A Foundation of Class-Indexing and Class-Semantic-Indexing based Term Weighting Approaches for Automatic Text Classification

Mohammad Golam Sohrab

A dissertation submitted to the University of Tokushima
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

September, 2013

Department of Information Science and Intelligent Systems
Graduate School of Advanced Technology and Science
The University of Tokushima, Japan

“Read. Read in the name of thy Lord who created; [He] created the human being from blood clot. Read in the name of thy Lord who taught by pen: [He] taught the human being what he did not know.”

-Al-Qur’an (96:1-5)

Acknowledgement

First, I am very much grateful to my Creator Almighty **ALLAH**, the most merciful and the most beneficent, for giving me the strength, patience and opportunity to work on a much needed research field like this during my study at the University of Tokushima and all my life.

I would like to thank my supervisor Professor **Fuji Ren**, for his constant guidance, kind support, help, continuous encouragement, and for his patience hearing to my problems throughout the accomplishment of these studies. I appreciate his in-exhaustive efforts, unending cooperation and advices, his deep insights that always found solutions when problems supervened and very creative criticism. I also thank him for introducing me to the Natural Language Processing fields of Information Science and Intelligent systems.

I would like to thank, Professor **Kenji Terada** and Professor **Masami Shishibori** for accepting to be staff members of the dissertation committee of this thesis, and their insightful comments and genuine encouragement during the dissertation review.

Special thanks are given to Associate Professor **Motoyuki Suzuki** and Associate Professor **Mohamed Abdel Fattah** for their useful advices and enthusiastic help during my research period. I also want to thank all of my Doctoral course teachers, who provided me valuable English documents for their certain course.

I would like to acknowledge the University of Tokushima, Graduate School of Advanced Technology and Science, Department of Information Science and System Engineering and Ren Lab (A1 Laboratory), for providing a good environment to complete my doctoral research.

I am so grateful to my wife, **Syeda Tasnuva akhter,** I could not complete this work without her unwavering love, support, patience, prayer, supplication (Doa'a), and understanding during many hours of working and preparing my studies for several years living in abroad.

Finally, I would like to deeply thank my parents (my beloved Father **Mohammad Abu Baker** who has passed away during my entire period of Ph.D. and ask to Allah (SWT) to except all his good deeds and provide him Al-Jannah; my beloved mother for supporting me by her Doa'a), and my father-in-law and mother-in-law for their unwavering love, patience and prayers. I am so grateful to my elder brothers and sisters for their unwavering love and supports.

The journey of this period will remain in my life forever.

Contents

List of Tables

List of Figures

Abstract

With the large amount of textual information available digitally, indexing plays a significant role to design incorporates adjustment of the weights of natural language documents and interdisciplinary concepts from linguistic data. Indexing is the process to optimize search speed and performance in finding relevant documents to enhance information retrieval task.

This thesis presents a foundation of class-indexing, combined term weighting schemes (CTWS), and class-semantic-indexing based weighting approaches to enhance classification task. Most of the previous studies related on different term weighting emphasize on the document-indexing-based and four fundamental information elements-based approaches to address automatic text classification (ATC).

In the section on class-indexing, we introduce class-indexing-based term-weighting approaches and judge their effects in high-dimensional and comparatively low-dimensional vector space over the TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, and TF.IDF term weighting approaches that are considered as the baseline approaches. First, we implement class-indexing-based TF.ICF, TF.IDF.ICF observational term weighting approaches in which the inverse class frequency (ICF) is incorporated. In the experiment, we investigate the effects of TF.IDF.ICF over the Reuters-21578, 20Newsgroups, and RCV1-v2 datasets as benchmark collections, which provide positive discrimination on rare terms in the vector space and biased against frequent terms in the text classification (TC) task. Therefore, we revised the ICF function and implemented a new inverse class space density frequency ($ICS_\delta F$), and generated the, TF.IDF.$ICS_\delta F$ method that provides a positive discrimination on infrequent and frequent terms. We present detailed evaluation of each category for the three datasets with different term weighting approaches. The experimental results show that the proposed class-indexing-based TF.IDF.$ICS_\delta F$ term weighting approach is promising over the compared well-known baseline term weighting approaches.

In the section on combined term weighting schemes (CTWS), where CTWS is incorporated with ten different weighting approaches, including TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.$ICS_\delta F$, TF.IDF.ICF , and TF.IDF.$ICS_\delta F$. To calculate the global weights (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}) using ten different weighting approaches, we therefore introduce five different models, including CTWS with Summation (CTWS-Sum), CTWS with average (CTWS-Avg.), CTWS with

Mathematical Regression Model (CTWS-MR), CTWS with Genetic Algorithm Model (CTWS-GA), and CTWS with Feed Forward Neural Network (CTWS-FFNN). In the experiment, we investigate the effects of CTWS-Sum, CTWS-Avg., CTWS-MR, CTWS-GA, and CTWS-FFNN over the Reuters-21578 and 20Newsgroups datasets. The experimental results show that the CTWS is promising in different classifiers to enhance automatic text classification.

Finally in the section on class-semantic-indexing, we present a prototype class-semantic-indexing which is incorporated with term, document, and class index; to compute a weight of a certain term and its semantic weight from a corpus-based and WordNet-dictionary-based combinational approach.

Chapter 1

Introduction

This thesis presents research on building a set of different class-indexing based term weighting approaches for automatic text classification; a new approach of indexing to the field of classification or information retrieval in order to enhance automatic text classification (ATC). Most automatic classification systems analyze a text statistically and linguistically, determine important terms from document and generate a text to vector representation from these important terms. A set of algorithms, are created to perform the automatic text classification systems.

1.1 Motivation

With large amount of textual information available digitally, effective retrieval is difficult to accomplish without good text to vector representation in order to enhance automatic text classification (ATC) [3, 8, 23, 33]. In the vector space model (VSM) [32], the content of a text is represented as a vector in the term space. The term weight is the degree of importance of term t_i in document d_j. The term weighting approach plays a very significant role to enhance automatic text classification (TC). Therefore, an effective indexing-based term weighting approach can generate more information-rich terms and assign appropriate weighting values to the terms.

In general, text to vector representation can be classified into two tasks: indexing and term weighting [14, 19]. Indexing based on the documents provides a meta-language for describing the document, where the information about a set of documents in a certain class c_k is missing. In the TC task, the analysis of document contents by traditional term indexing that is based on documents is not enough to enhance the performance of classification task. Accurate retrieval depends on the exactness of the document retrieval process and class description for a certain term $t_i = \{t_1, t_2, \ldots\ t_n\}$ in the TC task. Most research does not show the diversity of category information for a certain class in the classification task.

The primary motivation for exploiting the class-indexing-based term weighting method for TC can be attributed to two main properties. First, a more information-rich term weighting method with effective indexing can generate a more effective classifier. Second, there is a demand for dimensionality reduction through inverse class space density frequency (ICS$_\delta$F).

