \tag{4.3}$$

or alternatively the estimation can be defined by

$$PMI_{max}(t) = \max_{j=1}^{C}\{PMI(t,\omega_j)\}. \tag{4.4}$$

The terms with higher scores are chosen using a threshold such that the terms with values below the threshold are discarded.

The PMI method has a weakness in that the score is strongly influenced by marginal probabilities of terms, as it can be noted in the equivalent form:

$$PMI(t;\omega) = \log P(t|\omega) - \log P(t)$$

*In most of the works it is called mutual information. According to [93] the name contradicts with the information theory. It should therefore be correctly called pointwise mutual information

Rare terms always have higher score than common terms especially those terms with equal conditional probability. Thus it is sensitive to probability estimation errors as it is highly influenced by the marginal probabilities.

4.2.3 Mutual Information Method (MI)

One of the most used conventional techniques for dimensionality reduction is the mutual information method (MI) [41, 81, 82]. It is one of the methods developed from the information theory. This is also called *Information gain* in [82, 97]. In this section we follow the definition from information theory [41].

Let $H(\mathbf{t}) = -\sum_{t\in\mathbf{t}} P(t)\log_2 P(t)$ and $H(\Omega) = -\sum_{\omega\in\Omega} P(\omega)\log_2 P(\omega)$ be the entropies of a random variables document $\mathbf{t}$, and category Ω, respectively. Denote $H(\mathbf{t},\Omega)$ as the joint entropy of these variables defined by

$$H(\mathbf{t},\Omega) = -\sum_{t\in\mathbf{t}}\sum_{\omega\in\Omega} P(t,\omega)\log_2 P(t,\omega). \tag{4.5}$$

The entropy is a measure of uncertainties in the particular random variables. Mutual information aims at reducing the uncertainty of a random variable as a result of knowing about the other variable. Formally, the mutual information can be defined by

$$\begin{aligned} MI(\mathbf{t};\Omega) &= MI(\Omega,\mathbf{t}) && (4.6)\\ &= H(\mathbf{t}) + H(\Omega) - H(\mathbf{t},\Omega) && (4.7)\\ &= \sum_{t\in\mathbf{t}}\sum_{\omega\in\Omega} P(t,\omega)\log_2\frac{P(t,\omega)}{P(t)P(\omega)}, && (4.8)\end{aligned}$$

where $P(t,\omega)$ is the joint probability between term t and class ω. $P(t)$ and $P(\omega)$ are marginal probabilities of term t and class ω, respectively.

In order to obtain a measure of mutual information for every term/word, equation (4.8) needs to be rewritten [81]. For C classes, mutual information between a term t and a set of categories Ω computed from equation (4.8) can be written as

$$\begin{aligned} t_{MI} &= MI(t,\Omega) && (4.9)\\ &= \sum_{j=1}^{C} P(t,\omega_j)\log_2\frac{P(t,\omega_j)}{P(t)P(\omega_j)}. && (4.10)\end{aligned}$$

In other words, MI compares the probability of observing a term t and ω together (i.e., joint probability) with the probabilities of observing t and ω independently. If there is a high association between t and ω, then the joint probability $P(t,\omega)$ will be larger than $P(t)P(\omega)$. Using (4.10), one can compute the measure of mutual information of every

term or feature. Based on a predetermined threshold, those features with higher values are retained for the classification process. Those features with values below the threshold are discarded.

4.2.4 Principal Component Analysis (PCA)

The PCA method falls into the category of methods of feature extraction. There are two contextual definitions of PCA that lead to the same algorithm. PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space such that the variance of the projected data is maximized. Equivalently, it can be defined as the linear projection that minimizes the average projection cost, referred to as the mean squared distance between the data points and their projections [7]. The former definition is preferably used in this work.

To solve the problem of high dimensionality, PCA can be applied [24, 32]. For the convenience of the reader, a short review is given below. From the set of training documents $\chi = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ the total covariance matrix Σ of the training sample is computed by

$$\Sigma = \frac{1}{N}\sum_{\mathbf{x}\in\chi}(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^T; \tag{4.11}$$

$$\mathbf{m} = \frac{1}{N}\sum_{\mathbf{x}\in\chi}\mathbf{x}, \tag{4.12}$$

where $\mathbf{m}$ is the total mean vector of training sample.

The corresponding matrix of eigenvalues $\Lambda = \text{diag}[\lambda_1 \ldots, \lambda_n]$ and eigenvectors $\Phi = [\Phi_1 \ldots \Phi_n]$ are obtained by the definition:

$$\Sigma\Phi = \Lambda\Phi, \tag{4.13}$$

provided that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.

Using eigenvectors which correspond to $m(m \leq n)$ largest eigenvalues, principal components $\mathbf{u} = [u_1 \ldots, u_m]$ are defined by the linear transformation

$$\mathbf{u} = \Phi^T\mathbf{x} \tag{4.14}$$

The reduced dimension of feature vectors is composed of m principal components $\mathbf{u} = [u_1 \ldots, u_m]$. This forms a projected or transformed data set $\Xi = \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ with the extracted principal components used in the classification process.

Note that the transformation matrix Φ satisfies orthonormal condition such that

$$\Phi^T\Phi = I \tag{4.15}$$

Multiplying Φ^T both sides of (4.13) and making use of (4.15) we obtain the maximized variance (i.e., eigenvalues) matrix by

$$\Phi^T \Sigma \Phi = \Lambda, \tag{4.16}$$

which satisfies the eigenvalue analysis of equation (4.13).

4.3 Proposed Methods for Feature Reduction

4.3.1 The PCA+CDA Algorithm

One drawback of PCA is that it ignores category specific information. It maximizes the total scatter across all classes (i.e. total variance) resulting in retaining non-discriminative information. Canonical discriminant analysis (CDA) [24, 32, 89] can be applied instead. However, direct application of CDA to high dimensional space data can lead into the singularity problem of the within-class covariance matrix. To avoid this problem, after reducing the dimensionality using PCA, we use CDA on the reduced features.

CDA considers category specific information since it uses the between class and the within-class scatter matrices that generalizes the equation (4.13) as

$$S_B \Phi = \Lambda S_W \Phi, \tag{4.17}$$

where S_B and S_W are between-class and within-class covariance matrices respectively. Their definitions are given by

$$S_B = \sum_{j=1}^{C} \frac{N_j}{N} (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T, \tag{4.18}$$

and

$$S_W = \frac{1}{N} \sum_{j=1}^{C} \sum_{\mathbf{u} \in \Xi_j} (\mathbf{u} - \mathbf{m}_j)(\mathbf{u} - \mathbf{m}_j)^T, \tag{4.19}$$

where the mean vector for each class $\mathbf{m}_j$ is defined by

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{u} \in \Xi_j} \mathbf{u}. \tag{4.20}$$

N_j and Ξ_j refer to the number of documents and the set of text sample in a particular class ω_j respectively. The other symbols are defined as before. The canonical discriminants can be obtained using equation (4.14). It is worth noting that since S_B is the sum of C matrices and each of the matrices is of rank 1 or less, hence S_B is of rank $C-1$. Therefore, there are at most $C-1$ nonzero generalized eigenvalues and their vectors. This implies

that $C-1$ dimensional space or less may give the highest classification performance.

It is worth noting that S_B and S_W are related to total scatter matrix Σ as

$$\Sigma_T = S_B + S_W. \tag{4.21}$$

The reader can note that the total scatter matrix Σ_T is defined in expression (4.11). The criterion for class separability in CDA is obtained by optimizing a function that leads to extraction of high discriminating power from data points. The objective is to achieve the highest separability of between-class data points meanwhile minimizing the variance of the within-class data points. One of the criterion is obtained by maximizing

$$J(\Phi) = \frac{\Phi^T S_B \Phi}{\Phi^T S_W \Phi}. \tag{4.22}$$

This criterion can also be written as

$$J(\Phi) = tr\{S_W^{-1} S_B\}, \tag{4.23}$$

or alternatively it can be written as

$$J(\Phi) = tr\{(\Phi S_W \Phi^T)^{-1}(\Phi S_B \Phi^T)\}, \tag{4.24}$$

which represents function of the projection matrix Φ explicitly. This means that $J(\Phi)$ should be large when the between-class matrix S_B is larger or when the within-class matrix is smaller. In so doing class separability is maximized.

4.3.2 Integrated Discriminant Analysis (IDA)

Motivated by some properties of CDA and PCA, this section proposes the integration of the two. The canonical discriminant analysis in 4.3.1 maximizes the variance ratio while extracting features with high discriminating power. In text classification usually one has to face smaller sample size compared to its dimensionality as discussed in the research problems (iii) and (v) of 1.4.3.1. Furthermore, CDA extracts $C-1$ features where C is the number of classes. These properties hinder the applicability of CDA in automated text classification (ATC). This is because the within-class matrix can be singular leading into a numerical instability problem. While the PCA in 4.2.4 maximizes the total variance [7], it does not consider class-specific information. This can result in retention of non-discriminative information.

The integrated discriminant analysis (IDA) in this section optimizes both variance ratio and the mean square error simultaneously. Therefore, the integrated discriminant analysis can be regarded as the integrated optimization of PCA and CDA. Consequently,

we call it Integrated Discriminant Analysis (IDA).

Let β be a constant in the range [0, 1]. Furthermore, let λ and Φ denote eigenvalues and corresponding eigenvectors, respectively. Then, IDA can be treated as a generalized eigenvalue analysis and defined by

$$(S_B + \beta S_W)\Phi = \{(1-\beta)S_W + \beta I\}\Phi\Lambda, \tag{4.25}$$

where S_B and S_W are between-class and within-class covariance matrices and their definitions are given in (4.18) and (4.19), respectively. I is the identity matrix. Λ and Φ are the eigenvalues and eigenvectors, respectively.

When $\beta = 0$, expression 4.25 tends to be equivalent to (4.17), therefore equivalent to the classical discriminant analysis and, when $\beta = 1$, it tends to be equivalent to equation (4.13), therefore equivalent to the principal component analysis (PCA). IDA solves the following optimization problem:

$$\max \frac{\Phi^T(S_B + \beta S_W)\Phi}{\Phi^T\{(1-\beta)S_W + \beta I\}\Phi}. \tag{4.26}$$

The determination process of the integration parameter β can be estimated via cross-validation techniques as proposed in [30, 36]. The expression (4.14) can be used to extract m features with high discriminatory information. Empirical results show that IDA is effective in TC.

It is worth noting that we compared IDA with a variant called regularized discriminant analysis (RDA) [30]. The definition of RDA is given by the modification of S_W satisfying the relation:

$$S_B\Phi = \{(1-\alpha)S_W + \alpha\frac{tr(S_W)}{n}I\}\Phi\Lambda, \tag{4.27}$$

where α is a constant in a range [0, 1]. n is the dimensionality. It is notable that the objective of RDA is to solve the singularity problem of S_W. Therefore, RDA is limited to $C-1$ features. In the case of limited number of document categories, it can extract insufficient features. On the other hand, IDA does not suffer from such limitations.

4.3.3 Discriminant Analysis for Multi-label Data

Let $\Omega = \{\omega_1, \omega_2, \ldots \omega_C\}$ be a finite set of labels or classes and its size be denoted by C. In classical text classification, multi-label texts are usually decomposed into C binary classification tasks [82], [96], [100] [41]. In other words, each document $\mathbf{x}_k \in \chi$ is labeled with a single label $\omega_j \in \Omega$. Methods in 4.3.1 and 4.3.2 can be directly applied to single label documents.

For data with multi-label, some considerations should be made to handle the problem. Each document $\mathbf{x}_k \in \chi$ is assigned multiple labels from Ω. Hence a labeled textual

document is a pair $(\mathbf{x}_k, L)$ where $L \subseteq \Omega$ is the set of labels assigned to $\mathbf{x}_k$. The single label problem is therefore a special case in which the size of L is one for every document.

One way of applying CDA to multi-label data, is to formulate C binary classification tasks. Consequently, a positive class and a negative class would be constructed. Therefore, CDA or IDA can be applied independently to all C problems. In doing so, it would result in at most one feature component per document. This may result in insufficiency of discriminative information. Another problem with binary decomposition for CDA, is that the correlation among classes is ignored.

Therefore, instead of constructing C binary problems, we make sure that every document with multiple labels contributes to all classes to which it belongs. In other words, when discriminant analysis techniques are applied, document $\mathbf{x}_k \in \chi$ appears in χ as many times as the size of L. The advantages of this technique include a high possibility of getting the optimal amount of discriminative information and the correlation of categories is taken into consideration. Experimental results show that this approach is effective in text classification.

4.4 PCA+CDA Experiments

4.4.1 Experimental Setup of Randomly Selected Samples

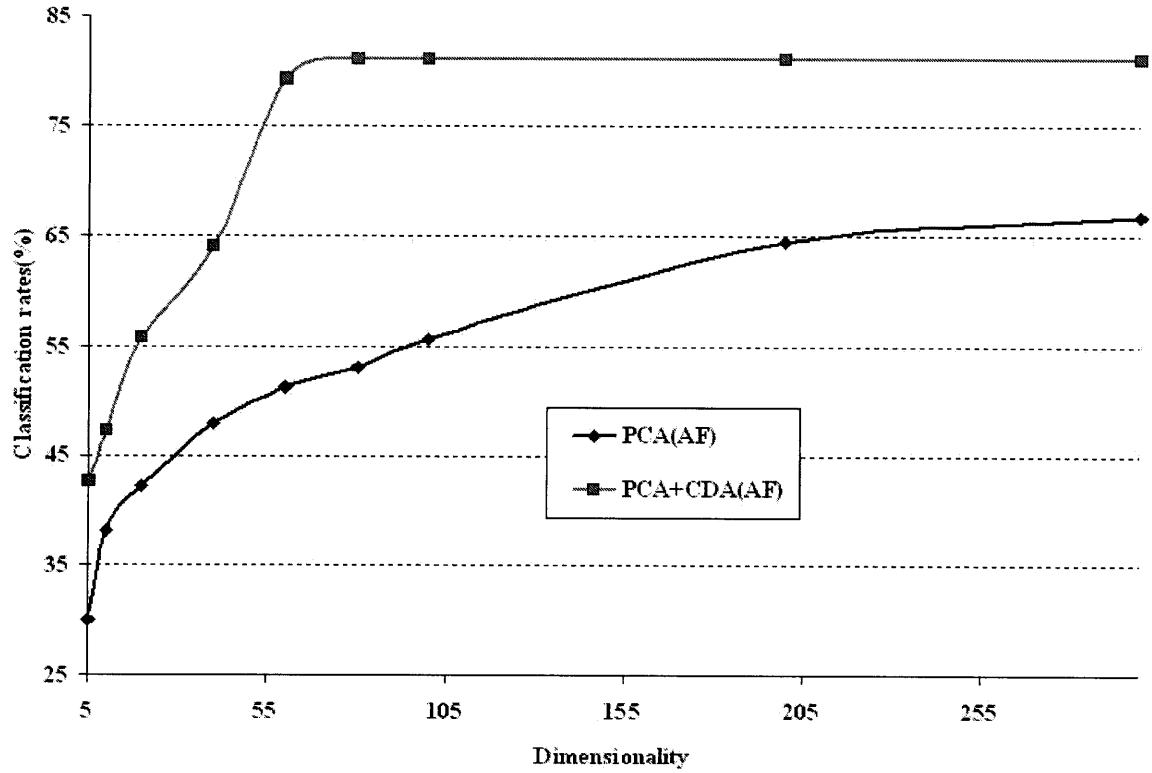
The data for experiments are described in 3.4.1. The vocabulary list was generated using the same procedure explained in 3.4.2. Learning methods used in the experiments are also described in 3.4.4. Particularly, we present results from Euclidean distance, modified projection distance and linear SVM.

The features used include absolute term frequency (AF) and relative term frequency with power transformation (RFPT).

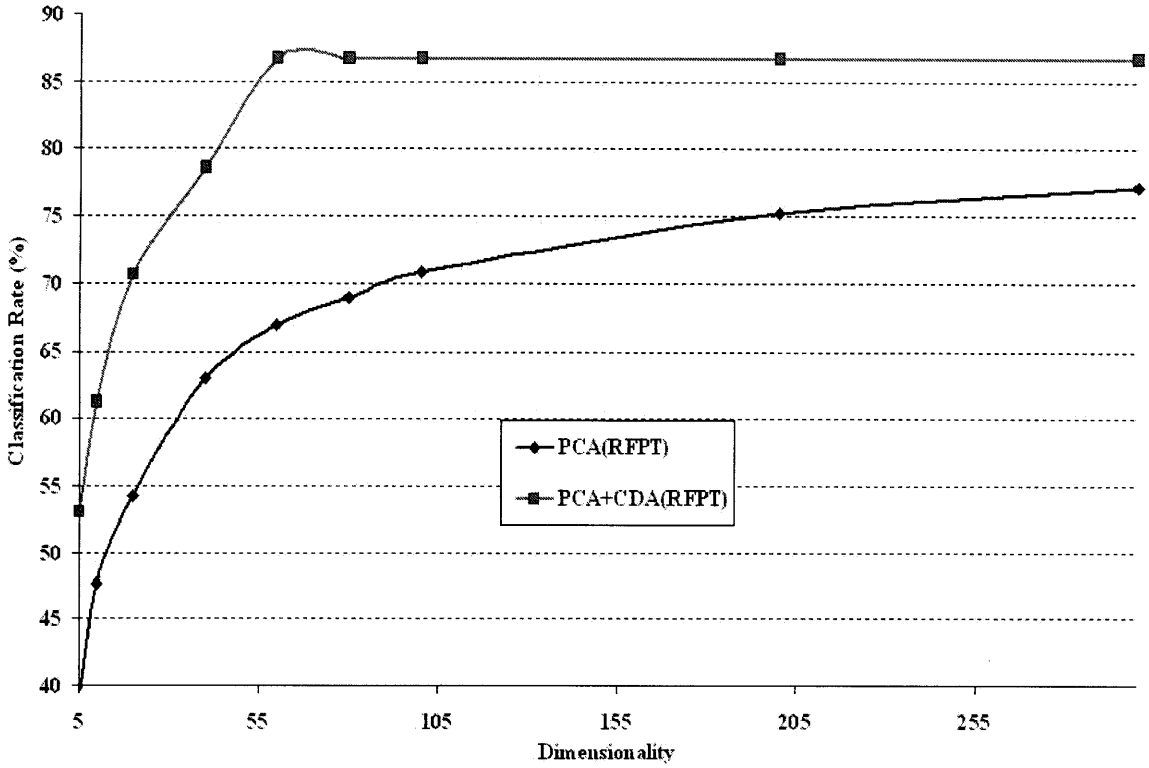
4.4.2 Empirical Results of Randomly Selected Samples

Figures 4.1 and 4.2 show the relationship between the average classification rates and the dimensionality of the feature vectors after power transformation from relative frequency features. It can be easily seen that the PCA+CDA algorithm achieved the highest classification rates.

The PCA+CDA algorithm reduced the dimensionality meanwhile extracting features with high discriminatory information. PCA+CDA was effective on both transformed and non-transformed features as illustrated in Fig. 4.1. It has been observed that PCA+CDA extracted features such that 1.5% (60/4000 dim.) of the feature could achieve the highest classification rates.

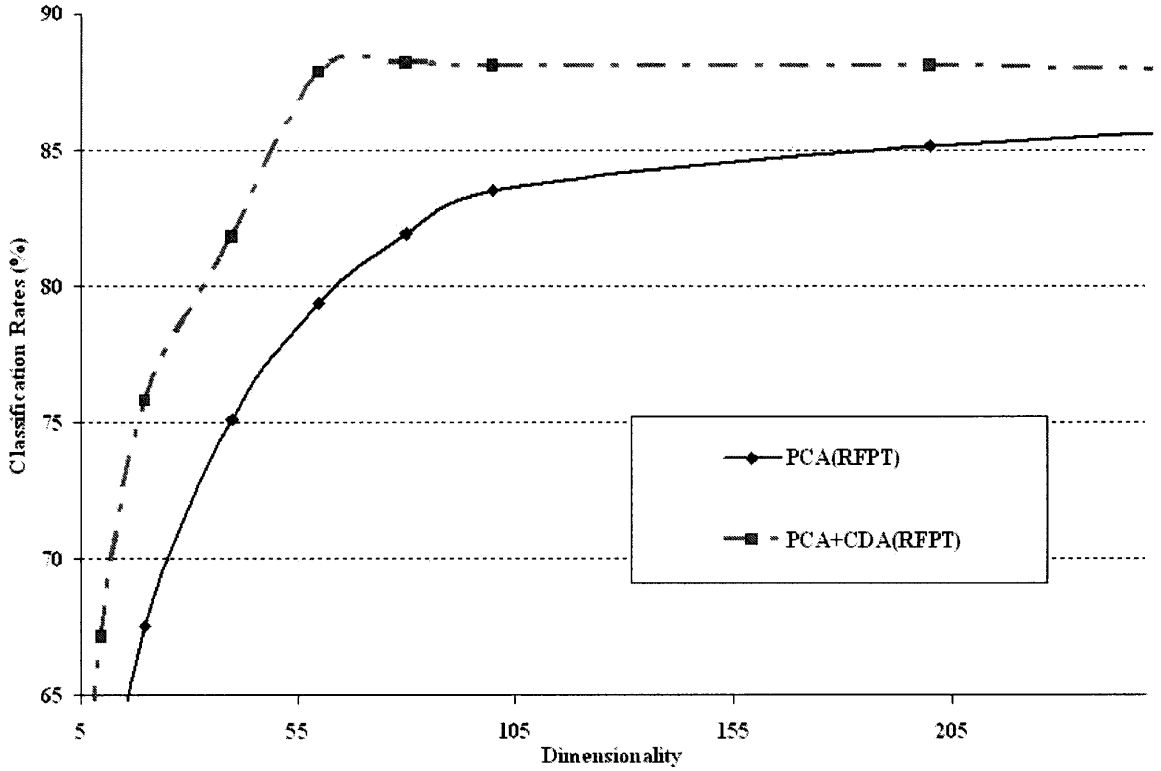


(a) Classification by Euclidean distance on AF

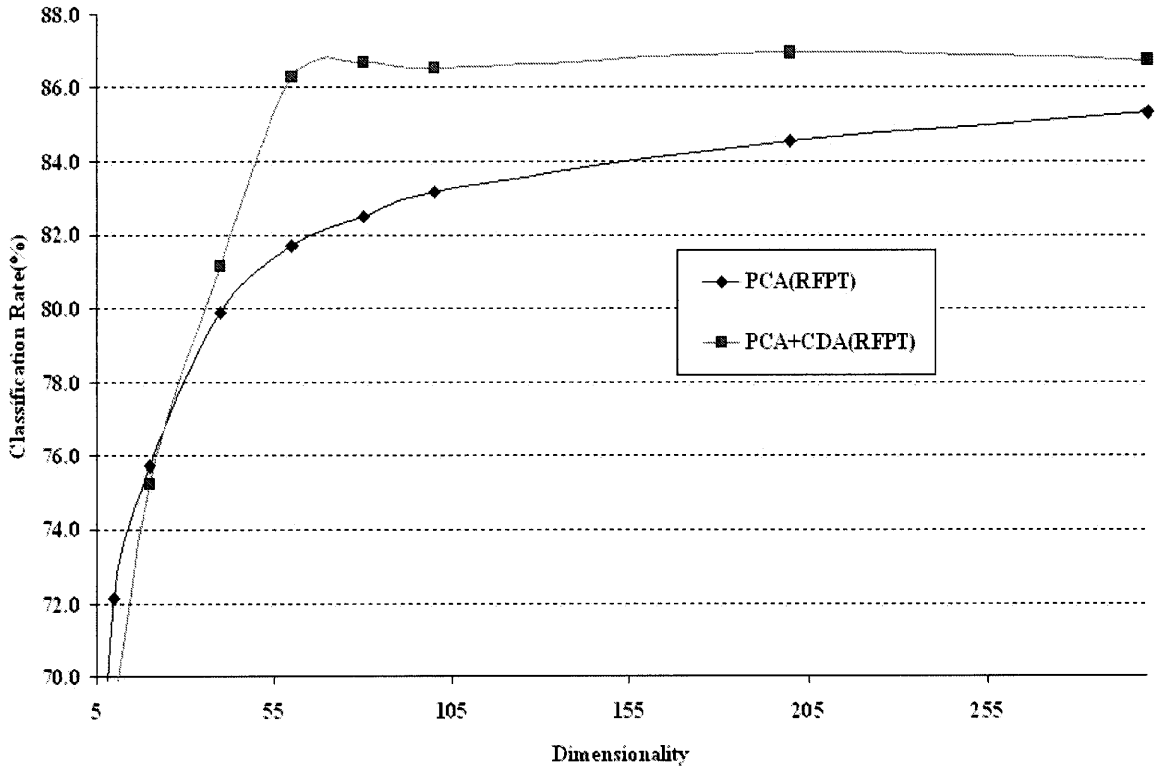


(b) Classification by Euclidean distance on RFPT

Figure 4.1: PCA+CDA effect on randomly selected sample of 10 categories from Reuters-21578 using Euclidean distance classifier



(a) Classification by Modified projection distance on RFPT



(b) Classification by linear SVM on RFPT

Figure 4.2: PCA+CDA effect on randomly selected sample of 10 categories from Reuters-21578 using modified projection distance and linear SVM as learning methods.

4.4.3 Experiments on Published Splits of Data Sets

This section describes the implementation issues of published splits of data sets. Specifically, we briefly describe the data for experiments, the adopted feature selection methods, the techniques used for dimensionality reduction, the classification process and the performance measures that we applied.

We used two popular data sets in the experiments. These are the Reuters-21578 and OHSUMED data collections. ModApte Split was used in the case of Reuters-21578. Heart diseases tree (HD-119) was used in the case of OHSUMED. The details of these data sets are described in 3.5. The vocabulary list was generated as explained in 3.5.2.

As described in section 3, PCA was used to reduce the dimensionality. We then applied CDA to the appropriate amount of principal components. We chose the dimensionality before application of CDA experimentally.

The learning methods used include kNN, linear and polynomial SVMs. We adopted the recall, precision and F-measure for performance evaluation of classification effectiveness. These measures are regarded as standard evaluation methods for classification systems in automatic text classification. The definitions of these measures can be found in 2.5.1. Micro-averaging and macro-averaging strategies are usually adopted. For comparability with other previous works in the literature we adopted micro-averaging and report F-measure scores.

4.4.4 The Effect of PCA+CDA Algorithm on Published Splits

In this subsection, we present the effect of the PCA+CDA algorithm based on the experiments. We included other dimensional reduction techniques for comparability. These include the mutual information method (MI), principal component analysis (PCA), canonical discriminant analysis (CDA) and the PCA+CDA algorithm.

Let us begin our discussion by considering the aspect of Fisher's ratio for techniques such as PCA and CDA. Fisher's ratio is a measure of the discriminating power of variables from various classes (groups). The higher the ratio the higher the separability of the samples. Fig. 4.3 plots the Fisher's ratios (a.k.a F ratio or variance ratios) of principal components (PCs) and canonical discriminants (CDs) from the training documents. This figure shows that CDs have significantly better separability than PCs. It is shown that RFPT have the highest F ratio implying better classification effectiveness than that of TFIDF.

Before we proceed to describe the results, it is good to note that, unless otherwise stated, CDA in this section is applied to 1000 principal components (PC) of RFPT or TFIDF instead of direct application to original features. The 1000 PCs were chosen because the dimensionality is computationally reasonable, and yet sufficiently high for feature representation.

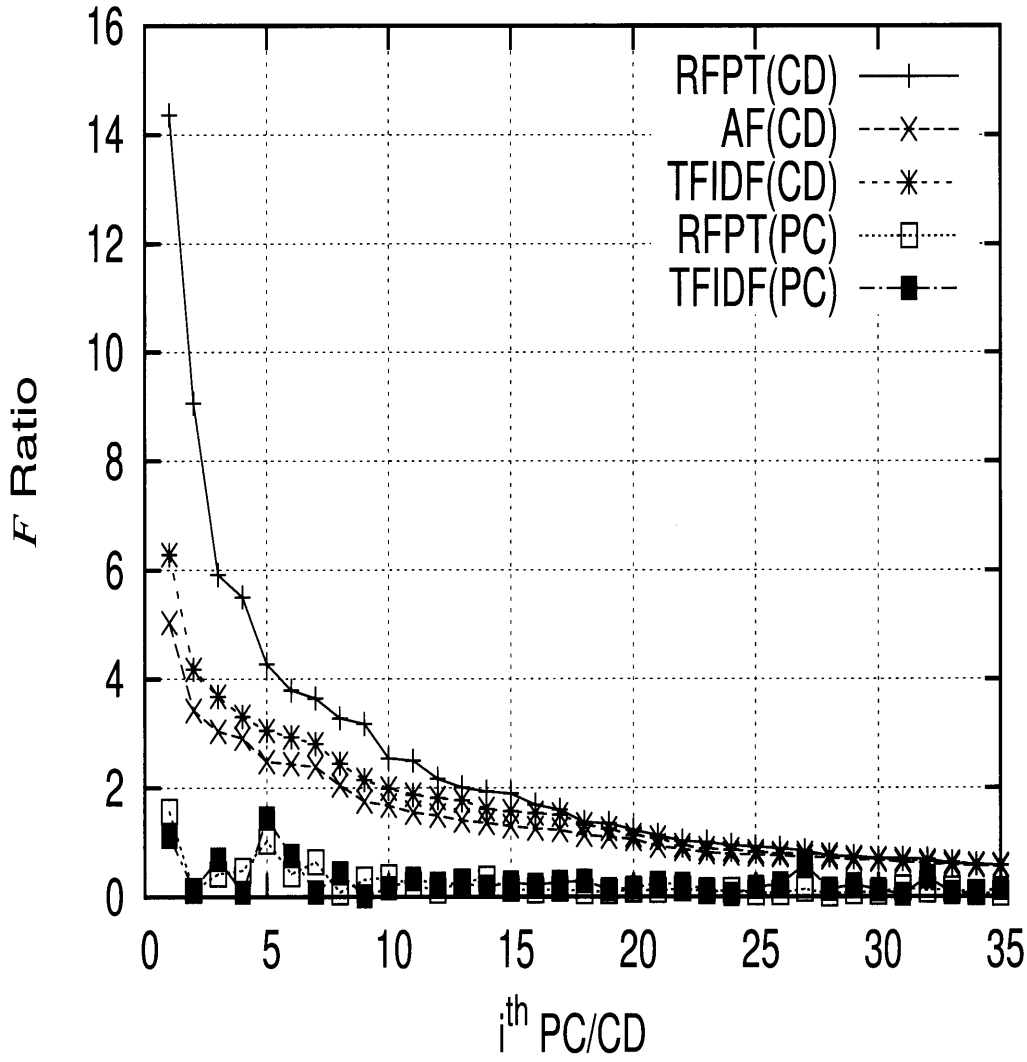
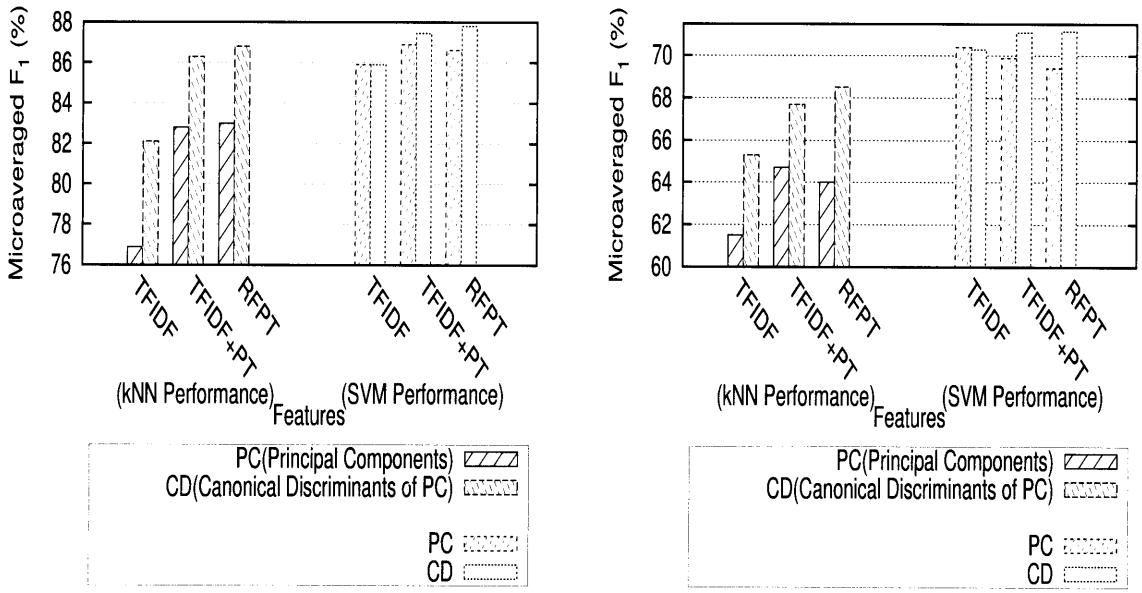


Figure 4.3: F ratio (a.k.a Fisher's ratio or simply variance ratio) of PC/CDs. AF (absolute term frequency), RFPT (relative frequency with power transformation), and TFIDF (term frequency weighted by inverse document frequency). The higher the F ratio the higher the separability. The ratios were obtained from ModApte split of Reuters-21578.