1.2 Problem Description

Recently, many experiments have been conducted using a document-indexing-based term weighting approach to address the classification task as a statistical method [6, 17, 24, 29, 30, 31, 37, 38]. TF.IDF is considered to be the most popular term weighting method in successfully performing the ATC task and document-indexing [30]. Salton & Buckley [30] discussed many term weighting approaches in the information retrieval (IR) field, and found that normalized TF.IDF is the best document weighting function. Therefore, beside other weighting methods, this term weighting scheme is used as a standard in this study and as a default term weighting function for TC. However, there are some drawbacks of the classic TF.IDF term weighting scheme for the TC task. In the training vector space, to compute the weight of a certain term, the category information is constantly omitted by the document-indexing-based TF.IDF term weighting method. In contrast, the inverse document frequency (IDF) function provides the lowest score of those terms that appear in multiple documents; because of this, the TF.IDF score gives positive discrimination to rare terms and is biased against frequent terms. At present, because TF.IDF uses a default term weighting parameter in the classification task, a variety of feature selection techniques, such as information gain [18], chi-square test, and document frequency [49], have been used to reduce the dimension of the vectors.

A major characteristic or difficulty of TC is the high dimensionality of the feature space [10]. Since the conventional TF.IDF term weighting scheme favors rare terms and the term space consists of hundreds of thousands of rare terms, sparseness frequently occurs in document vectors. Therefore, a novel term weighting method is needed to overcome the problem of high dimensionality and for the effective indexing based on class to enhance the classification task.

1.3 Contribution

In the last few years, researchers have attempted to improve the performance of TC by exploiting statistical classification approaches [46, 47] and machine learning techniques, including probabilistic Bayesian models [20, 28, 43], support vector machines (SVMs) [7, 11, 26, 44], decision trees [20], Rocchio classifiers [21], and multivariate regression models [35, 48]. However, a novel term weighting scheme with a good indexing technique is needed in addition to these statistical methods to truly enhance the classification task. Therefore, in this study, we have developed a novel class-indexing-based term weighting scheme that enhances the indexing process to generate more informative terms. We propose an automatic indexing method using the combination of document-based and class (category)-based approaches. This study makes the following primary contributions with introducing proposed class-indexing-based term weighting approaches, combined term weighting scheme and class-semantic-indexing to enhance automatic text classification.

- The TF.IDF.ICF term weighting approach that can compute weight of a certain term t_i from a certain document d_j with respect to a certain category c_k. This weighting approach favors the rare terms and is biased against frequent terms. It is capable to enhance the classification task than conventional approaches.
- The TF.IDF.ICS$_\delta$F term weighting approach that can compute the density weight of a certain term t_i from a certain document d_j with respect to a certain category c_k. This term weighting that gives a positive discrimination both to rare and frequent terms. The TF.IDF.ICS$_\delta$F approach is very prominent and considering a novel weighting approach.
- The proposed class-indexing-based TF.IDF.ICS$_\delta$F term weighting approach is outperformed with existing term weighting approaches.
- The proposed TF.IDF.ICS$_\delta$F approach is very effective in high-dimensional and comparatively low-dimensional vector spaces. Therefore, this approach is capable to overcome the problem of high dimensionality.
- The proposed class-indexing method expands the existing document-indexing method in term indexing and generates more informative terms based on a certain category through use of inverse class frequency (ICF) and ICS$_\delta$F functions.

Alongside of class-indexing, we therefore, introduce Combined Term weighting Scheme (CTWS), which is integrated with ten different weighting approaches. These ten different weighting approaches are associated with ten global weights. In this work, we introduce five different models to compute the global weight from a certain weighting scheme, including TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.ICS$_\delta$F, TF.IDF.ICF, and TF.IDF.ICS$_\delta$F from a certain dataset. Finally, we introduce class-semantic-indexing which is incorporated with class-indexing and semantic-indexing. In this method, we introduce how to determine the sense density of a certain term which is appeared in the corpus through corpus-based and Wordnet-dictionary-based approach.

1.4 Outline

This chapter gives the motivation, problem description, contribution and outline of the thesis. Chapter 2 goes over prerequisite concepts of classification system. It explains the basic concepts of information retrieval, machine learning, vector space model, and text preprocessing and performance evaluation. Chapter 3 gives an overview of proposed class-indexing, is considered the core indexing method of this work. Chapter 4 presents another term weighting approaches, combining with all term weighting approaches which are used in section 3. Chapter 5 gives an overview of class-semantic-indexing, another way of combining corpus and WordNet based weighting approach to enhance automatic text classification. Finally, in chapter 6 conclusion and future work are discussed.

Chapter 2

Prerequisite Concepts of Classification

This chapter presents a brief introduction of prerequisite essential knowledge on text classification and state-of-the-art of this research. In section 2.1 gives an overview of information retrieval with their available models that are widely used. In section 2.2 presents introduce with and machine learning technique which is widely used in to extract information from data automatically. In section 2.3 gives an overview and the properties of vector space model. In section 2.4 gives an overview text preprocessing. Finally section 2.5 shows widely used evaluation methods in the field of text classification.

2.1 Information Retrieval

Information Retrieval (IR) is the act of storing, searching, and retrieving information that match a user's request [51]. Automatic is opposed to manual and information is opposed to data or fact. First implementation of information systems were introduced in the 1950s and 1960s. By 1990 several different techniques had been shown to perform well on small text corpora [52]. Retrieved documents are then either directly sent to the user or ranked by relevance and then sent to the user. With the recent few years there have been number of proposed and used in different types of model like self-theoretic (standard Boolean, extended Boolean and Fuzzy retrieval), algebraic (vector space model, generalized vector space model, Topic-based vector space model, enhanced topic-based vector space model, latent semantic indexing aka), and probabilistic model (Binary independence retrieval, probabilistic relevance model, uncertain inference, language model, divergence-from-randomness model, latent dirichlet allocation). The most popular and widely used models are: the Boolean model and the vector space model.

2.2 Machine Learning Techniques

Machine learning (ML) is a broad subfield of artificial intelligence, which is concerned with the design and development of algorithms and techniques that allows computer to learn. The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. The ML system focuses on prediction, based on known properties learned from the training dataset. In the ML-based classification task, documents are assigned into its predefined categories in the training phage. In the machine-learning workbench, several learning algorithms, including Naïve Bayes, K-nearest neighbor, centroid classifier, SVM, etc. are using to train the model. In the testing phage, a set of independent data that follows the same probability distribution as the training data set is used to obtain the classification model performance such as accuracy, F-measure and so on. Several steps are considered in the machine learning techniques such as text pre-processing, feature selection, text to vector generation using different term weighting approaches and classifier construction.

2.3 Vector Space Model

The vector space model, introduced by Salton [29, 30] in the 70s, describes documents and queries in terms of term, or keyword, vectors. Conversion into vectors allows for algebraic manipulation and comparison. The vector space model allows for partial matches, ranking of results and does not treat all terms equally. Construction of a vector space model can be done in two steps. The first step is to build an index of the document collection. The second step is to weight the individual terms.