(a) Reuters-21578 data set

(b) OHSUMED data set

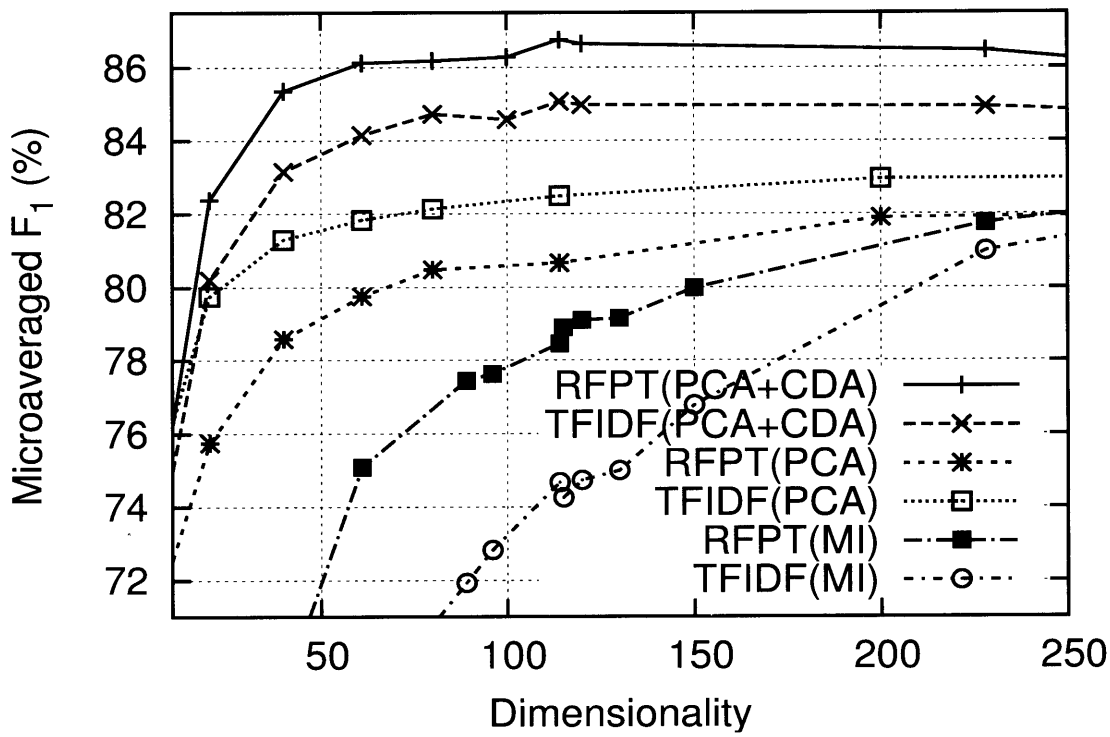
Figure 4.4: The effect of combined dimension reduction (PCA+CDA). Example of features compared include power transformed relative frequency (RFPT), term frequency weighted by inverse document frequency(TFIDF), and power transformed TFIDF (TFIDF+PT)

In Fig. 4.4, it can be revealed that PCA+CDA considerably improves the classification performance. Although TFIDF responded positively to canonical discriminant analysis, RFPT generally gave better classification performance than TFIDF. This is even obvious with the kNN classifier on the Reuters data set. The difference for SVM is smaller than that of the kNN classifier. This is possibly because SVM is closer to the Bayes optimal classifier for this data set than kNN.

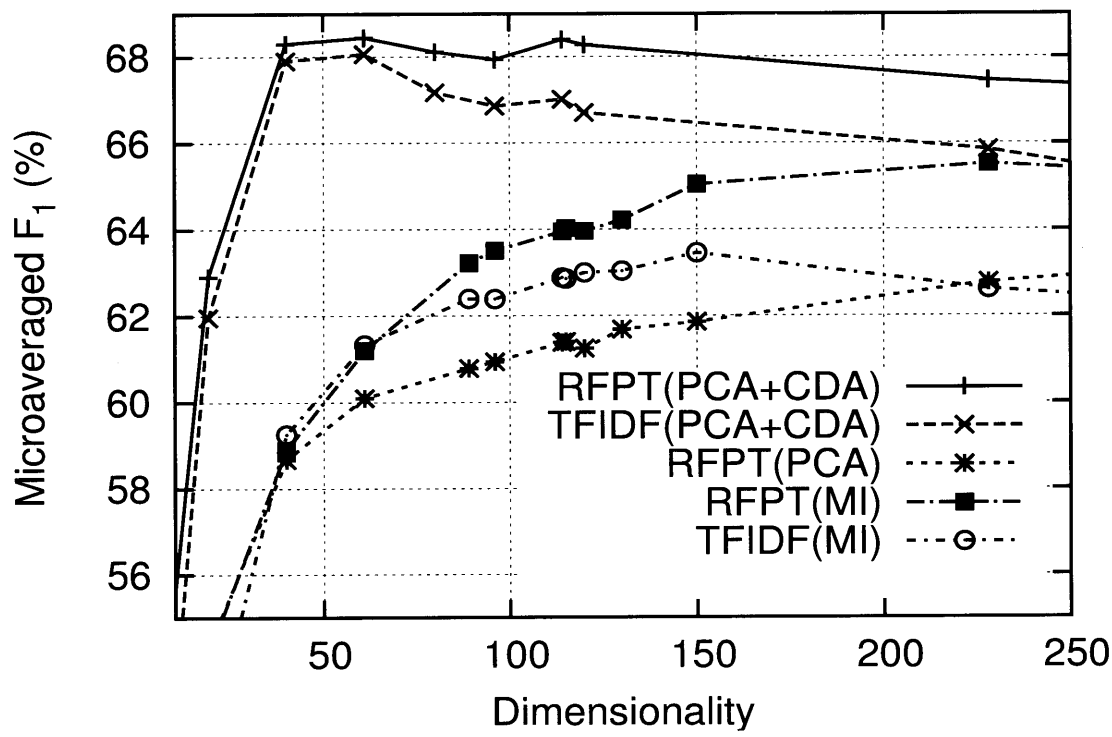
It is worth mentioning that direct application of CDA to original RFPT of 7474 dimensionality achieved 76.4% (micro-averaged F_1) with kNN on Reuters. This is significantly low than the application of PCA+CDA which achieved micro-averaged F_1=86.8%. This is because of rank deficiency and numerical instability problems emanating from inadequate sample size per dimensionality. This problem occurs when the sample size is smaller than the size of the feature vector.

The Figures 4.5, 4.6, and 4.7 give the relationship between dimensionality and micro-averaged F_1 or F-measure. The term dimensionality refers to the amount of features used. Comparisons of various features and classifiers are given. Similarly it is clear that RFPT performs considerably better than TFIDF.

PCA+CDA outperforms all compared methods in this case. This is especially better at lower dimensionality. This means PCA+CDA algorithm can extract fewer features with higher discriminatory information. When kNN (Fig. 4.5) and polynomial SVM

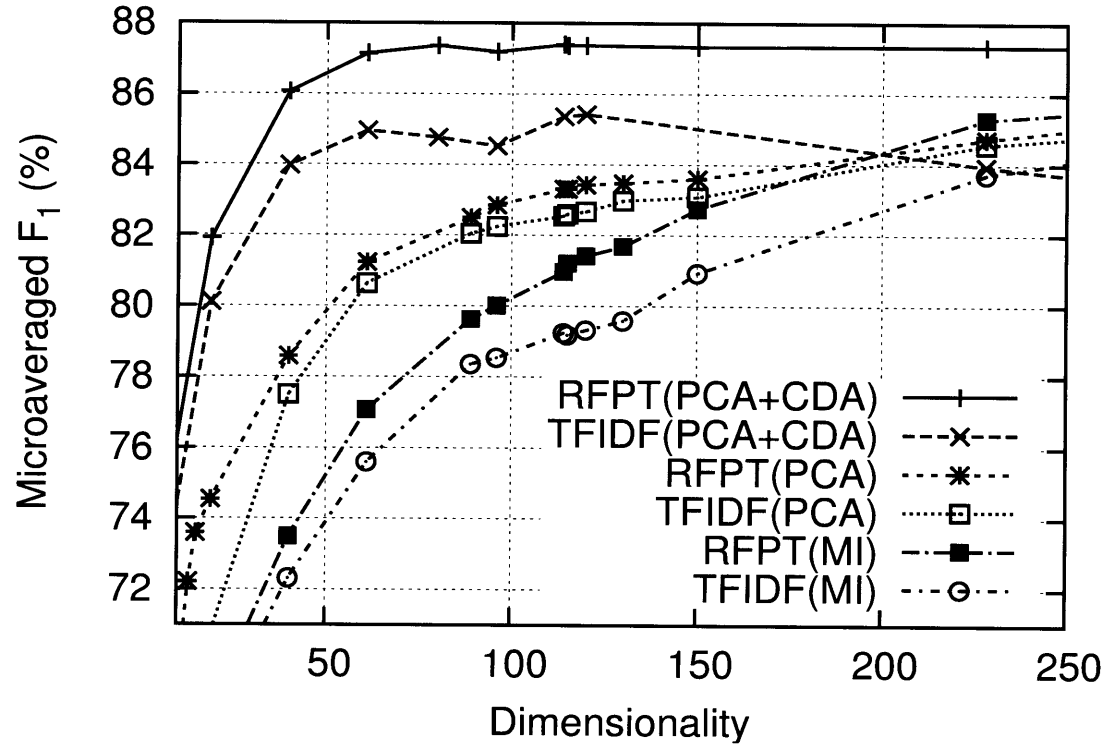


(a) kNN on ModApte split of Reuters-21578

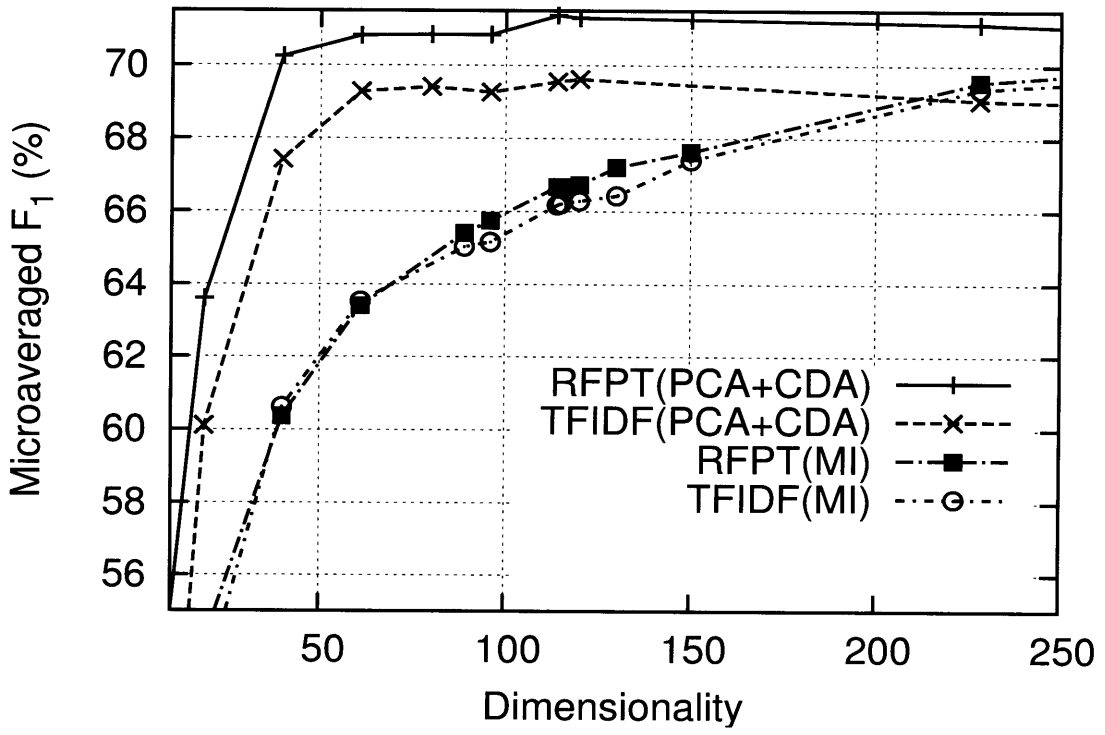


(b) kNN on HD-119 tree of OHSUMED

Figure 4.5: The effect of PCA+PCA algorithm on published splits when kNN learning method is used.



(a) Polynomial SVM on ModApte split of Reuters-21578



(b) Polynomial SVM on HD-119 tree of OHSUMED

Figure 4.6: The effect of PCA+PCA algorithm on published splits when polynomial SVM is used.

(Fig 4.6) are used, the differences with the conventional methods for feature reduction are higher at lower dimensionality.

Furthermore, it is noted that even when high amount of features were added above the optimum level there was no performance improvement. From section 4.3.1, it can be recalled that CDA algorithm produces $C-1$ nonzero generalized eigenvalues, leading to equivalent dimensional space that gives highest performance.

Before proceeding to the next subsection, it is worth noting that the mutual information (MI) method gives lower performances than its counterparts. This is even obvious at lower dimensionality, or when relatively fewer features are used. The main reason is that the features selected by the MI method are mutually correlated, while PCs and CDs are uncorrelated.

4.4.5 Classifier's Efficiency Improvements by PCA+CDA

Nonparametric classifiers such as kNN may be slower if subjected to high dimensionality, which is always the case in automatic text classification. However, the combined dimensionality reduction improved classifier's efficiency with improved classification performance.

Table 4.1: kNN classifier's efficiency on Reuters (time in milliseconds per text), LT = linear transformation

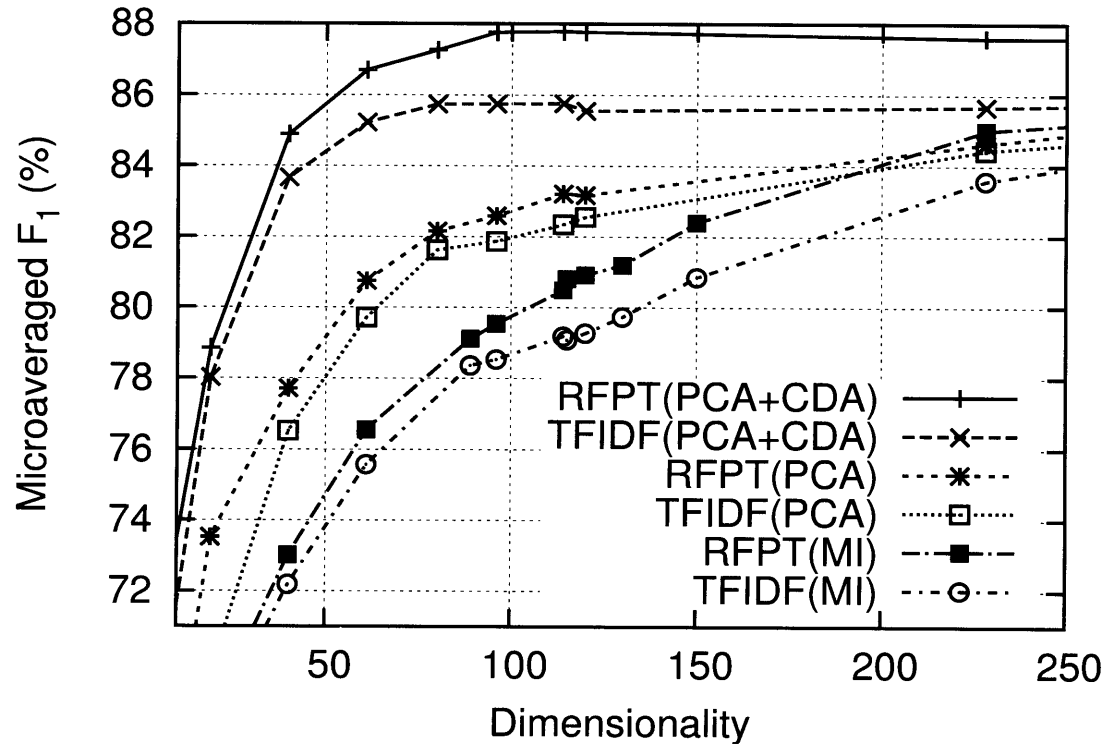
	Dimensionality	LT Time	kNN Time	Total Time
All Features	7474	-	708.3	708.3
PCA	1000	92.4	110.0	202.4
CDA	114	14.7	31.6	46.3

Table 4.1 summarizes the time used for classification at different dimensionality. The linear transformation column represents the time used to perform linear transformation defined by equation (4.14). It can be seen that kNN time was reduced from 708.3 to 31.6 milliseconds per text, which is about 22 times less than using all words. Similarly the total time was reduced from 708.3 to 46.3 milliseconds per text, which is about 15 times less. The encouraging thing is that this efficiency improvement goes along with better classification performance.

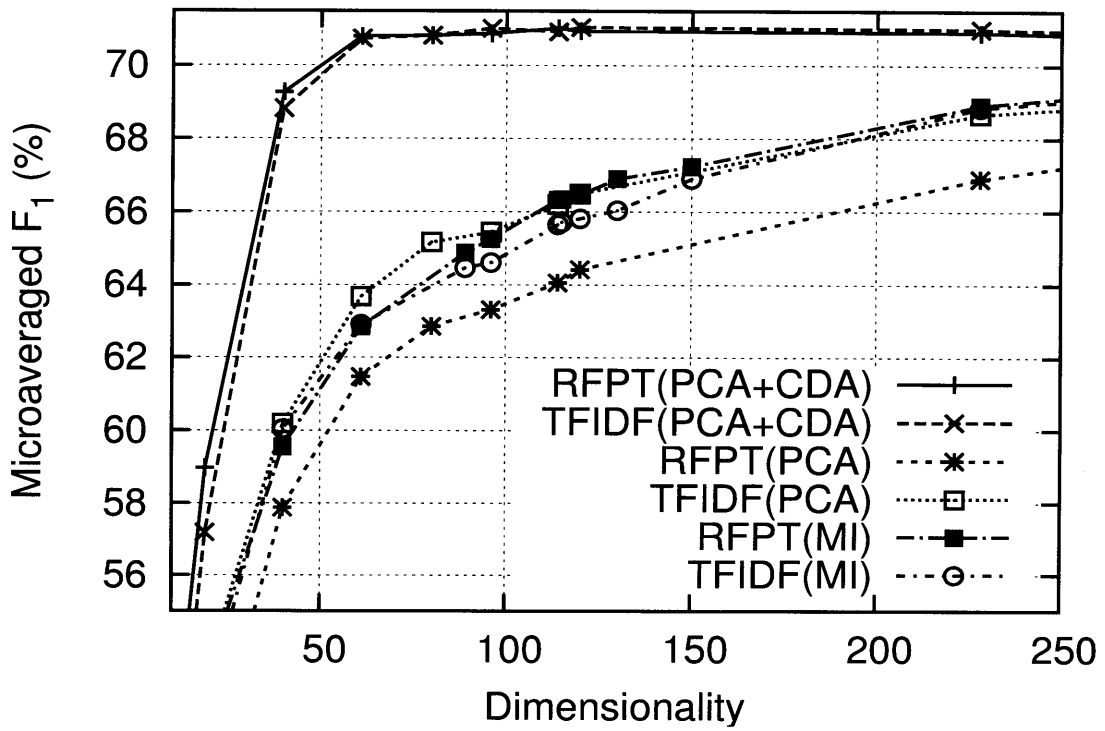
4.5 IDA Experiments and Results

In order to verify the performance of integrated discriminant analysis (IDA), we used the published splits of data sets.

For comparison reasons we included results of PCA+CDA which are presented in 4.4.4. Results for regularized discriminant analysis (RDA) are also included in this section.



(a) Linear SVM on ModApte split of Reuters-21578



(b) Linear SVM on HD-119 tree of OHSUMED

Figure 4.7: The effect of PCA+PCA algorithm on published splits when linear SVM learning method is used

The classifiers included kNN, support vector machines using linear and polynomial kernels. All these classifiers are described in chapter 2.

The performance measures adopted here are recall, precision and F_1-measure. The definitions of every method are presented in Section 2.5.

4.5.1 Experiments on Published Splits of Data

We used two popular data sets in our experiments. These are Reuters-21578 and OHSUMED data collections. We used the ModApte Split of Reuters-21578 and Heart Disease Tree of 119 Medical Subject Heading (MeSH). The details of these data sets are described in 3.5. The vocabulary list was generated as explained in 3.5.2.

4.5.2 Effect of IDA on Published Splits of Data

Figures 4.8, 4.9 and 4.10 show the relation between dimensionality and F_1-measure. In these figures only the well optimized results are reported.

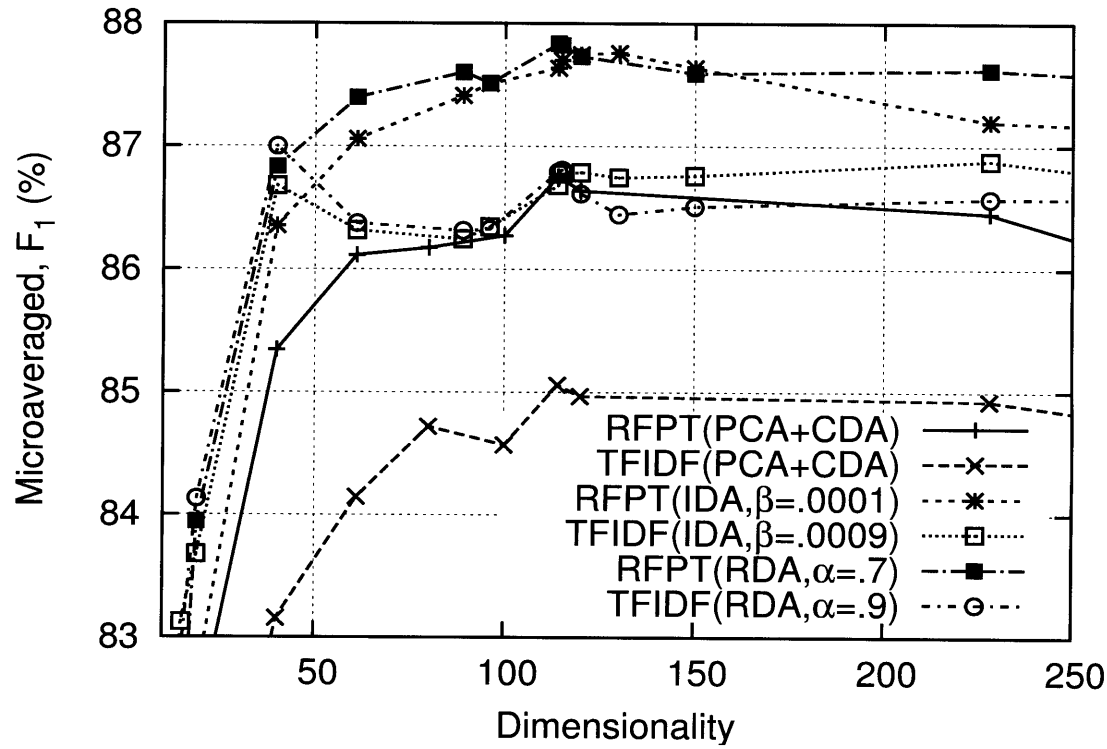
It can be observed that when kNN is used in Fig. 4.8, IDA improves the performance of this classifier. For both data sets IDA achieved the highest classification performance especially at the range of features between 100 and 150. It is interesting also to note that RFPT generally outperformed TFIDF.

A closer perusal of Fig. 4.9 reveals that IDA outperformed other methods for features reduction. The differences with other methods is even significant in Fig. 4.9(b) when the OHSUMED data set is used.

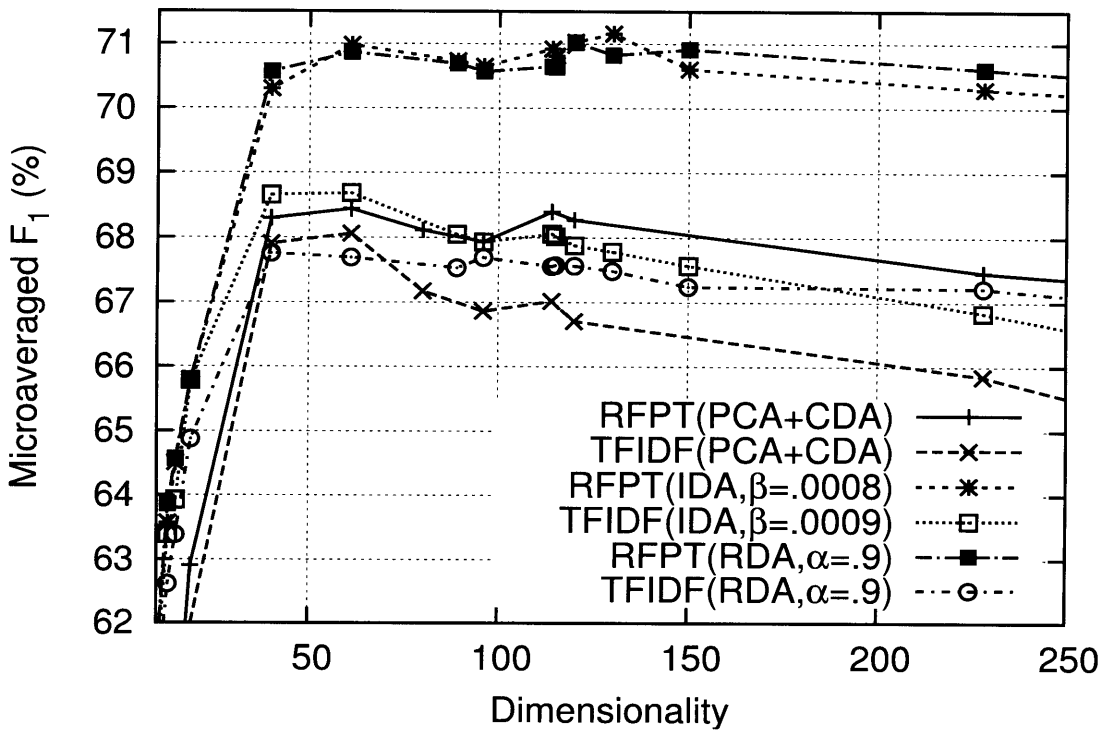
A similar trend is seen in Fig. 4.10, linear SVM is also favored by IDA. This is particularly true with the Reuters-21578 data set. The interesting point that can be seen in figures 4.10(b) and 4.10(a) is when IDA outperformed the other feature reduction methods, especially when RFPT is used. It can be observed that IDA on RFPT performed best, especially when more than 114 features are used. This demonstrates the fact that IDA is not limited to $C-1$ features. In contrary CDA and RDA are limited to $C-1$ features, as demonstrated by the fall of performance after $C-1$ features.

The reasons for the improved performance by IDA are described in Section 4.3.2. IDA simultaneously optimizes the variance ratio of CDA and the mean square error of PCA. In so doing it extracts features with richer discriminatory information. Furthermore it does not suffer from the singularity problem due to smaller sample size than its dimensionality. In addition it is not limited to $C-1$ classes of data. This makes it more practical than RDA.

Based on these empirical results, which are plotted in figures 4.8 to 4.10, it can be said that IDA is effective in automated text classification (ATC). It can extract optimal features with high discriminating power leading to improved classification performance in

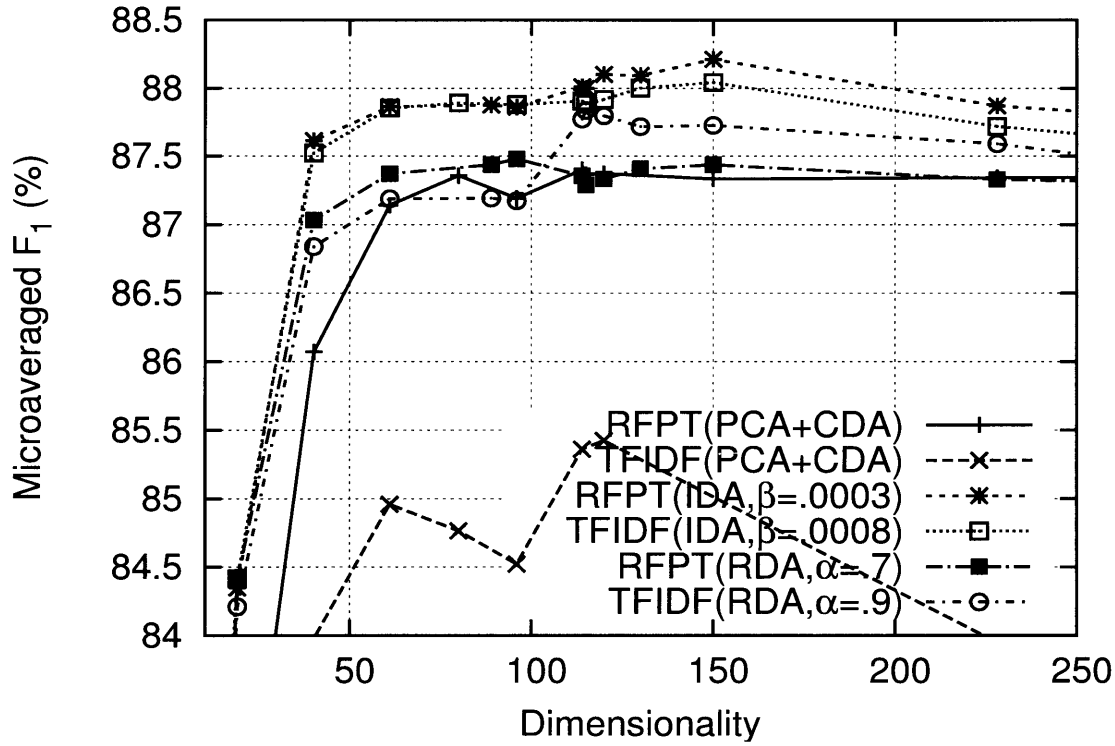


(a) kNN on ModApte split of Reuters

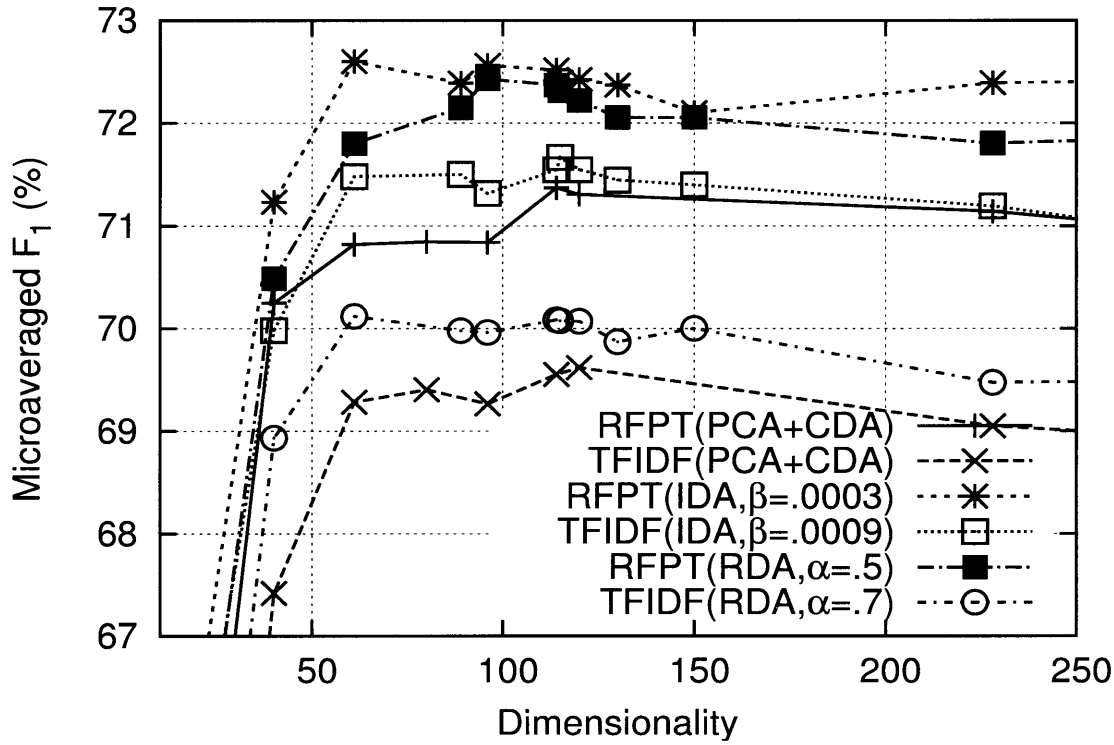


(b) kNN on HD-119 tree of OHSUMED

Figure 4.8: The effect of integrated discriminant analysis (IDA) on dimensionality reduction in comparison with other conventional methods. Features include relative frequency with power transformation (RFPT), term frequency weighted by inverse document frequency (TFIDF).

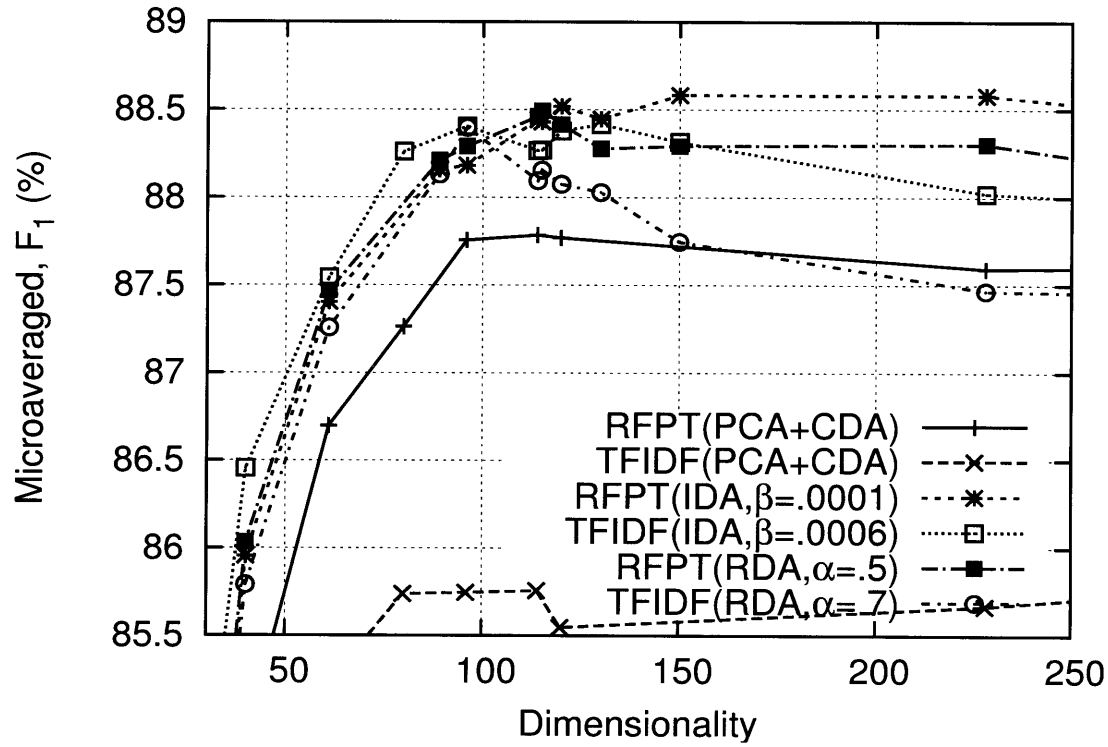


(a) Polynomial SVM on ModApte split of Reuters

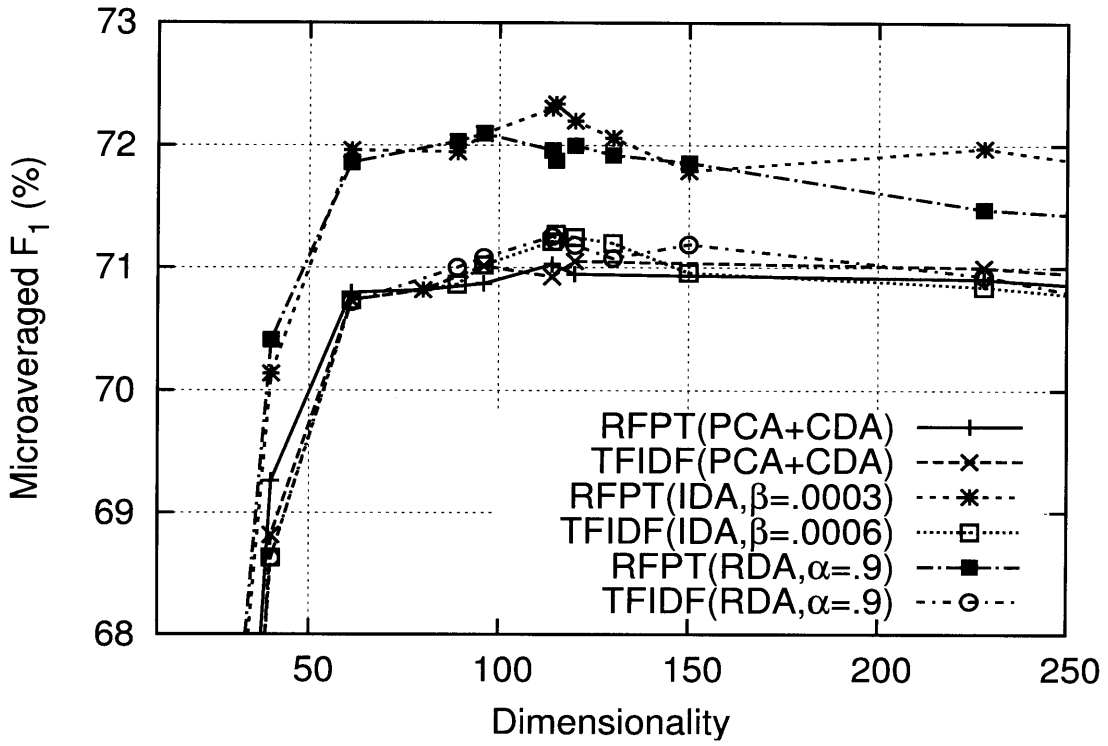


(b) Polynomial SVM on HD-119 tree of OHSUMED

Figure 4.9: The effect of integrated discriminant analysis (IDA) on dimensionality reduction in comparison with other conventional methods. Features include relative frequency with power transformation (RFPT), term frequency weighted by inverse document frequency (TFIDF).