2.3.1 Indexing

In information retrieval and automatic text processing, the construction of effective indexing vocabularies has always been considered the single most important step [34]. Therefore, an information-rich term weighting method with effective indexing can generate a more effective classifier.

2.3.2 Term Weighting

The most widely used traditional document-indexing-based TF.IDF term weighting approach for ATC has two subtasks. First is the term frequency based analysis of the text contents, and second is the determination of terms that are in the document space d_j. In addition, measuring term specificity was first proposed by Sparck Jones [37] and it later became known as the IDF. The reasoning is that a term with a high document frequency (DF) count may not be a good discriminator, and it should be given less weight. Therefore, IDF means that a term that occurs in many documents is not a good document discriminator. The IDF function assigns the lowest score to those terms that appear in multiple documents in a document space $D = d_1, d_2... d_n$. To quantitatively judge the similarity between a pair of documents, a weighting method is needed to determine the significance of each term in differentiating one document from another. The common term weighting method is TF.IDF [30], as defined in Eq. 1:

$$W_{TF.IDF}(t_i, d_j) = tf_{(t_i,d_j)} \times \left(1 + log\,\frac{D}{d(t_i)}\right). \tag{1}$$

Its cosine normalization is denoted as

$$W_{TF.IDF}^{norm}(t_i, d_j) = \frac{tf_{(t_i,d_j)} \times (1 + log\frac{D}{d(t_i)})}{\sqrt{\sum_{t_i \in d_j}[tf_{(t_i,d_j)} \times (1 + log\frac{D}{d(t_i)})]^2}}, \tag{2}$$

where D denotes the total number of documents in the collection, $tf_{(ti,dj)}$ is the number of occurrences of term t_i in document d_j, $d(t_i)$ is the number of documents in the collection in which term t_i occurs at least once, $\frac{d(t_i)}{D}$ is referred to as the DF, and $\frac{D}{d(t_i)}$ is the IDF of term $t_{i.}$

Figure 1 shows a simple overview of document-indexing. The classic document-indexing is incorporated with term index and document index. Figure 2 shows an example of creating a document-indexing-based vector space model for a sample of eight documents. Recently, various term weighting methods alongside with document-indexing, including relevance frequency (RF) [16, 24], probability based (PB) approach

[24], mutual information (MI) [24, 33], odds ratio (OR) [24, 33], correlation coefficient (CC) [24] have been reported that these term weighting methods can significantly improve the performance of ATC. Therefore, to compare with proposed weighting approaches, we implement these term weighting methods based on four fundamental information elements. The mathematical expression of TF.CC, TF.MI, TF.OR, TF.PB, and TF.RF weighting schemes are as,

$$TF.CC = tf_{(t_i,d_j)} \times \left(\frac{\sqrt{N}(AD-BC)}{\sqrt{(A+C)(B+D)(A+B)(C+D)}}\right), \tag{3}$$

$$TF.MI = tf_{(t_i,d_j)} \times log\left(\frac{AN}{(A+B)(A+C)}\right), \tag{4}$$

$$TF.OR = tf_{(t_i,d_j)} \times log\left(\frac{AD}{BC}\right), \tag{5}$$

$$TF.PB = tf_{(t_i,d_j)} \times log\left(1 + \frac{A}{B}\frac{A}{C}\right), \tag{6}$$

$$TF.RF = tf_{(t_i,d_j)} \times log\left(2 + \frac{A}{C}\right), \tag{7}$$

A denotes the number of documents belonging to category c_k where the term t_i occurs at least once; *B* denotes the number of documents not belonging to category c_k where the term t_i occurs at least once; *C* denotes the number of documents belonging to category c_k where the term t_i does not occur; *D* denotes the number of documents not belonging to category c_k where the term t_i does not occur.

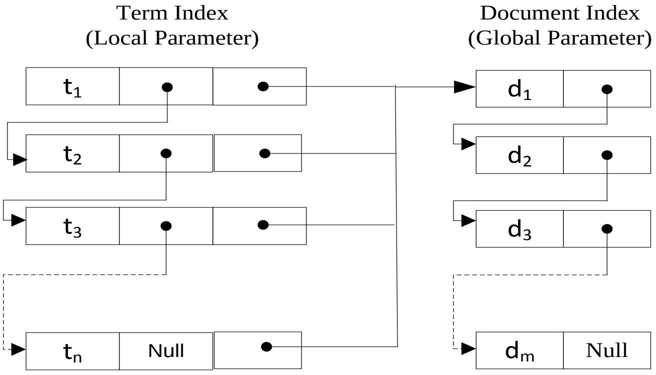
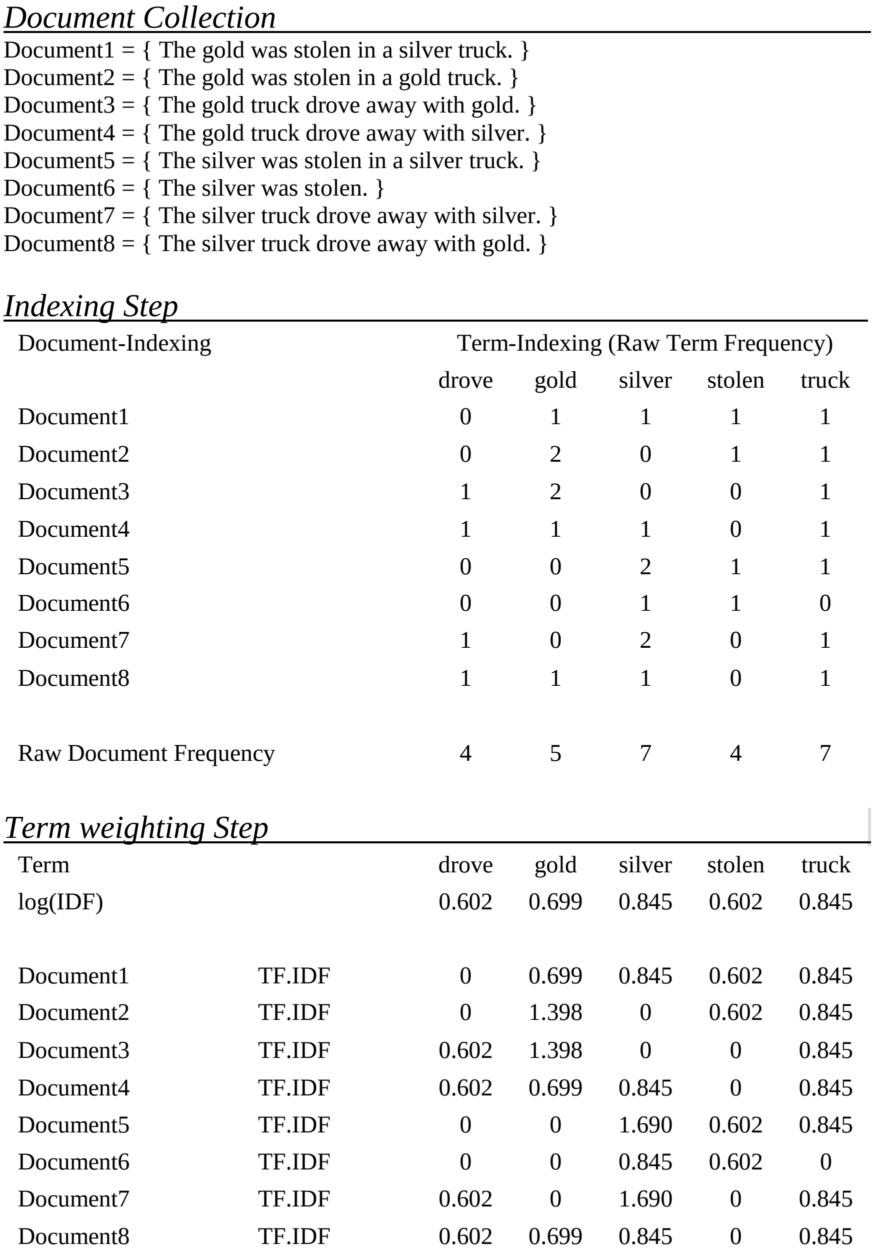


Figure 1. Document-indexing: the formulation of classic TF.IDF



Document Collection

Document1 = { The gold was stolen in a silver truck. }
Document2 = { The gold was stolen in a gold truck. }
Document3 = { The gold truck drove away with gold. }
Document4 = { The gold truck drove away with silver. }
Document5 = { The silver was stolen in a silver truck. }
Document6 = { The silver was stolen. }
Document7 = { The silver truck drove away with silver. }
Document8 = { The silver truck drove away with gold. }

Indexing Step

Document-Indexing	Term-Indexing (Raw Term Frequency)				
	drove	gold	silver	stolen	truck
Document1	0	1	1	1	1
Document2	0	2	0	1	1
Document3	1	2	0	0	1
Document4	1	1	1	0	1
Document5	0	0	2	1	1
Document6	0	0	1	1	0
Document7	1	0	2	0	1
Document8	1	1	1	0	1
Raw Document Frequency	4	5	7	4	7

Term weighting Step

Term		drove	gold	silver	stolen	truck
log(IDF)		0.602	0.699	0.845	0.602	0.845
Document1	TF.IDF	0	0.699	0.845	0.602	0.845
Document2	TF.IDF	0	1.398	0	0.602	0.845
Document3	TF.IDF	0.602	1.398	0	0	0.845
Document4	TF.IDF	0.602	0.699	0.845	0	0.845
Document5	TF.IDF	0	0	1.690	0.602	0.845
Document6	TF.IDF	0	0	0.845	0.602	0
Document7	TF.IDF	0.602	0	1.690	0	0.845
Document8	TF.IDF	0.602	0.699	0.845	0	0.845

Figure 2. Example of Constructing a Document-indexing-based Vector Space Model

2.4 Text Preprocessing

In Natural Language Processing, text preprocessing is often the first step to analyze language. Text preprocessing is the task to normalize the text documents into a common tokenization. In the text processing step, first normalize all the documents for a certain dataset[1] using several text preprocessing steps, including converting uppercase letters

[1] In this experiment: Reuters-21578, 20Newsgroups, and RCV1-v2

into a token to lowercase, removing punctuation, eliminating stop-words, and reducing inflected words to their root form using stemming.

2.4.1 Stemming

Stemming is the process for reducing inflected words of their stem, base or root form, generally a written word form. Stemming is widely using in Search engine technology for query expansion and natural language processing. In this experiment, we used Porter stemming algorithm[2] for text normalization.

2.4.2 Stop Words

Stop words or stop-words, is the name given to words which are filtered out prior to, or after, processing of natural language data (text). As for statistical data corpus, frequently appearing stop list[3] that can decrease the feature parameters score value. Removing stop words is the task to assist reducing vector dimension, decrease the system computational cost and improve the classification performance.

2.5 Evaluation Method

This section will present some the most widely used evaluation metrics for IR. It will present in detail precision, recall and F-measure.

2.5.1 Precision, Recall and F-measure

The standard methods used to judge the performance of a classification task are precision, recall, and the F_1 measure [10, 47]. These measures are defined based on a contingency table of predictions for a target category c_k. The precision $P(C_k)$, recall $R(C_k)$, and the F_1 measure $F_1(C_k)$ are defined as follows:

[2] http://maya.cs.depaul.edu/~classes/ds575/porter.html

[3] Available at: http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

$$P(C_k) = \frac{TP(C_k)}{TP(C_k)+FP(C_k)} \qquad (8)$$

$$R(C_k) = \frac{TP(C_k)}{TP(C_k)+FN(C_k)} \qquad (9)$$

$$F_1(C_k) = \frac{2.P(C_k).R(C_k)}{P(C_k)+R(C_k)} = \frac{2TP(C_k)}{2TP(C_k)+FP(C_k)+FN(C_k)} \qquad (10)$$

TP(*C*$_k$) is the set of test documents correctly classified to the category *C*$_k$, *FP*(*C*$_k$) is the set of test documents incorrectly classified to the category, *FN*(*C*$_k$) is the set of test documents wrongly rejected, and *TN*(*C*$_k$) is the set of test documents correctly rejected. To compute the average performance, we used macro-average, micro-average, and overall accuracy. The macro-average of precision (P^M), recall (R^M), and the F_1 measure (F_1^M) of the class space are computed as

$$P^M = \frac{1}{m}\sum_{k=1}^{m} P(C_k) \qquad (11)$$

$$R^M = \frac{1}{m}\sum_{k=1}^{m} R(C_k) \qquad (12)$$

$$F_1^M = \frac{1}{m}\sum_{k=1}^{m} F_1(C_k) \qquad (13)$$

Therefore, The micro-average of precision (P^μ), recall (R^μ), and the F$_1$ measure (F_1^μ) of the class space are computed as

$$P^\mu = \frac{\sum_{k=1}^{m} TP(C_k)}{\sum_{k=1}^{m}(TP(C_k)+FP(C_k))} \qquad (14)$$

$$R^\mu = \frac{\sum_{k=1}^{m} TP(C_k)}{\sum_{k=1}^{m}(TP(C_k)+FN(C_k))} \qquad (15)$$

$$F_1^\mu = \frac{2.P^\mu.R^\mu}{P^\mu+R^\mu} \qquad (16)$$

2.6 Summary

This chapter presented an overview of the basic concepts of text classification that are used and built upon within this thesis. First, the important concept of information retrieval, machine learning, vector space model and text preprocessing were discussed. Next, the standard evaluation methods for text summarization were introduced.

Chapter 3

Class-Indexing

This chapter presents a brief introduction of building essential knowledge on class-indexing and state-of-the-art of prototype class-indexing systems. In section 3.1 gives an overview of related work of term weighting approaches in information retrieval that are widely used. In section 3.2 presents introduce of proposed class-indexing based term weighting scheme with and machine learning technique which is used in to extract weight from data automatically. In section 3.4 gives an overview of different machine learning based classifiers. In section 3.5 shows experiment design and evaluation of class-indexing based approaches. Finally section 3.6 shows overall performances and discussions in the field of text classification.