(a) Linear SVM on ModApte Split of Reuters-21578



(b) Linear SVM on HD-119 Tree of OHSUMED

Figure 4.10: The effect of integrated discriminant analysis (IDA) on dimensionality reduction in comparison with other conventional methods. Features include relative frequency with power transformation (RFPT), term frequency weighted by inverse document frequency (TFIDF).

ATC. By extracting fewer features, the classifier's efficiency improvement can be realized similarly to what is discussed in 4.4.5.

4.6 Summary of Feature Reduction

This chapter mainly deals with feature reduction. It starts by reviewing the conventional methods used in text classification which are mainly based on feature selection. While these methods have been widely applied to text classification, feature extraction methods are sparsely applied.

This chapter proposes methods for feature extraction to reduce the dimensionality. Various experiments on different data sets are presented. These include randomly selected data which are experimented using the 3-fold cross-validation technique. Furthermore published splits by other researchers were used in the experiments. These are relatively large samples with many categories reflecting the real world problems.

Feature extraction methods studied include the combination of principal component analysis and canonical discriminant analysis (PCA+CDA). The studied dimensional reduction approach drastically reduced the dimensionality. This goes along with improved classification performance.

Furthermore, integrated discriminant analysis (IDA) is proposed in this chapter. Empirical results show that IDA is suitable and practical for text classification. This implies that higher classification performance can be achieved even at lower dimensionality. This is realized with improved classifier's efficiency in terms of learning and classification speed. Future research includes employing more samples.

Chapter 5

Feature Integration and Ensembles

5.1 Introduction

The objective of feature integration is to seek to represent documents appropriately in a machine learning system. It seeks to obtain a more informative representation that can enhance the discriminating power by the learning algorithms on textual data.

In this chapter we propose a technique for feature integration by incorporating pre-processed, transformed and reduced features. This integration generates composite features that have enhanced discriminating power. The integration performed in this study involves transformed and term weighted features.

The aim of transforming the features before integration is to avoid text length variation and asymmetry of sample distribution. Therefore, transformed features can be more informative on class separability. On the other hand, it is seen that term weighting is applied by using inverse document frequency to the other set of features for integration. The essence of term weighting is to deemphasize the terms that appear in most of the documents. This is because terms that appear in most documents may mask discriminative information for classification. Term weighted vectors are assumed not to suffer from the inadequacy of discriminative information.

Feature reduction using techniques such as principal component analysis (PCA) is carried out. The reason for using feature reduction is to obtain a more manageable amount of features with high discriminating power. After the integration, discriminant analysis (DA) methods can be applied for two reasons. First of all, DAs unify the discriminative information shared by the integrated features. In so doing, extracting a more informative representation of data. Secondly, DAs simultaneously extract fewer features which improve classification effectiveness and the computational speed (efficiency).

This chapter also proposes multiple feature-classifier combination (MFC). In the conventional method of multiple classifier combination, only one type of feature is used. Unlike the conventional way of classifier combination, in this technique, various types

of features are separately fed to different classifiers. Then the classification algorithms are combined to improve the classification effectiveness. The classifier decisions were combined by the use of the majority vote function.

The rest of this chapter is organized as follows. Section 5.2.1 describes the proposed feature integration. Section 5.2.2 introduces the details on MFC. Section 5.3 presents experiments conducted to investigate the applicability of the proposed methods. Section 5.4 describes the empirical results. Section 5.5 gives the concluding remarks.

5.2 Methodology

5.2.1 Classification Approach with Feature Integration (FI)

The general steps for the approach that incorporates FI are summarized in Fig. 5.1. First we generate feature vectors which represent the documents as described in 3.3. The second step consists of feature transformation which include transforming absolute term frequency to relative term frequency (RF) and power transformation (PT). This is especially for step 2(a) in Fig. 5.1. Step 2(b) represents term weighting process given in equation (3.2). Step 3(a) and 3(b) depict dimensionality reduction by using principal component analysis (PCA) (see Section 4.14). They are separately shown because they are applied to different types of features.

Step 4 is for the novel technique for feature integration which is proposed in this work. Details are given in this Chapter. After this integration, canonical discriminant analysis (CDA) or any other discriminant analysis methods can be performed in step 5. Learning is carried out in step 6. Finally, classification is executed in step 7.

In our method, feature integration (FI) can be defined as the combination of various features to generate composite features that give higher classification effectiveness. The objective for FI is to compose more informative features by extracting discriminative information from the transformed and/or weighted features. Consequently, it improves the separability of the between-class documents. The integration studied here include use of the concept of concatenation. Therefore this technique can be known as feature integration by concatenation (FIC). Improved separability of class documents can be achieved by suitably integrating reduced transformed features (RFPT) and the term weighted features (TFIDF).

Class document separability is improved because the integrated composite features receive the discriminative information from both RFPT and TFIDF. Particularly, RFPT does not depend on document length, and its variable distributions have the properties of Gaussian-like distribution. The reader can recall from 3.3 that Gaussian-like distributions reduce classification errors yielded by non-optimal classifiers. The TFIDF discriminative information emanate from the assignment of high degree of importance to terms that

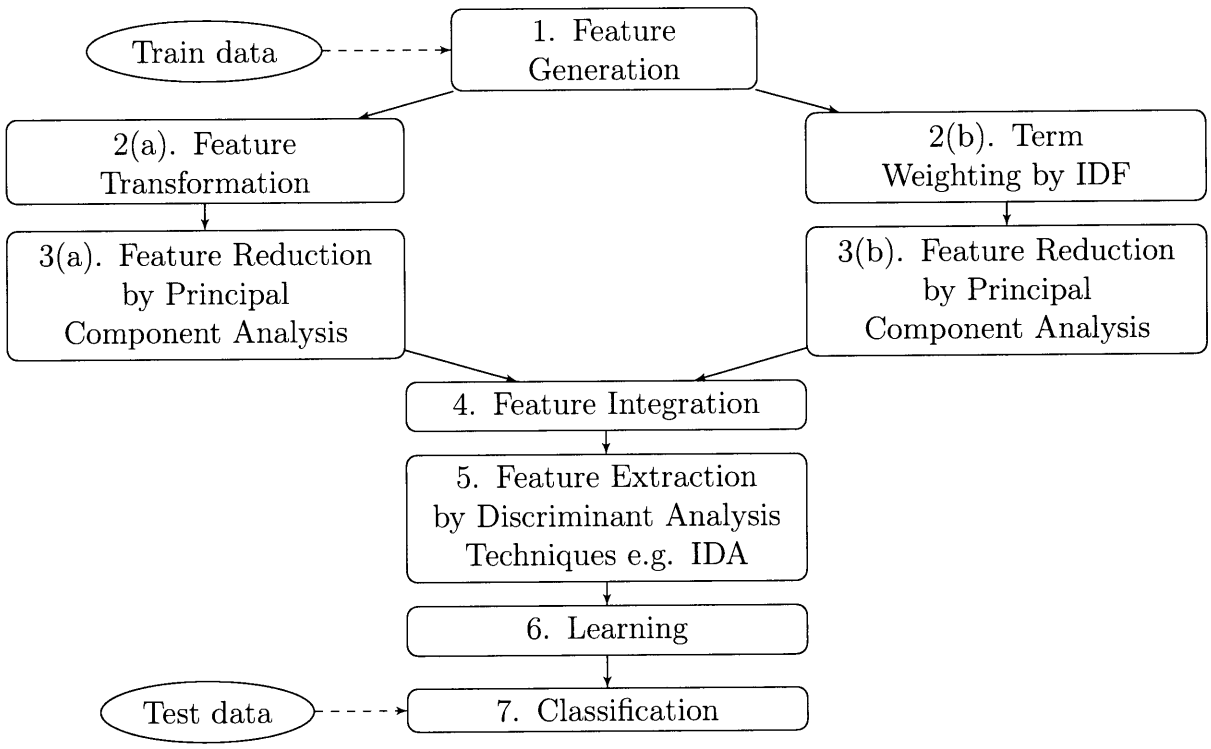


Figure 5.1: Algorithm for automated text classification with feature integration. IDF is the abbreviation for inverse document frequency.

occur in only few documents of the text data set.

Let us assume that there are two feature vectors for integration: (1) the term frequency weighted by inverse document frequency (TFIDF) and (2) the transformed features, called relative frequency with power transformation (RFPT). For simplicity, we denote RFPT with a superscript t and TFIDF with a superscript w in equations 5.1 – 5.4. Consequently, we can define such feature vectors as

$$\mathbf{x}^{(t)} = \left[x_1^{(t)}\ x_2^{(t)}\ \ldots\ x_n^{(t)}\right]^T, \tag{5.1}$$

for the RFPT features. The TFIDF features can be expressed as

$$\mathbf{x}^{(w)} = \left[x_1^{(w)}\ x_2^{(w)}\ \cdots\ x_n^{(w)}\right]^T, \tag{5.2}$$

and the concatenation of these document features can be defined as

$$I = \mathbf{x}^{(t)} \oplus \mathbf{x}^{(w)} \tag{5.3}$$

$$= \left[x_1^{(t)}\ x_2^{(w)}\ \ldots x_n^{(t)}\ x_{n)}^{(w)}\right]^T. \tag{5.4}$$

We use this proposed technique for FI to generate composite features to improve classification performance. We show the effect of FI in Section 5.4.

There are various added advantages of the text classification with feature integration approach. These can also be regarded as the reasons for the improved classification effectiveness.

Firstly, FI uses transformed features which do not depend on textual length, as a result, the within-class variability can be avoided. Secondly, as it is described in Section 3.3, transformed features provide a Gaussian-like sample distribution which makes it easier to find better decision boundary by the classification system. Thirdly, the singularity problem due to size of the sample being smaller than its dimensionality is solved at PCA stage.

The fourth advantage can be realized by improving separability by applying discriminant analysis techniques. This is applied to the dominant principal components rather than directly to the original features. Therefore, numerical stability and classifier's efficiency can be realized. The fifth advantage is that FI can incorporate new features other than those from TFIDF and RFPT. In the end, we obtain improved classification performance. In short, it can be said that the limitations of each technique can be avoided by using FI.

5.2.2 Multiple Feature-Classifier Combination (MFC)

Multiple classifier combination (MCC) (also known as ensemble) has been reported to improve text classification. There are various combination functions. The simplest one is by majority vote (MVR). An odd number of classifiers is commonly used to avoid ties in the number of votes [82]. Conventionally, MCC is carried out by getting decisions from one type of features.

Unlike the conventional way of combining the classifier decisions, we used the proposed RFPT and the conventional features, namely TFIDF, and combined classifiers' decisions from these features, thus the name multiple feature-classifier combination (MFC). Figure 5.2 illustrates the classification procedure that makes use of the algorithm with multiple feature-classifier combination.

In the experiments, the best three performers out of the classifiers were used. These are linear SVM, kNN and polynomial SVM. In the first case which is abbreviated to MFC3, we carry out the feature-classifier combination as follows:

(i). Linear SVM's decisions from RFPT

(ii). Polynomial SVM's decisions from RFPT

(iii). Linear SVM's decision from TFIDF

In the second case which is abbreviated to MFC5, the feature-classifier combination is formed as follows:

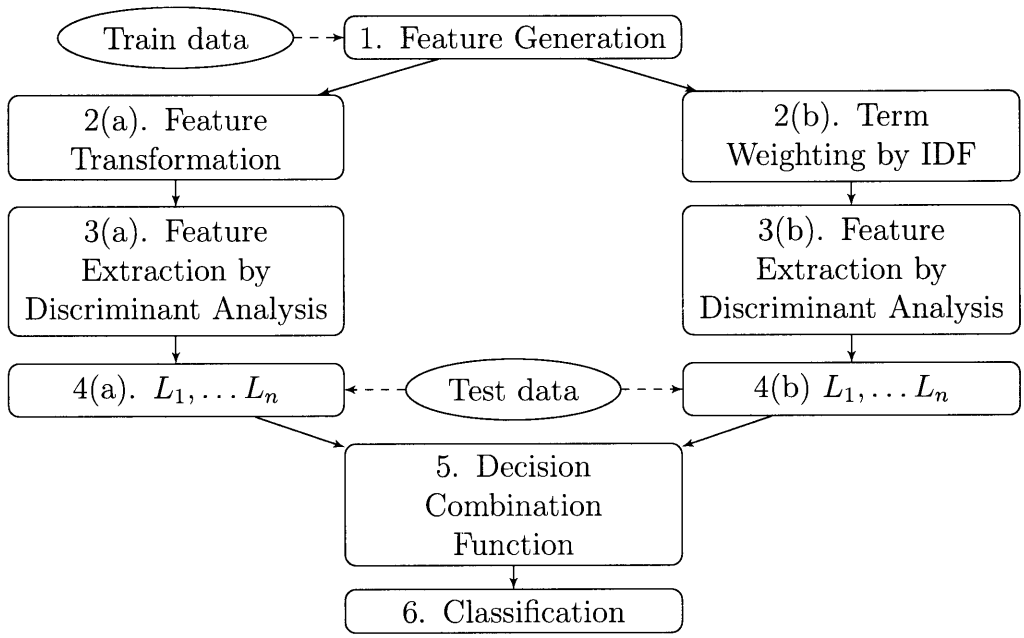


Figure 5.2: The automated text classification algorithm for multiple feature-classifier combination (MFC). DA refers to discriminant analysis techniques proposed in this work. Examples of the DA techniques are the integrated discriminant analysis (IDA) and the regularized discriminant analysis (RDA). L_i refers to the learning methods for classification which are trained before the unseen data (test data) can enter the classification algorithm. IDF is the abbreviation for inverse document frequency.

(i). Linear SVM's decisions from RFPT

(ii). Polynomial SVM's decisions from RFPT

(iii). Linear SVM's decisions from TFIDF

(iv). kNN's decisions from RFPT

(v). kNN's decisions from TFIDF

Figure 5.3 illustrates an example on how the experiments were conducted following the the second case (i.e. MFC5). MFC also improved classification performance by outperforming the best performer from single feature type when one classifier was used.

5.3 Experiments

This section describes the implementation issues. Specifically, we briefly describe the data for experiments, the adopted feature selection methods, the techniques used for dimension reduction, the classification process and the performance measures that we applied.

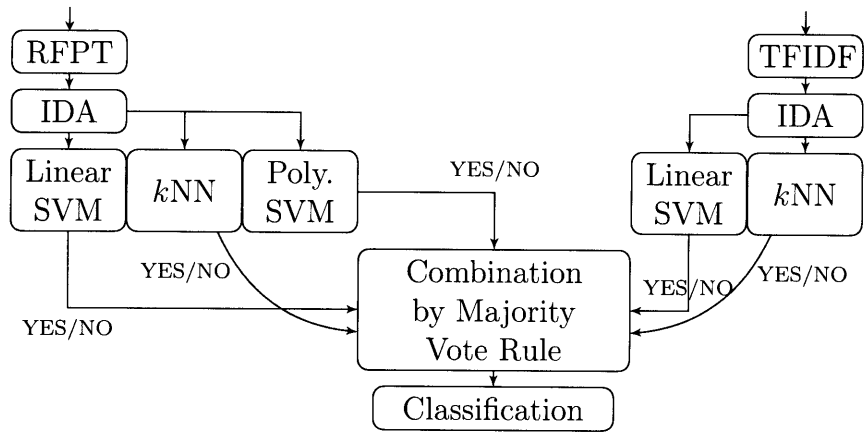


Figure 5.3: Example of the experiments with multiple feature-classifier combination (MFC). This figure illustrates the experiments for MFC5. IDA refers to integrated discriminant analysis. Features include relative frequency with power transformation (RFPT) and term frequency weighted by inverse document frequency (TFIDF).