3.1 Related Work

In the statistical-based classification method [6, 17, 24, 29, 30, 31, 37, 38], many experiments have been conducted using different term weighting methods to address classification task. Salton & Buckley [30] discussed many term weighting approaches in the information retrieval (IR) field, and reported that normalized TF.IDF is the best document weighting function. Flora & Agus [4] performed some experiments to recommend the best term weighting function for both document and sentence-level novelty mining. The results show that the TFIDF weighting function outperforms in TREC2003 and TREC2004 datasets. Lartnatte and Theeramunkong [42] investigated the various combinations of inter-class standard deviation, class standard deviation and standard deviation with TF.IDF, and the average results showed that the TF.IDF was superior in nine out of ten term weighting methods. Only one method performed better than TF.IDF by 0.38% on average.

In the last few years, researchers have attempted to improve the performance of TC by exploiting statistical classification approaches [46, 47] and machine learning techniques, including probabilistic Bayesian models [20, 28, 43], support vector machines (SVMs) [7, 11, 26, 44], decision trees [20], Rocchio classifiers [21], and multivariate regression models [35, 48]. However, a novel term weighting scheme with a good indexing technique is needed in addition to these statistical methods to truly enhance the classification task.

3.2 Class-indexing-based Term Weighting Scheme

Of greater interest is that term and document frequencies that depend only on the occurrence characteristics of terms in documents are not enough to enhance the classification task. In terms of class-oriented indexing, a subset of documents from document space $D = d_1, d_2 ... d_n$ is allocated to a certain class in order to create a VSM in the training procedure. In the traditional TF.IDF method, a list of identical documents is linked with a single term and the IDF function provides the lowest score for that term. However, it is not obvious that in the classification task, the IDF function provides the lowest score for those terms that are identical. These identical documents linked with a certain term t_i might be a sub-part of a certain category c_k. Therefore, it is important to explore the occurrence characteristics of terms in both the document space $D = \{d_1, d_2 ... d_n\}$ and the class space. In class-oriented indexing, a subset of documents from the global document space is allocated to a certain class c_k ($k = 1, 2, ... m$) according to their topics in order to create a boundary line vector space in the training procedure. Therefore, the class space is defined as $C = \{(d_{11}, d_{12}, ... d_{1n}) \in C_1, (d_{21}. d_{22}, ... d_{2n}) \in C_2, \dots \dots (d_{m1}, d_{m2}, ... d_{mn}) \in C_m\}$ where a set of documents with same topics is assigned to a certain class c_k in the training procedure. In the class space, where the category itself has its own domain and domain size is dissimilar to other categories, which is based on the number of documents that the domain contains. For this reason, the knowledge of a set of relevant documents containing a certain term t_i in a certain class c_k is considered to determine the category mapping in the training procedure.

3.2.1 Prototype Class-indexing System

A simple overview of the proposed class-indexing is shown in Figure 3 where the proposed class-based indexing is incorporated with term, document and class index. The nodes of term, document and class index contain two fields: data and link. While provide the dataset as an input to assign scores for lexical terms, the data field of term index contains a term and the corresponding three outbound link fields where the first link points to the node containing the next term. The second and third link points to the class and document index to find out the relevant class and the relevant documents respectively that the term falls into. The data field of document index contains a document and the corresponding link field points to the node containing the next document. Therefore the combination of term and document index or the combination of term, document and class index, we can generate TF.IDF or TF.IDF.ICF term weighting scheme respectively.

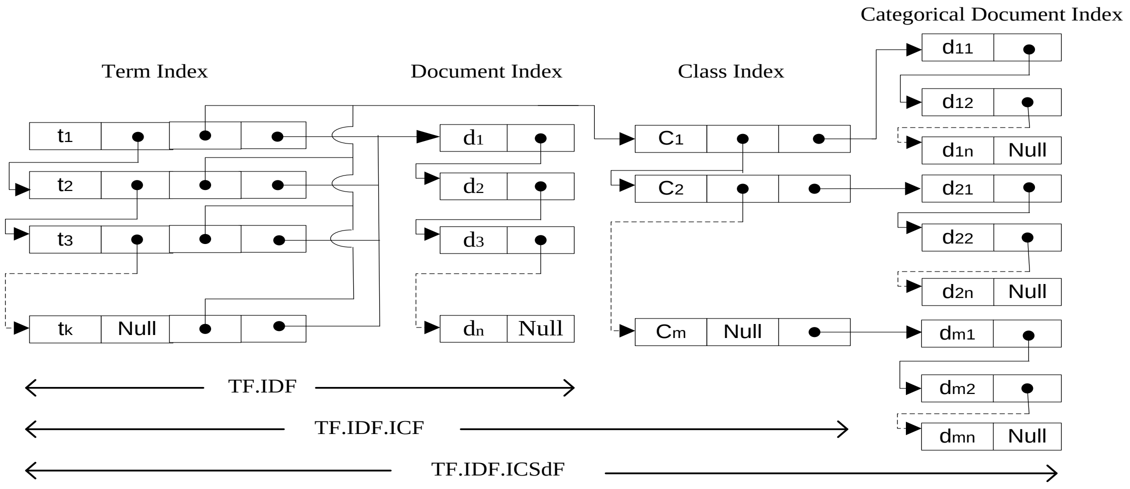


Figure 3. Class-oriented-indexing: the formulation of TF.IDF, TF.IDF.ICF, and TF.IDF. ICS$_{\delta}$F.

In contrast, the data field of class index contains class information and the corresponding two link fields where the first link points to the node containing the next class. The second outbound link field points to the document index to give the identity of a set of documents that a certain term falls into a certain class. Which is later we call the class space density frequency. The effectiveness of this indexing is that, the lexical terms obtain the prior knowledge from class index in the training mode to create vector space model. Using class-indexing based term weighting method, more informative term can be generate to examine the existence of a term not only to compute the occurrence of a term in a certain class but also compute the occurrence of a set of relevant documents in a certain class. From Figure 3 and the above discussions it is noticeable that, the proposed class-oriented-indexing requires more space to store additional data as well as computational cost is higher than document-indexing for the training formulation.

In class-indexing-based term weighting method, terms are examined for frequency and to compute their occurrence in a certain class. They are also used to compute the occurrence of a set of relevant documents that are clustered [2] for a certain term t_i in a certain class c_k. Two factors are considered for category mapping using class-indexing-based term weighting scheme: (1) class-frequency (CF)-based category mapping of each term and (2) class-space-density-based category mapping of each term.

3.2.2 Class-frequency-based Category Mapping: TF.IDF.ICF Term Weighting Scheme

In the classic document-indexing-based VSM, the criterion of numeric representation of text to document vectors are products of local (term frequency) and global (IDF) parameters, that is, TF.IDF. In the class-frequency-based category mapping term weighting scheme, the ICF is multiplied by TF.IDF, generating TF.IDF.ICF. In this method, the existence of a category is indicated by assigning a numerical value 1 when a term is relevant for a certain category c_k, and 0 when a term is not relevant. The TF.IDF.ICF term weighting scheme introduces the concept of category mapping using the ICF function. In ICF, a term that occurs in many categories is not a good discriminator. Consequently, the ICF function gives the lowest score to those terms that appear in multiple classes in class space $C = c_1, c_2, \dots c_m$. Therefore, the numeric representation of a term is the product of term frequency (local parameter), IDF (global parameter), and ICF (categorical global parameter), represented as

$$W_{TF.IDF.ICF}\left(t_i, d_j, c_k\right) = tf_{(t_i,d_j)} \times \left(1 + log\,\frac{D}{d(t_i)}\right) \times \left(1 + log\,\frac{C}{c(t_i)}\right), \tag{17}$$

Its cosine normalization is denoted as

$$W^{norm}_{TF.IDF.ICF}\left(t_i, d_j, c_k\right) = \frac{tf_{(t_i,d_j)} \times (1+log\frac{D}{d(t_i)}) \times (1+log\frac{C}{c(t_i)})}{\sqrt{\sum_{t_i \in d_j; t_i \in c_k} [tf_{(t_i,d_j)} \times (1+log\frac{D}{d(t_i)}) \times (1+log\frac{C}{c(t_i)})]^2}}, \tag{18}$$

Another representation of class-frequency-based category mapping as TF.ICF where IDF is not incorporated, represented as

$$W_{TF.ICF}(t_i, d_j, c_k) = tf_{(t_i, d_j)} \times \left(1 + log\frac{C}{c(t_i)}\right), \tag{19}$$

Its cosine normalization is denoted as

$$W^{norm}_{TF.ICF}(t_i, d_j, c_k) = \frac{tf_{(t_i,d_j)} \times (1 + log\frac{C}{c(t_i)})}{\sqrt{\sum_{t_i \in d_j; t_i \in c_k} [tf_{(t_i,d_j)} \times (1 + log\frac{C}{c(t_i)})]^2}} \tag{20}$$

where C denotes the total number of predefined categories in the collection, $c(t_i)$ is the number of categories in the collection in which term t_i occurs at least once, $\frac{c(t_i)}{C}$ is referred to as the CF, and $\frac{C}{c(t_i)}$ is the ICF of term $t_{i.}$

The TF.IDF.ICF approach is our primary motivation to revise the document-indexing-based term weighting approach and lead to a new class-indexing-based term weighting approach. For a valid term, the evidence of clustered relevant documents, where documents are grouped into a certain category c_k for a certain term t_i, is missing in this method. Such a method gives positive discrimination to rare terms and is biased against frequent terms. Because, the IDF and ICF functions are incorporated with TF and both assigns the lowest score to those terms that appear in multiple documents in document space and multiple categories in class space, respectively. Therefore, we redesigned the ICF and implemented a new inverse class space density frequency (ICS$_\delta$F) method that provides positive discrimination for both rare and frequent terms.

3.2.3 Class-space-density-based Category Mapping: TF.IDF.ICS$_\delta$F Term Weighting Scheme

Both TF.IDF and TF.IDF.ICF term weighting schemes emphasize on rare terms, which favor terms appearing in only a few documents multiplied by the IDF function and a few classes multiplied by the ICF function, respectively. In the class-space-density-based category mapping of each term, the inverse class-space-density frequency (ICS$_\delta$F) is multiplied by the TF.IDF to generate the TF.IDF.ICS$_\delta$F. Because the ICF function gives the lowest score to those terms that appear in multiple classes without any prior knowledge of the class space, a new class-space-density-frequency-based category mapping is proposed. It should be possible to improve the classification performance by addressing the prior knowledge of a certain term in a certain category in the training procedure, in particular, by determining the number of outbound document links for a certain term t_i through a certain category c_k. More specifically, the weight of each term depends not only on its occurrence characteristics in documents and classes but also on its class density (C_δ) where a set of relevant documents are linked with a certain term t_i.

3.2.3.1 Determining the Category Mapping by Class Density of Each Term

In a VSM, documents are usually represented as vectors in n-dimensional space by describing each document d_j by a numerical value of a feature vector. For a certain term t_i, first the class weight is assigned a numerical value between 1 and 0 for overlapping and non-overlapping terms in a certain class c_k, respectively.

$$W_{c_k}(t_i) = \begin{cases} 1, if\ term\ appears\ in\ a\ certain\ category\ that\ t_i\ \in |c_k| \\ \qquad\qquad\qquad 0, otherwise \end{cases}$$

If a certain term t_i falls into certain class c_k, in the next computational process, the number of total outbound document links is calculated through a certain class c_k that the term t_i falls into. The outbound document links represent a collection of tightly clustered documents for a certain term t_i in a certain class. Thus, the class density C_δ is defined as a rate of documents that include the term t_i in the category c_k. In mathematical representation,

$$C_\delta(t_i) = \frac{n_{c_k}(t_i)}{N_{c_k}}, \tag{21}$$

where $nc_k(t_i)$ denotes the number of documents that include the term t_i and are a member of the category c_k, and $\mathrm{Nc_k}$ denotes the total number of documents in a certain category c_k.

3.2.3.2 Determining the Class Space Density of Each Term

In the computational process, the class space density CS_δ is the sum of the outbound class links for a certain term t_i and each class link is measured by class density. In mathematical representation,

$$CS_\delta(t_i) = \sum_{c_k} C_\delta(t_i), \tag{22}$$

Therefore, the inverse class space density frequency is denoted as

$$ICS_{\delta}F(t_i) = log\left(\frac{C}{CS_{\delta}(t_i)}\right), \tag{23}$$

The numeric representation of a term is the product of term frequency (local parameter), IDF (global parameter), and inverse class space density frequency (ICS$_{\delta}$F), which represents the combination of categorical local and global parameters. Therefore, the proposed term weighting scheme TF.IDF.ICS$_{\delta}$F for a certain term t_i in document d_j with respect to category c_k is defined as

$$W_{TF.IDF.ICS_{\delta}F}\left(t_i, d_j, c_k\right) = tf_{(t_i,d_j)} \times \left(1 + log\,\frac{D}{d(t_i)}\right) \times \left(1 + log\,\frac{C}{CS_{\delta}(t_i)}\right), \tag{24}$$

Its cosine normalization is denoted as

$$W^{norm}_{TF.IDF.ICS_{\delta}F}\left(t_i, d_j, c_k\right) = \frac{tf_{(t_i,d_j)}\times(1+log\frac{D}{d(t_i)})\times(1+log\frac{C}{CS_{\delta}(t_i)})}{\sqrt{\sum_{t_i\in d_j;t_i\in c_k}[tf_{(t_i,d_j)}\times(1+log\frac{D}{d(t_i)})\times(1+log\frac{C}{CS_{\delta}(t_i)})]^2}}, \tag{25}$$

We can also present class-space-density-based category mapping as TF.ICS$_{\delta}$F where IDF is not incorporated, represented as

$$W_{TF.ICS_{\delta}F}\left(t_i, d_j, c_k\right) = tf_{(t_i,d_j)} \times \left(1 + log\,\frac{C}{CS_{\delta}(t_i)}\right), \tag{26}$$

Its cosine normalization is denoted as

$$W^{norm}_{TF.ICS_{\delta}F}\left(t_i, d_j, c_k\right) = \frac{tf_{(t_i,d_j)}\times(1+log\frac{C}{CS_{\delta}(t_i)})}{\sqrt{\sum_{t_i\in d_j;t_i\in c_k}[tf_{(t_i,d_j)}\times(1+log\frac{C}{CS_{\delta}(t_i)})]^2}}, \tag{27}$$

where $\frac{CS_{\delta}(t_i)}{C}$ is referred to as the class space density frequency (CS$_{\delta}$F) and $\frac{C}{CS_{\delta}(t_i)}$ is the inverse class space density frequency (ICS$_{\delta}$F) of term t_i. Figure 4 shows an example of creating a class-indexing-based vector space model for a sample of eight documents.