We used two popular data sets in our experiments. These are the Reuters-21578 and OHSUMED data collections. The details of these data sets are described in 3.5. The vocabulary list was generated as explained in 3.5.2.

The classification procedure with feature integration is generally illustrated in Figure 5.1. As described in 4.2.4, PCA was used to reduce the dimensionality and use fewer principal components. We then applied discriminant analysis techniques such as CDA, IDA or RDA to appropriate amount of principal components. We chose the number of principal components before application of these techniques experimentally.

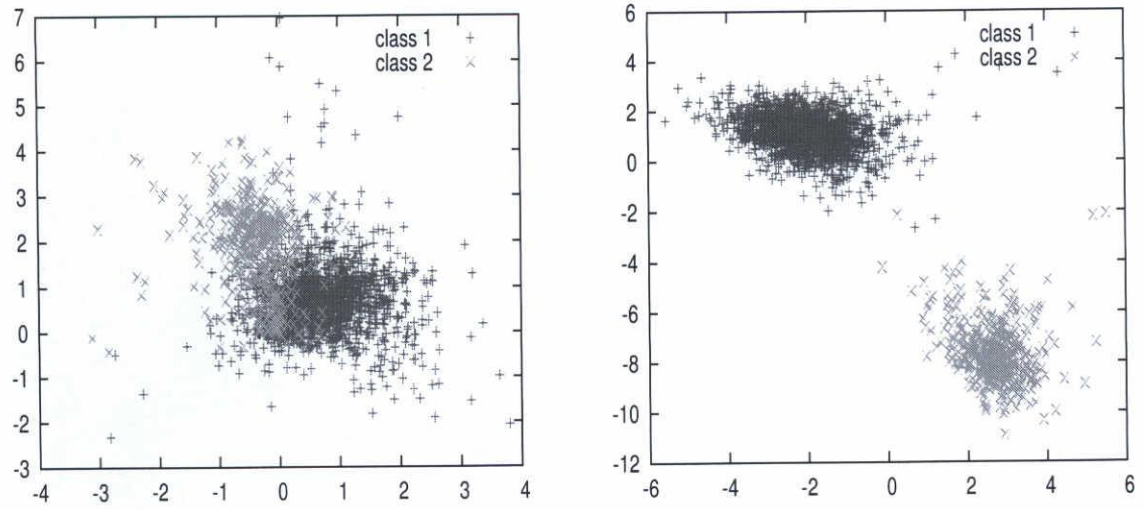
We adopt the recall, precision and F-measure for performance evaluation of classification effectiveness. These measures are regarded as standard evaluation methods for classification systems in automatic text classification. The definitions of these measures can be found in 2.5.1. The Micro-averaging and macro-averaging strategies are usually adopted. For comparability with other previous works in the literature, we adopted micro-averaging and report F-measure scores.

5.4 Empirical Results

5.4.1 Effect of Feature Integration on Classification Effectiveness

First of all, let us investigate the effect of FI by a perusal of an example given in Fig. 5.4. We observe that the data points of all classes in Fig. 5.4(a) are clustered together in feature vector space. Clustering of these data points leads to difficulties in classifying them. While in Fig. 5.4(b) where FI has been applied, the separability is clear such that the decision boundary can be easily determined. Consequently, higher classification performance can be achieved. This indicates that FI improves the separability of the

between-class documents.



(a) TFIDF

(b) Integration of TFIDF and RFPT

Figure 5.4: Class separability for (a) TFIDF and (b) FI. This effect is illustrated from real data used in experiments i.e., Acquisition category (class 1) and Money-fx category (class 2).

Let us continue with our discussion by focusing on Fig. 5.5, 5.6 and 5.7. It is notable that all figures show that the highest classification performance is from the FI features. This means that the proposed technique for feature integration by concatenation (FIC) improved the between-class document separability. The improvement is more obvious when SVM was used. An exception is seen in Fig. 5.5(b) where its effectiveness is uncertain. Since the other five figures indicate improvements, certainly FIC is effective.

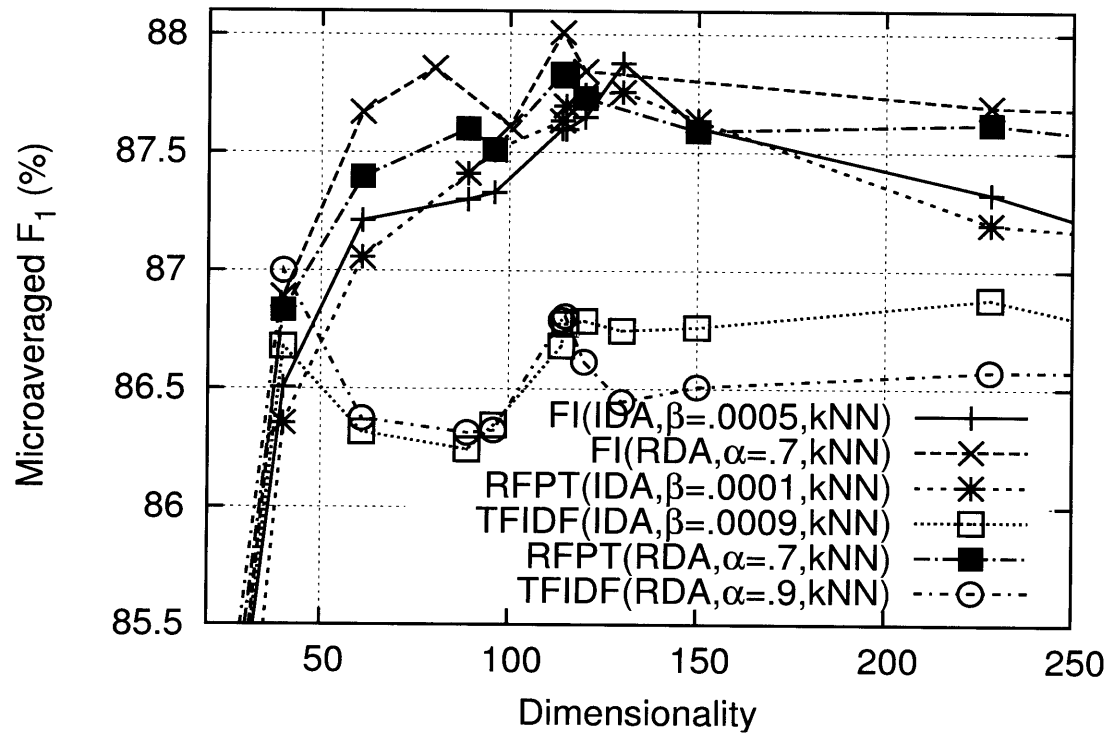
The question on what constitutes the integrated features is described in section 5.2. For the reader's convenience, we shortly describe what constitutes Fig. 5.8. The feature integration (FI) in Fig. 5.8 is from the features, namely TFIDF and RFPT which generated composite features, CF(1) and CF(2). The composite features are defined by

$$CF(1) : CD_{144}(PC_{1000}(TFIDF)) \oplus CD_{114}(PC_{1000}(RFPT)), \tag{5.5}$$

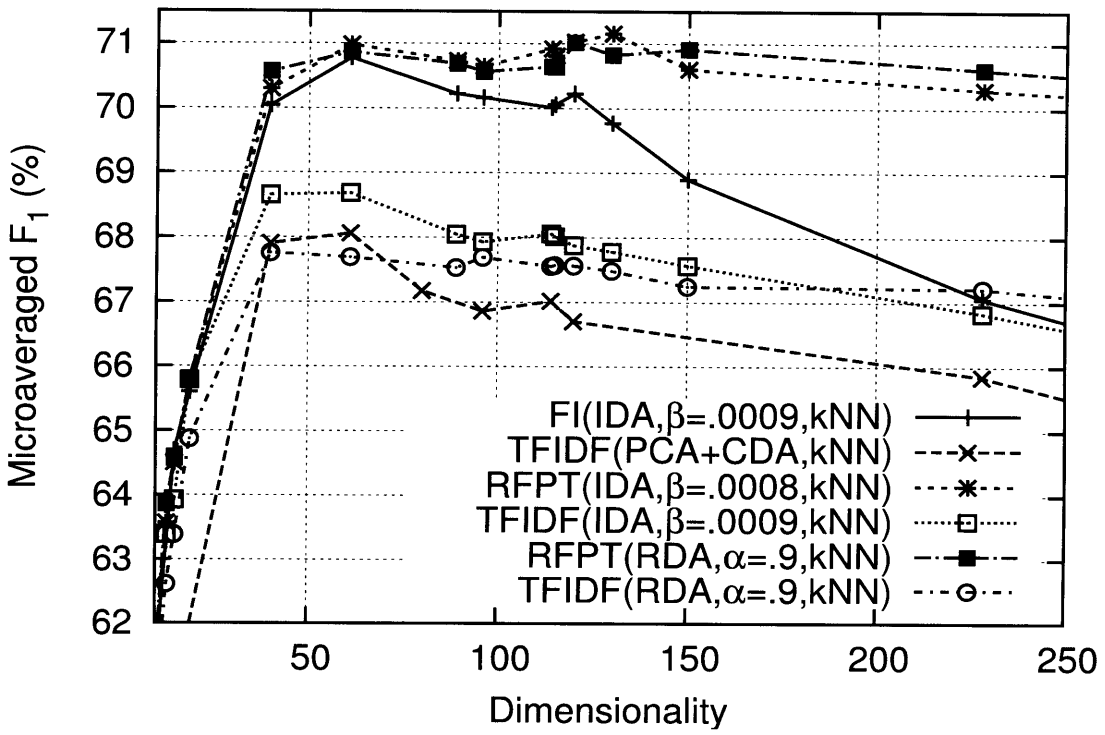
$$CF(2) : CD_{114}(PC_{1000}(TFIDF) \oplus PC_{1000}(RFPT)), \tag{5.6}$$

where $CD_m(\mathbf{x})$ and $PC_m(\mathbf{x})$ denote m discriminants and m principal components of feature $\mathbf{x}$ respectively.

Fig. 5.8 compares the results of TFIDF+PT, RFPT, and those of CF(1) and CF(2). It can be seen that FI improves the classification performance. This confirms our expectation of separability improvement as demonstrated by the classification improvement.

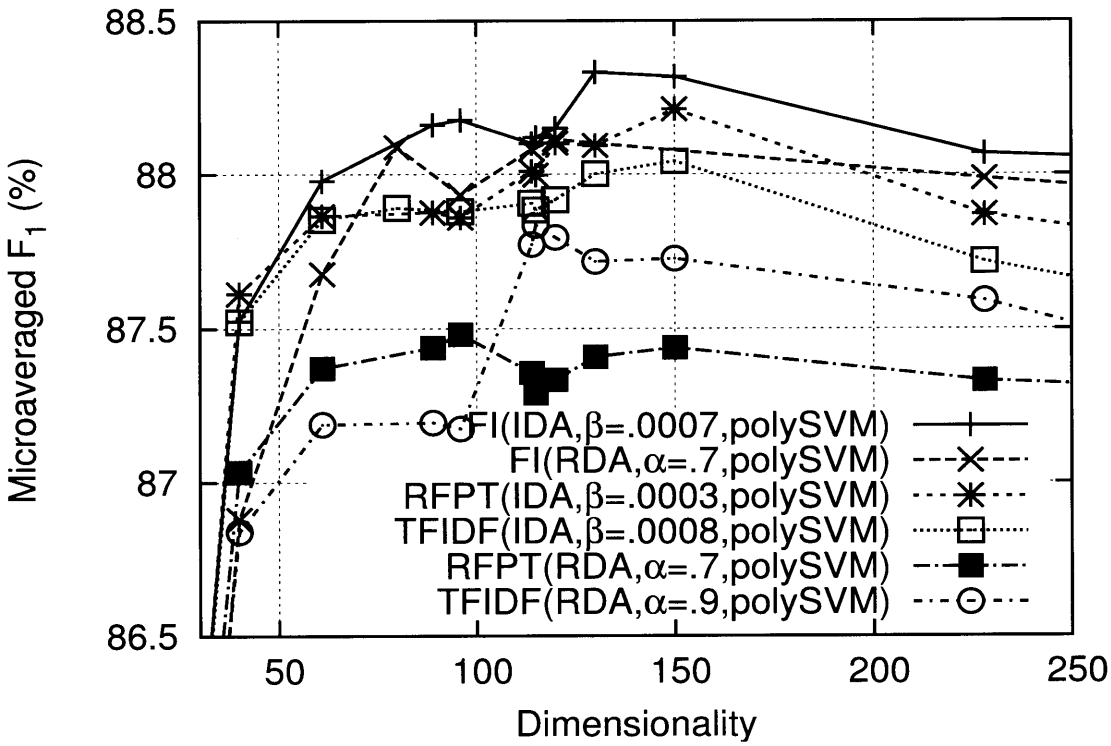


(a) kNN on ModApte split of Reuters

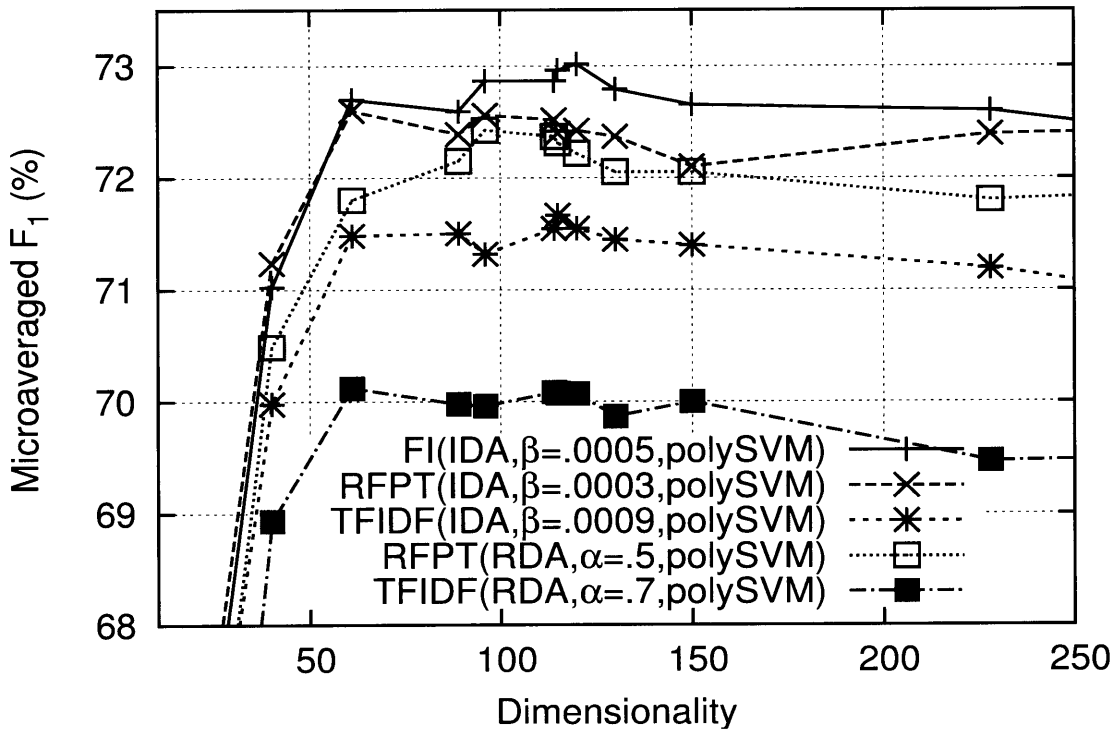


(b) kNN on HD-119 tree of OHSUMED

Figure 5.5: The effect of feature integration (FI) in comparison with other conventional methods when using kNN learning method. Features include relative frequency with power transformation (RFPT), term frequency weighted by inverse document frequency (TFIDF). Feature integration of RFPT and TFIDF is followed by integrated discriminant analysis (IDA).