Document Collection

Document1 = { The gold was stolen in a silver truck. }
Document2 = { The gold was stolen in a gold truck. }
Document3 = { The gold truck drove away with gold. }
Document4 = { The gold truck drove away with silver. }
Document5 = { The silver was stolen in a silver truck. }
Document6 = { The silver was stolen. }
Document7 = { The silver truck drove away with silver. }
Document8 = { The silver truck drove away with gold. }

Indexing Step

Class-Indexing	Document-Indexing	Term-Indexing (Raw Term Frequency)				
		drove	gold	silver	stolen	truck
Class1	Document1	0	1	1	1	1
Class1	Document2	0	2	0	1	1
Class1	Document3	1	2	0	0	1
Class1	Document4	1	1	1	0	1
Class2	Document5	0	0	2	1	1
Class2	Document6	0	0	1	1	0
Class2	Document7	1	0	2	0	1
Class2	Document8	1	1	1	0	1
Raw Document Frequency		4	5	7	4	7

Term weighting Step

Term			drove	gold	silver	stolen	truck
log(IDF)			0.602	0.699	0.845	0.602	0.845
log(ICSδF)			0.301	0.204	0.125	0.301	0.058
Class1	Document1	TF.IDF.ICSδF	0	0.143	0.106	0.181	0.049
Class1	Document2	TF.IDF. ICSδF	0	0.285	0.000	0.181	0.049
Class1	Document3	TF.IDF. ICSδF	0.181	0.285	0.000	0.000	0.049
Class1	Document4	TF.IDF. ICSδF	0.181	0.143	0.106	0.000	0.049
Class2	Document5	TF.IDF. ICSδF	0	0.000	0.211	0.181	0.049
Class2	Document6	TF.IDF. ICSδF	0	0.000	0.106	0.181	0.000
Class2	Document7	TF.IDF. ICSδF	0.181	0.000	0.211	0.000	0.049
Class2	Document8	TF.IDF. ICSδF	0.181	0.143	0.106	0.000	0.049

Figure 4. Example of Constructing a Class-indexing Model

3.3 Term Representation: The Effectiveness of Class Space Density Frequency

With the rapid growth of large textual information available digitally, a major difficulty of TC is to compute a statistical weight of a certain term that might be a member of different domain or categories. Therefore, we use two examples to judge the effectiveness of class-space-density frequency in the vector space, especially in the balanced corpus and unbalanced corpus distribution. We assume that a dataset contains of two categories C_1 and C_2. If term t_i appears in any category, the weight of t_i is assigned by binary value, 0 or 1 from Level1 to Level2 in Figure 5 and 6.

$$\text{Tag}_{\text{Level1 to Lebel2}} = \begin{cases} 1, \text{if term appears in a certain class that } t_i \in |c_k| \\ 0, \text{otherwise} \end{cases}$$

In the next computational step from Level2 to Level3 in Figure 5 and 6, the weight of t_i is $1/n$ when it appears in relevant documents in a certain class domain otherwise zero. Where n is the total number of documents in a certain class.

$$\text{Tag}_{\text{Leve2 to Lebel3}} = \begin{cases} \frac{1}{n}, \text{if term appears in a certain class that } t_i \in |c_k| \text{ and } t_i \in |d_j| \\ 0, \text{otherwise} \end{cases}$$

Example1: Balanced Corpus Distribution

In the balanced corpus distribution, we assume that a dataset contains of two categories C_1 and C_2. A term t_1 appears in two categories C_1 and C_2. C_1 and C_2, each contain four documents as a balance corpus distribution. The three and two dotted rectangular area C_1 and C_2 respectively, in Figure 5 indicates that these documents are relevant for a term t_1. The class density of term t_1 in category C_1 is

$C_\delta(t_1) = \frac{n_{C_1}(t_1)}{N_{C_1}} = \frac{3}{4} = 0.75$, and

The class density of term t_1 in category C_2 is

$C_\delta(t_1) = \frac{n_{C_2}(t_1)}{N_{C_2}} = \frac{2}{4} = 0.50$,

Therefore, the class space density of term t_1 is

$CS_\delta(t_1) = \sum C_\delta(t_1) = 0.75 + 0.50 = 1.25.$

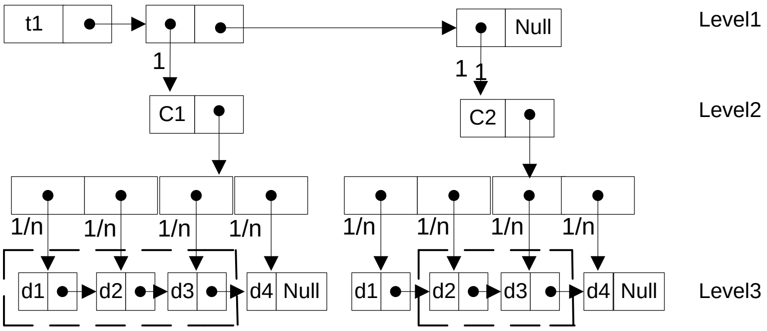


Figure 5. Term representation in class-space-density-frequency indexing: balanced corpus distribution

Example2: Unbalanced Corpus Distribution

In the unbalanced corpus distribution, we assume that a dataset contains of two categories *C*$_1$ and *C*$_2$. A term *t*$_1$ appears in two categories *C*$_1$ and *C*$_2$. *C*$_1$ and *C*$_2$, contains five and three documents respectively, as an unbalanced corpus distribution. The three and two dotted rectangular area *C*$_1$ and *C*$_2$ respectively, in Figure 6 indicates that these documents are relevant for a term *t*$_1$.