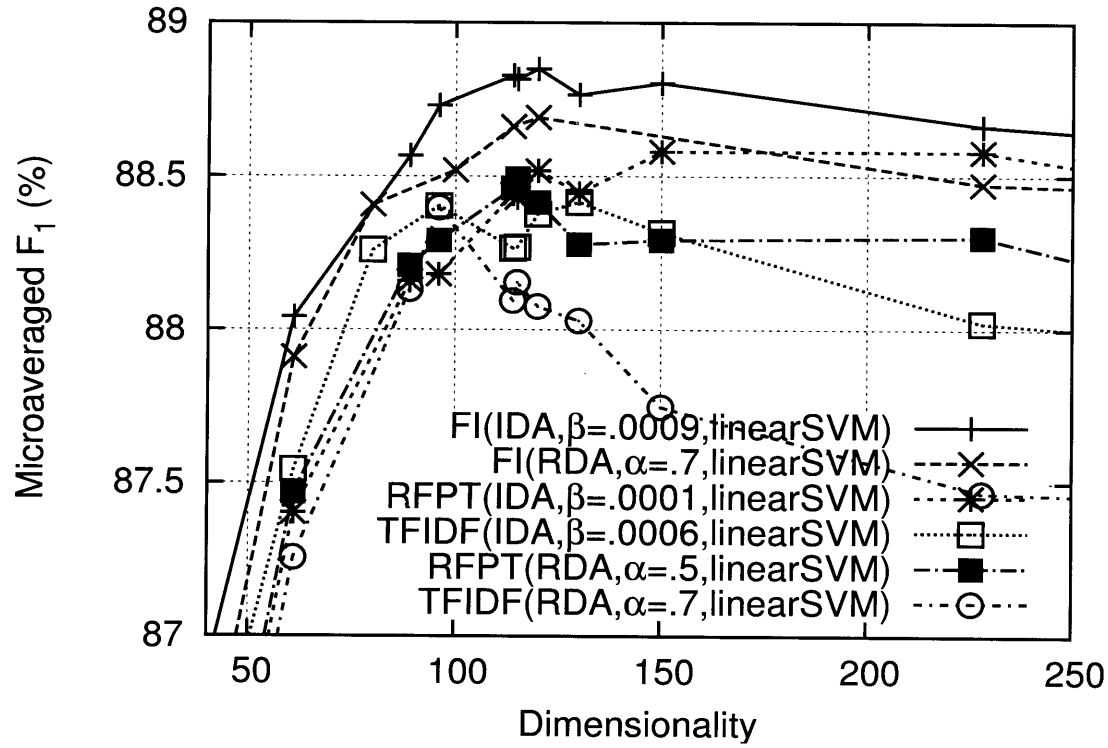


(a) Polynomial SVM on ModApte split of Reuters

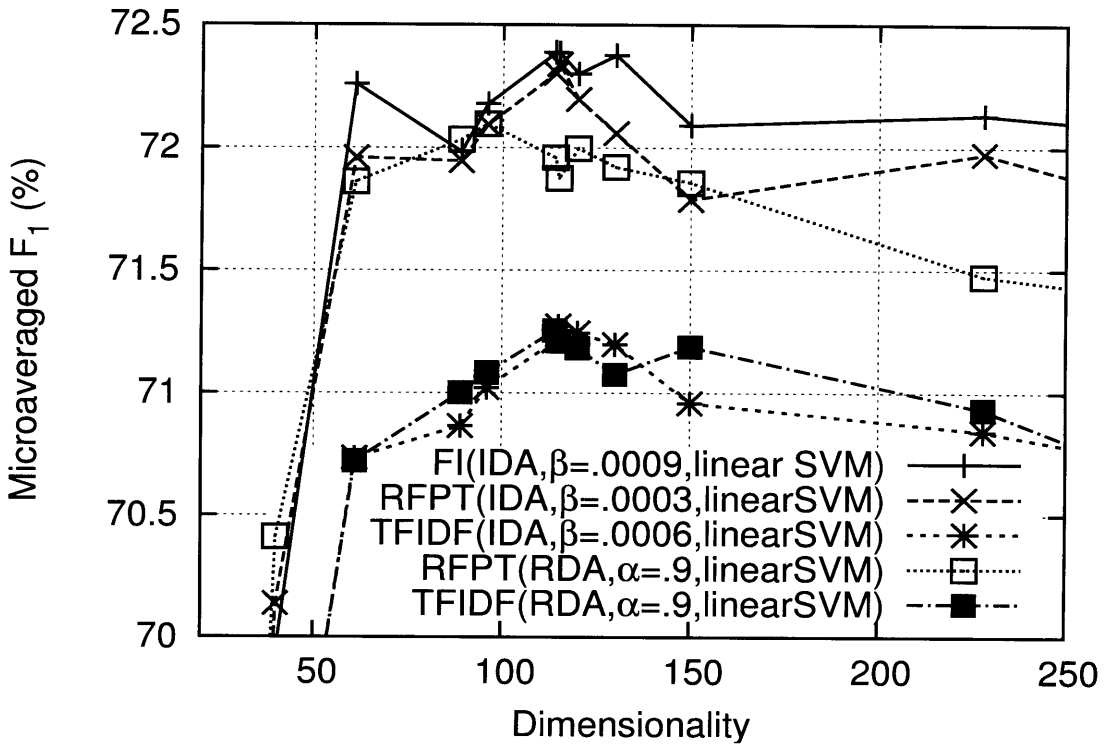


(b) Polynomial SVM on HD-119 tree of OHSUMED

Figure 5.6: The effect of feature integration (FI) in comparison with other conventional methods when using polynomial SVM. Features include relative frequency with power transformation (RFPT), term frequency weighted by inverse document frequency (TFIDF). Feature integration of RFPT and TFIDF is followed by integrated discriminant analysis (IDA).

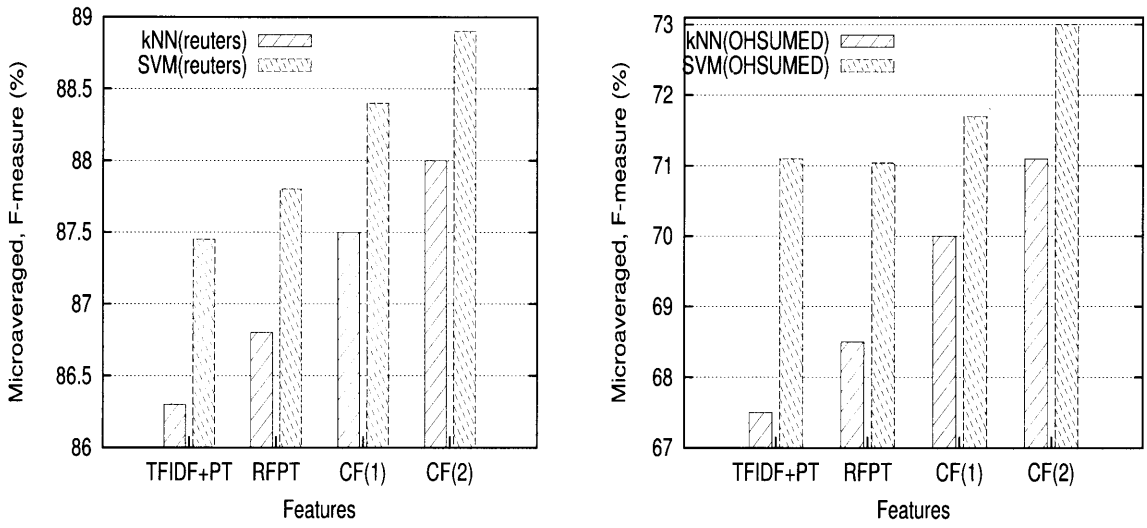


(a) Linear SVM on ModApte Split of Reuters-21578



(b) Linear SVM on HD-119 Tree of OHSUMED

Figure 5.7: The effect of feature integration (FI) in comparison with other conventional methods when using linear SVM. Features include relative frequency with power transformation (RFPT), term frequency weighted by inverse document frequency (TFIDF). Feature integration of RFPT and TFIDF is followed by integrated discriminant analysis (IDA).



(a) Reuters-21578 (b) OHSUMED

Figure 5.8: The effect of feature integration. CF(1) = composite feature by concatenation of CDs of TFIDF and RFPT (classification at $114 * 2$ dimensionality). Although CF(1) is not advocated, it is given here for comparison reasons. CF(2) = composite feature by the concatenation of principal components of TFIDF and RFPT followed by integrated discriminant analysis (IDA).

5.4.2 Statistical Analysis of Improvements

In this section we are interested in statistically analyzing the significance of improvements by feature integration. The statistical test is between RFPT and the integration of it with TFIDF. The null hypothesis is that RFPT and feature integration would achieve the same performance on test data.

Table 5.1 summarizes the statistical analysis results. When using kNN we found that feature integration achieved statistically better results ($p < 0.01$) than RFPT. Similarly, when using SVM, feature integration performed statistically better ($p < 0.05$). Since we found significant improvement in comparison with RFPT, we argue that feature integration was even far better than TFIDF.

Table 5.1: Results of statistical analysis: RFPT versus feature integration (CF(2)). p-values are indicated as p

	Reuters		OHSUMED	
	kNN	SVM	kNN	SVM
RFPT (F-measure)	86.8%	87.8%	68.5%	71%
CF(2) (F-measure)	88%	88.9%	71.1%	73%
McNemar's test	p = 3.177e-05	p =8.487e-08	$p < 2.2$e-16	p = 4.028e-05
χ^2 Test	p = 3.145e-07	p = 0.0273	p = 3.145e-07	p =0.04682
Z-test	p = 3.177e-05	p = 8.487e-08	p = 1.482e-27	p = 4.028e-05

More importantly, the composite feature, CF(2) in Fig. 5.8(a) and 5.8(b) achieved

the highest performances. That is to say, kNN on 'ModApte' split, CF(2) gave a micro-averaged $F_1 = 88\%$ and on OHSUMED it gave a micro-averaged $F_1 = 71.1\%$. Similarly linear SVM on 'ModApte' split, CF(2) gave a micro-averaged $F_1 = 88.9\%$ and on OHSUMED(HD-119) it gave a micro-averaged $F_1 = 73\%$. Considering these data sets and the splits in question, these could be the highest performance scores ever reported in the ATC literature.

5.4.3 The Effect of Multiple Feature-Classifier Combination

As it is described in 5.2.2, we also carried out experiments using multiple feature-classifier combination (MFC). The performance of MFC is higher than the best individual performer. The highest performance by MFC are given by MFC5 of Section 5.2.2. This method performed competitively with FI.

Table 5.2: Summary of the micro-averaged F_1 scores (%) obtained from various methods in comparison with multiple feature-classifier combination. Features include relative frequency with power transformation (RFPT) and term frequency weighted by inverse document frequency (TFIDF). MFC3 and MFC5 refers to multiple feature-classifier combination. Features in MFC3 include RFPT and TFIDF. Classifiers include linear SVM and polynomial SVM. MFC5 is similar to MFC3 except that decision from kNN using RFPT and TFIDF are included.

Data Set	RFPT	CF(2)	MFC3(PCA+CDA)	MFC5(PCA+CDA)	MFC3(IDA)	MFC5 (IDA)
Reuters	87.8	88.9	88.09	88.48	88.9	**89.3**
OHSUMED	71	73	70.3	72.3	72.27	**73.7**

Table 5.2 summarizes the performance of MFC. In both data sets, CF(2) performed slightly better than MFC5 when features were extracted using PCA+CDA. The situation is different when features were extracted by the integrated discriminant analysis (IDA) method. In other words, in the case of the Reuters and OHSUMED data sets, MFC5(IDA) outperformed CF(2) by achieving the highest micro-averaged $F_1 = 89.3\%$ and $F_1 = 73.7\%$, respectively. Considering these data sets and the respective splits, these could be the highest performance scores ever reported in the ATC literature.

Table 5.3 summarizes some results found in the literature in comparison with results presented here. Based on the comparison between classifiers used, the highest performances are in boldface. In other words we compare a particular classifier's results versus the same classifier of the other researchers. For example, we compare our kNN results versus the results in [85] on the same data set. The comparisons we give are just indicative figures in strict terms. This is because a lot of experimental design and pre-processing may differ from other groups of researchers. Although they are just indicative figures, our results are higher than the works in the ATC literature. Our results are consistent with other previous researchers' results [41, 85, 96] in the sense that SVM performed slightly higher than kNN.

It is also worth noting that we used different features such as RFPT. Unlike our work, most of the works in the literature and those in Table 5.3 represent textual data using term weighted vectors commonly called TFIDF.

Most importantly, the proposed techniques achieved not only the highest classification performance as compared to the those in the ATC literature but also used the lowest dimensionality ever before.

Table 5.3: Indicative comparison of results (%) from the literature and our results using Reuters-21578's ModApte Split and 119 MeSH categories for Heart Diseases (HD-119) of OHSUMED data sets. Based on the comparison between classifiers, the highest performances are in boldface. For example our kNN results are compared to other researchers' kNN results. Similarly for SVMs. BEP=Break even point, FI = the proposed Feature Integration method. See Table 5.2 for details on MFC5(IDA).

Researcher	Data Set	Classes	Method Summary	Micro-F_1/BEP
This paper	Reuters-21578	115	FI, kNN, k=5, 114 features	**88**
			FI, SVM, 120 features	**88.9**
			MFC5(IDA), 120 features	**89.3**
	OHSUMED	90	FI, kNN, k=11, 130 features	**71.1**
			FI, polySVM 120 features	**73**
			MFC5(IDA), 120 features	**73.7**
Soucy and Mineau [85]	Reuters-21578	90	ConfWt, kNN, thousands features	86.4
			ConfWt, SVM, thousands features	88.2
	OHSUMED	49	ConfWt, kNN, thousands features	68.7
			ConfWt, SVM, thousands features	70.7
Lam and Han [49]	Reuters-21578	90	TFIDF, GIS, ?features	84.5
	OHSUMED	84	TFIDF, GIS, ?features	58.3
Zhang and Oles [100]	Reuters-21578	115	binary vector, ModLeast Square, 10,000 features	87.2
	OHSUMED	-	-	-
Yang, Y. [95]	Reuters-21578	90	TFIDF, kNN, 24,240features	85
	OHSUMED Anonymous split	?	TFIDF, kNN, ?features	≈ 47
Joachims, T. [41]	Reuters-21578	90	TFIDF, kNN, 1000 features	82.6
			TFIDF, SVM; all features	87.5
	OHSUMED (Not closely related MeSH)	23	TFIDF, kNN, 38,679? features	63.4
			TFIDF, SVM, 38,679 features	71.6

5.5 Summary of Feature Integration and Ensembles

In this chapter, we considered a novel technique for feature integration by concatenation (FIC). It basically takes the advantages of discriminative information from the features involved in the integration. FI uses the concept of concatenation which is applied to the feature vectors.

As expected, the composite features generated using this method improved the classification of classifiers. Statistical analysis of improvements show that the improved performance is statistically significant.

The proposed multiple feature-classifier combination also improved the classification performance outperforming the best individual classifier. Not only that but also it achieved the highest classification score.

It is important to note that the proposed techniques achieved not only the highest classification performance as compared to the those in the ATC literature but also used the lowest dimensionality ever before.

Potential future research on feature integration includes use of multi-classifier combination of the integrated features and use of more samples. In addition, it may be of interest if this technique could be experimentally studied further in applications such as spam filtering and automated survey coding.

Chapter 6

Conclusion

6.1 Introductory Remarks

In this work, the author attempts to address various problems posed by natural language in text classification. The problems tackled include variation of text length, asymmetric sample distribution, high-dimensional space and, the under-sampled problem.

While theoretical backgrounds have been discussed, empirical studies are comprehensively carried out. Statistical analysis of the improvements is also performed. Respective chapters demonstrate that the improvements are statistically significant.

In Section 6.2 therefore we summarize the scientific contributions as per proposed techniques. In this chapter we summarize the conclusions in Section 6.3. In Section 6.4 we outline the future research problems.

6.2 Summary of Contributions

In this Section, we outline the scientific contributions as a consequent of this study. This work proposes various techniques for improving automated text classification. The following items summarize the general contributions that lead to improved machine learning. This dissertation:

(i). Proposed Integrated Discriminant Analysis (IDA) in Text Classification.

In Chapter 4, we propose an integrated discriminant analysis (IDA) which outperforms its counterparts. The multi-label setting was tackled in a similar way as in PCA+CDA algorithm. A comparative study was also carried out. Finally, we conclude that IDA increased learning ability of various methods. This conclusion is based on the improved classification effectiveness and the statistical significance of the improvements.

(ii). Demonstrated the use of the PCA+CDA algorithm in Text Classification for the first time.

Chapter 4 proposes various methods for dimensionality reduction. In the first place, we experimentally studied principal component analysis (PCA), which is not common in ATC. We noted setbacks in PCA method and experimentally studied the canonical discriminant analysis.

However we noted that due to the reason described in Section 1.4.3(v), CDA couldn't be a good choice for ATC. Therefore, we studied the PCA+CDA algorithm. The classical CDA couldn't handle multi-label problems. Since ATC can involve multi-label data, we extended CDA and the PCA+CDA algorithms to handle multi-label learning tasks.

(iii). Developed a Feature Integration (FI) technique to generate a more informative set of features.

Chapter 5 proposes a feature integration (FI) technique that generates composite features with higher discriminating power. Experimental results show that FI improves the performance of ATC.

(iv). Developed a Normalized-weighted metric function for k Nearest Neighbor (kNN).

This work proposes a method called normalized-weighted metric (NWM) for the kNN learning method. We described NWM in 2.4.1. Section 2.6.3.2 presents a comparative study with the classical kNN. It is clear that NWM improves the performance of kNN.

(v). Proposed a function for computing *a posteriori* probability (APP) for Distance Based Learning Methods (DBL).

To the best of our knowledge DBL methods studied in this work are not seen in the TC literature. Therefore DBL were experimentally evaluated as a preliminary study in 2.3.1.

Furthermore we propose use of *a posteriori* probability (PPD) based on DBL methods in 2.4.2. Empirical results show that PPD is far better than the use of distance classifiers in text classification.

(vi). Presented a comprehensive study of RFPT for the first time.

The traditional way of representing textual data is by the term frequency weighted by inverse document frequency (TFIDF). In contrast, we propose the relative term frequency with power transformation (RFPT) in Chapter 3. Empirical results and statistical analysis show that RFPT is better than TFIDF.

(vii). Applied power transformation to TFIDF for the first time.

Furthermore we employed power transformation on TFIDF, and we refer to it as TFIDF+PT. Empirical results show that TFIDF+PT outperforms the conventional TFIDF.

(viii). Demonstrated that RFPT is robust even when noisy texts such as OCR-based Texts are used.

Optical character recognition (OCR) is applied in various areas in daily life. OCR-based texts are full of errors such that the conventional methods are not effective enough. In this work the proposed RFPT is applied to these kind of data and empirical results show that RFPT is suitable to use in this context. Section 3.6 is the topic of this contribution. The experimental study with noisy data show that RFPT is robust regardless of the use of error-prone OCR texts.

(ix). Proposed Multiple Feature-Classifier Combination (MFC)

Chapter 5 also introduces a contribution based on multiple features and multi-classifiers combination (MCC). Unlike the conventional methods of MCC, we used various features which are separately fed to various classifiers then combine their decisions by majority vote rule. Experimental results demonstrate that MFC is effective in automated text classification.

6.3 Conclusions

This work proposes various techniques for improving automated text classification. Statistical analysis is provided to give an understanding on whether the proposed techniques contribute to the improvement. Based on the measures of classification effectiveness and the statistical analysis we can draw conclusions as follows.

(i). The relative frequency (RF) solves the problem of variation in text lengths as shown in 3.3. Empirical study shows that normalizing the text length by use of RF is practical and suitable to be applied in text classification.

(ii). The power transformation solves the asymmetry of sample distribution by removing skewness and kurtosis. Relative frequency with power transformation (RFPT) is better than the classical features, namely term frequency weighted by inverse document frequency (TFIDF). Unlike TFIDF, RFPT have Gaussian-like sample distribution properties. It turns out that RFPT improves the learning ability of classification systems. Therefore, RFPT is better than its counterparts.

(iii). RFPT is robust even when noisy samples like OCR-generated texts are used.

(iv). Application of power transformation to TFIDF improves the symmetry of the sample by removing skewness and kurtosis. Consequently, improved classification effectiveness can be realized.

(v). The PCA+CDA algorithm improves text classification. The reason for improved performance of text classification comes from the fact that the PCA+CDA algorithm solves the problem of singularity of the within-class scatter matrix. This problem occurs due to smaller sample size than its dimensionality. Other reasons for improvements by the PCA+CDA algorithm are as follows. Firstly, it extracts informative features for the classification process. Secondly, it reduces the dimensionality, leading to improved learning efficiency of the classification algorithm.

(vi). Integrated discriminant analysis (IDA) improves text classification. The improvements are brought by a number of reasons. Firstly, it directly solves the problem of singularity by effectively reducing the dimensionality. Secondly it maximizes the between-class meanwhile minimizing the within-class documents, resulting in a well optimized classification task. Thirdly, it extracts informative features for the classification systems. Fourthly the reduced dimensionality leads to improved learning efficiency of the classification system. Empirical results demonstrate the effectiveness of this method.

(vii). The extension of discriminant analysis techniques to handle multi-label data works well with improved classification performance.

(viii). The proposed function namely normalized-weighted metric for kNN is superior to the conventional majority vote rule. This is because it gives more weight to close samples than those which are far from the incoming sample.

(ix). The proposed function for computing *a posteriori* probability from distance based learning methods is superior to the conventional way of using distances in classification decisions.

(x). The proposed multiple feature-classifier combination improves automated text classification.

6.4 Future Research

While this work proposes various techniques for improving text classification, further improvement might be desirable. This section outlines the open research problems which spur further investigation for more improvements in automated text classification (ATC).

The following items briefly describe the areas for future research:

(i). Applications of text classification.

This study focused on machine learning in text classification in general. However, there are a number of applications that can pose different challenges if TC techniques are shipped directly into them. It is therefore interesting to apply the proposed techniques to applications, such as spam filtering and automated survey coding to reveal the applicability of these techniques.

(ii). Can class based term selection improve ATC further?

Term selection before applying the proposed techniques was done based on all words in a given data set. It would be desirable to find out the effect of performing term selection on a class-based (local) vocabulary list first before incorporating such a list into a global vocabulary list. This remains to be an open research problem that can be investigated in the near future.

(iii). Adoption of unsupervised learning methods.

With supervised learning, it requires that there must be manually classified examples, consequently, constraining organizations to hire experts to manually classify learning samples. Possibly this area therefore can be a challenging task to be carried out in the near future; and if unsupervised learning will give promising results, it may lead to freeing organizations from making examples manually.

(iv). This work did not attempt to address the problem of imbalanced sample size. It could be desirable to directly address this problem in future.

(v). Oversampling to address the problem of under-sampled data directly.

Oversampling methods have been applied in some fields to address the problem of imbalanced sample size. Similar techniques can be applied with the objective of mitigating the under-sampled problems before applications of the proposed techniques.

(vi). Research on automated multimedia categorization based on textual data contained in the media. In addition, automated content based indexing of multimedia documents can pose challenges that can be addressed in somehow similar to this work.

(vii). Can the proposed techniques be applied to other languages?

The experimental data used in this work are documents written in English. It would be of interest to use the techniques with other languages such as Swahili and Japanese.

The list of open research problems may not be exhaustive. Therefore other questions may be included for further research.

Bibliography

[1] J. A. Anderson. Separate sample logistic discrimination. *Biometrika*, 59(1):19 – 35, 1972.

[2] C. Apté, F. Damerau, and S. M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[3] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, 2004.

[4] M. W. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval (Software, Environments, Tools), Second Edition.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[5] Y. Bi, D. Bell, H. Wang, G. Guo, and J. Guan. Combining multiple classifiers using dempster's rule for text categorization. *Applied Artificial Intelligence*, 21(3):211–239, 2007.

[6] D. Bicknes. Measuring the accuracy of the OCR in the making of America. Technical report, University of Michigan, 1998.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] H. Borko and M. Bernick. Automatic document classification. *J. ACM*, 10(2):151–162, 1963.

[9] L. S. P. Busagala and T. Elly. Performance of ten internet search engines in retrieving scientific literature in Tanzanian public university libraries. *University of Dar es Salaam Library Journal*, 6(1):1–17, 2004.

[10] L. S. P. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura. Machine learning with transformed features in automatic text classification. In *Proceedings of ECML/PKDD-05 workshop on Sub-symbolic Paradigms for Learning in Structured Domains (Relational Machine Learning)*, pages 11– 20, 2005.

[11] L. S. P. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura. Integrated feature analysis for automatic text classification. In P. Perner, editor, *Machine Learning and Data Minining in Pattern Recognition*, pages 16 – 26. IBaI Publishing, 2007.

[12] L. S. P. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura. Improving automatic text classification by integrated feature analysis. *IEICE Transactions on Information and Systems*, E91-D(4):1101 – 1109, 2008.

[13] L. S. P. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura. Improving text classification by using transformed and integrated features. *Submitted to Pattern Recognition Journal*, 2008.

[14] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318, New York, NY, USA, 1998. ACM.

[15] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[16] H. Chen and T. K. Ho. Evaluation of decision forests on text categorization. In D. P. Lopresti and J. Zhou, editors, *Document Recognition and Retrieval VII*, pages 19–199, 1999.

[17] S. Chhabra, W. S. Yerazunis, and C. Siefkes. Spam filtering using a markov random field model with variable weighting schemas. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 347–350, Washington, DC, USA, 2004. IEEE Computer Society.

[18] G. Chowdhury. *Introduction to modern information retrieval.* Facet publishing, 2004.

[19] W. W. Cohen and H. Hirsh. Joins that generalize: text classification using WHIRL. In R. Agrawal, P. E. Stolorz, and G. Piatetsky-Shapiro, editors, *Proceedings of KDD-98, 4th International Conference on Knowledge Discovery and Data Mining*, pages 169–173, New York, US, 1998. AAAI Press, Menlo Park, US.

[20] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[21] D. R. Cox. Some procedures associated with logistic qualitative response curve. In J. Neyman and F. David, editors, *Research Papers in Statistics.*

[22] N. E. Day and D. F. Kerridge. A general aaximum likelihood discriminant. *Biometrics*, 23:313 – 23, 1967.

[23] T. G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[24] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley & Sons, Inc., second edition, 2001.

[25] S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263, New York, NY, USA, 2000. ACM.

[26] S. T. Dumais, J. C. Platt, D. Hecherman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, and L. Bouganim, editors, *Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management, Bethesda, Maryland, USA, November 3-7, 1998*, pages 148–155. ACM, 1998.

[27] G. Escudero, L. Màrquez, and G. Rigau. Boosting applied toe word sense disambiguation. In *ECML '00: Proceedings of the 11th European Conference on Machine Learning*, pages 129–141, London, UK, 2000. Springer-Verlag.

[28] P. Frasconi, G. Soda, and A. Vullo. Text categorization for multi-page documents: a hybrid naive bayes hmm approach. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 11–20, New York, NY, USA, 2001. ACM.

[29] P. Frasconi, G. Soda, and A. Vullo. Hidden markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems*, 18(2-3):195–217, 2002.

[30] J. H. Friedman. Regularized discriminant analysis. *Journal of American Statistical Association*, 84(405):165–175, 1989.

[31] T. Fukumoto, T. Wakabayashi, F. Kimura, and Y. Miyake. Accuracy improvement of handwritten character recognition by glvq. In *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition Proceedings(IWFHR VII)*, pages 271 – 280, 2000.

[32] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Inc, second edition, 1990.

[33] G. P. C. Fung, J. X. Yu, H. Wang, D. W. Cheung, and H. Liu. A balanced ensemble approach to weighting classifiers for text classification. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 869–873, Washington, DC, USA, 2006. IEEE Computer Society.

[34] L. Gillick and S. Cox. Some statistical issues in the comparison of speech recognitionalgorithms. In *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 532– 535, 1989. Glasgow.

[35] D. Giorgetti and F. Sebastiani. Automating survey coding by multiclass text categorization techniques. *J. Am. Soc. Inf. Sci. Technol.*, 54(14):1269–1277, 2003.

[36] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning.* Springer, August 2001.

[37] P. Herron. Automatic text classification of consumer health web sites using wordnet. Technical report, North Carolina Health Info (NCHI) project, National Library of Medicine, December 2005.

[38] W. Hersh. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 192–201, Dublin, Ireland, 1994.

[39] J. M. G. Hidalgo, M. de Buenaga Rodríguez, and J. C. Cortizo. The role of word sense disambiguation in automated text categorization. In *NLDB*, pages 298–309, 2005.

[40] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, 1998.

[41] T. Joachims. *Learning to classify text using support vector machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers Boston Dordrecht London, 2001.

[42] J. D. Jobson. *Applied Multivariate Data Analysis, Vol. I: Regression and Experimental Design.* Springer-Verlag, 1991.

[43] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–12, 1972.

[44] M. Junker and R. Hoch. An experimental evaluation of ocr text representations for learning document classifiers. *IJDAR*, 1(2):116–122, 1998.

[45] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. Multinomial naive bayes for text categorization revisited. In *Proceedings of AI-04, 17th Australian Joint Conference on Artificial Intelligence*, volume 3339 of *Lecture Notes in Artificial Intelligence*, pages 488–499, Cairns, Australia, 2004. Springer-Verlag.

[46] H. Kim, P. Howland, and H.Park. Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research*, 6:37–53, 2005.

[47] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

[48] S. Lam and L. Lee. Feature reduction for neural network based text categorization. In *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Advanced systems for advanced applications*, pages 195–202, 1999. Hsinchu, TW.

[49] W. Lam and Y. Han. Automatic textual document categorization based on generalized instance sets and a metamodel. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):628–633, 2003.

[50] W. Lam, M. E. Ruiz, and P. Srinivasan. Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):865–879, 1999.

[51] L. S. Larkey. A patent search and classification system. In *DL '99: Proceedings of the fourth ACM conference on Digital libraries*, pages 179–187, New York, NY, USA, 1999. ACM.

[52] L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 289–297, Zürich, CH, 1996. ACM Press, New York, US.

[53] J. Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *AIME '01: Proceedings of the 8th Conference on AI in Medicine in Europe*, pages 63–66, London, UK, 2001. Springer-Verlag.

[54] J. Laurikkala. Instance-based data reduction for improved identification of difficult small classes. *Intelligent Data Analysis*, 6(4):311–322, 2002.

[55] S. L. Lawrence. Online or invisible? *Nature*, 411(6837):521, 2001.

[56] D. Lewis, R. E. Schapire, J. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of* 19^{th} *Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[57] D. D. Lewis. Evaluating Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Morgan Kaufmann, 1991.

[58] D. D. Lewis. Evaluating and Optimizing Autonomous Text Classification Systems. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, Washington, 1995. ACM Press.

[59] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, London, UK, 1998. Springer-Verlag.

[60] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In W. W. Cohen and H. Hirsh, editors, *Proceedings of ICML-94, 11th International Conference on Machine Learning*, pages 148–156, New Brunswick, US, 1994. Morgan Kaufmann Publishers, San Francisco, US.

[61] H. Li and K. Yamanishi. Text classification using ESC-based stochastic decision lists. In *Proceedings of CIKM-99, 8th ACM International Conference on Information and Knowledge Management*, pages 122–130, Kansas City, US, 1999. ACM Press, New York, US.

[62] Y. Li and A. K. Jain. Classification of text documents. *The Computer Journal*, 41(8):537–546, 1998.

[63] Library Digital Initiatvive Project. Measuring search retrieval accuracy of uncorrected OCR: Findings from the harvad-radcliffe online historical reference shelf digitization project. Technical report, Harvard University Library, 8 2001.

[64] H.-S. Lim. Improving kNN based text classification with well estimated parameters. In *International Conference on Neural Information Processing*, pages 516 – 523, 2004.

[65] B. Mandelbrot. A note on a class of skew distribution functions: Analysis and critique of a paper by h. a. simon. *Information and Control*, 2(1):90–99, 1959.

[66] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[67] A. Mccallum and K. Nigam. A comparison of event models for naive bayes text classification. Technical report, American Association for Artificial Intelligence Workshop on Learning for Text Categorization, 1998.

[68] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[69] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 1947.

[70] M. Murata, L. S. P. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura. The impact of OCR accuracy and feature transformation on automatic text classification. In H. Bunke and A. spdz, editors, *Document Analysis Systems VII*, pages 506 – 517. Springer, 2006.

[71] A. Myka and U. Güntzer. Measuring the effects of ocr errors on similarity linking. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 968–973, Washington, DC, USA, 1997. IEEE Computer Society.

[72] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2-3):103–134, 2000.

[73] Z.-Y. Niu, D.-H. Ji, and C. L. Tan. A semi-supervised feature clustering algorithm with application to word sense disambiguation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 907–914, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[74] M. Ohta, A. Takasu, and J. Adachi. Retrieval methods for English-text with missrecognized ocr characters. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 950–956, Washington, DC, USA, 1997. IEEE Computer Society.

[75] R. J. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.

[76] J. Rennie, L. Shih, J. Teevan, and D. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 616–623, Washington DC, 2003).

[77] S. Robertson. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 2004.

[78] D. Roth. Learning to resolve natural language ambiguities: a unified approach. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 806–813, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[79] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[80] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[81] K.-M. Schneider. Weighted average pointwise mutual information for feature selection in text categorization. In *Principles and Knowledge Discovery in Databases (PKDD)*, pages 252–263, 2005.

[82] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[83] R. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to text categorization. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM)*, pages 78 – 85, 2000.

[84] C. Siefkes, F. Assis, S. Chhabra, and W. S. Yerazunis. Combining winnow and orthogonal sparse bigrams for incremental spam filtering. In *PKDD '04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 410–421, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[85] P. Soucy and G. Mineau. Beyond TFIDF weighting for text categorization in the vector space model. In *Proceeding of the international Joint Conference on Artificial Intelligence (IJCAI)*, pages 1130–1135, 2005.

[86] K. Taghva, T. Nartker, J. Borsack, S. Lumos, A. Condit, and R. Young. Evaluating text categorization in the presence of OCR errors. In *Proc. IS&T/SPIE 2001 Intl. Symp. on Electronic Imaging Science and Technology*, pages 68–74, San Jose, CA, January 2001.

[87] C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management: an International Journal*, 38(4):529–546, 2002.

[88] C. J. van Rijsbergen. *Information Retrieval.* Butterworth, London, 1979. Chapter 7.

[89] P. Verboon and I. A. van der Lans. Robust canonical discriminant analysis. *Psychometrika Journal*, 59(4):48–507, 1994.

[90] M. Wang and J. Nie. A latent semantic structure model for text classification. In *Proceedings of ACM SIGIR workshop on Mathematical/formal methods in information retrieval*, Toronto, Canada, 2003.

[91] T.-Y. Wang and H.-M. Chiang. Fuzzy support vector machine for multi-class text categorization. *Inf. Process. Manage.*, 43(4):914–929, 2007.

[92] A. R. Webb. *Statistical Pattern Recognition, 2nd Edition.* John Wiley & Sons, October 2002.

[93] Y. Xu, G. Jones, J. Li, B. Wang, and C. Sun. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*, 1(2):203–213, 2005.

[94] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.

[95] Y. Yang. A study on thresholding strategies for text categorization. In *Proceedings of the 24th ACM/SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 137–145, 2001.

[96] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the Twenty-First International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

[97] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[98] F. Yates. Contigency tables involving small numbers and the χ^2 test. *Journal of Royal Statistical Society*, Supplement(1):217 – 235, 1934.

[99] J. H. Zar. *Biostatistical Analysis.* Prentice Hall, fourth edition, 9999.

[100] T. Zhang and F. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval Journal*, 4:5–31, 2001.

[101] G. K. Zipf. *Human Behavior and the Principle of Least-Effort.* Addison-Wesley (Reading MA), Cambridge, MA, 1949.

[102] G. Zu, M. Murata, W. Ohyama, T. Wakabayashi, and F. Kimura. The impact of ocr accuracy on automatic text classification. In C.-H. Chi and K.-Y. Lam, editors, *Proceedings of Advanced Workshop Content Computing(AWCC)*, volume 3309 of *Lecture Notes in Computer Science*, pages 403–409. Springer, 2004.