The class density of term *t*$_1$ in category *C*$_1$ is

$C_\delta(t_1) = \frac{n_{C_1}(t_1)}{N_{C_1}} = \frac{3}{5} = 0.60$, and

The class density of term *t*$_1$ in category *C*$_2$ is

$C_\delta(t_1) = \frac{n_{C_2}(t_1)}{N_{C_2}} = \frac{2}{3} = 0.67$,

Therefore, the class space density of term *t*$_1$ is

$CS_\delta(t_1) = \sum C_\delta(t_1) = 0.60 + 0.67 = 1.27.$

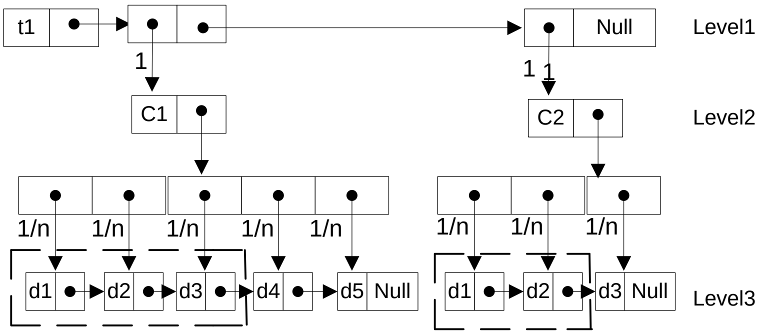


Figure 6. Term representation in class-space-density-frequency indexing: unbalanced corpus distribution

From the above computation, it is noticeable that the score of the class density of category C_1 is higher than that of C_2, and C_2 is higher than C_1 in terms of balance and unbalance corpus distribution. Therefore, in the training vector space, while we compute the average score of a certain term t_i in a certain category c_k, the class density will cause the greatest possible separation between the categories because it can make use the categories to explore the outbound document links through a certain category for a certain term.

In contrast, since the term t_1 appears in two categories in Figure 5 and 6, therefore, without explore the categories of C_1 and C_2, the ICF score will be log(2/2) = 0; because the ICF only conveys the information whether a term belongs to a category or not. In other words, we assume that t_1 appears in almost all documents in category 1 and in only one document in category 2. From the initial information, it is comprehensible that the term t_1 belongs to category 1, but according to the ICF function, because the term t_1 appears in two categories, the ICF score will be zero and fails to provide the greatest possible separation between the categories. Thus, the ICF function provides positive discrimination for rare terms which may not include in a certain category domain. It is noticeable that, the ICS$_\delta$F function provides a judgment score that is not biased against frequent or infrequent terms.

3.4 Classifier

Machine learning method constructs a classification model to predict the category of new test documents by learning the statistic of training data. In the machine learning workbench, some classifiers [46, 50] like centroid-based, support vector machine (SVM), and naïve bayes (NB) have achieved great success in TC. In order to evaluate the effects of the proposed term weighting approaches over existing weighting methods, these three classification schemes are used in three different datasets.

3.4.1 Centroid Classifier

In this study, for simplicity and linearity, we implement a centroid classifier [9, 40, 41], a commonly used method, to judge the effectiveness of the proposed class-indexing-based term weighting approaches and to compare it with the conventional TF.IDF and different term weighting approaches. In the VSM, each document d_j is considered to be a vector in the term space. To calculate the centroid of a certain class c_k, add the training set of document vectors d_j ($j = 1, 2\ldots n$) in the class c_k ($k = 1, 2\ldots m$):

Sum centroid, $C_k^{sum} = \sum_{d\in c_k} d_j$, (28)

The normalized version of C_k^{sum} is as follows:

Normalized centroid, $C_k^{norm} = \frac{C_k^{sum}}{\|C_k^{sum}\|_2}$, (29)

where $\|C_k^{sum}\|_2 \left\|C_k^{sum}\right\|_2$ is defined as 2-norm vector $C_k^{sum}\, C_k^{sum}$.

Next, calculate the similarity between a query document and each normalized centroid class vector by inner-product measure, as follows:

$sim\left(d_j, c_k\right) = d_j . C_k^{norm}$ (30)

Consequently, the test vector (or query vector) d_j is assigned to the class level c_k whose class prototype vector is very similar to the query vector in performing the classification task.

$L(C_k) = argmax_{c_k\in C}\left(d_j . C_k^{norm}\right)$. (31)

3.4.2 Naïve Bayes Classifier

Naïve Bayes (NB) [50] classifier is one of the oldest formal classification algorithms, and widely used classification algorithm in the field of ATC. In Bayesian model, the assumption is based on a prior and posterior probability. Finding the probability of a certain document type $d_j \epsilon C$ can only be based on the observation t_i. The conditional probability $P(C|t_i)$ can be written according to Bayes' rule:

$$P(C|t_i) = \frac{P(t_i|C)P(C)}{P(t_i)}, \qquad (32)$$

Since, the denominator does not depend on the category, we can therefore omit the probability $P(t_i)$. The probability $P(t_i|C)$can be estimated as:

$$P(t_i|C_k) = \prod_{i=1}^{m} P(t_i|C_k), \qquad (33)$$

By assuming that each term follows a probability distribution function for a normal distribution with mean μ and standard deviation σ in each category c, therefore the Eq. 33 can be written as:

$$P(t_i|C) = \prod_{i=1}^{m} P\left(t_i; \mu_{i,c}; \sigma_{i,c}\right),$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_{i,c}{}^2}} e^{-\frac{(t_i-\mu_{i,c})^2}{2\sigma_{i,c}^2}}, \qquad (34)$$

Given this probability for a single term, the logarithm of the probability for all *m* terms in the data is

$$\ln P(t_i|C_k) = \ln \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_{i,c}{}^2}} e^{-\frac{(t_i-\mu_{i,c})^2}{2\sigma_{i,c}^2}}, \qquad (35)$$

$$= \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma_{i,c}{}^2}} e^{-\frac{(t_i-\mu_{i,c})^2}{2\sigma_{i,c}^2}}, \qquad (36)$$

$$= -\frac{m}{2}\ln(2\pi) - \frac{m}{2}\ln\left(\sigma_{i,c}^2\right) - \frac{m}{2}\sum_{i=1}^{m} \frac{\left(t_i-\mu_{i,c}\right)^2}{\sigma_{i,c}^2}, \qquad (37)$$

The prior probability $P(C)$ of a certain class is estimated using a Laplacean prior as:

$$P(C) = \frac{1+N_{t,c}}{D+N_c}, \tag{38}$$

$N_{t,c}$ is the number of documents in a certain class C_k, D is the total number of documents and N_c is the total number of classes.

3.4.3 Support Vector machines

In the machine learning approaches, support vector machines (SVMs) are considered one of the most robust and accurate methods among all well-known algorithms [45]. Therefore, as a third learning classifier, SVM based classification toolbox SVM_Light[4] (Joachims, 1998, 1999, 2002) is used in this experiment. All parameters were left at default values. The parameter *–c* was set to 1.0, which is considered as a default setting in this toolbox.

3.5 Experiments and Evaluations

In this section, we provide empirical evidence for the effectiveness of the proposed term weighting approaches over eight different term weighting approaches. The experiment results show the consistency of the proposed term weighting approaches outperforms the conventional approaches in high-dimensional vector space consisting of rare terms and in the comparatively low-dimensional vector space where reduce the rare terms using threshold setting through a certain category c_k from the training data set.

3.5.1 Experimental Datasets

To evaluate the performance of proposed term weighting approaches with existing baseline approaches, we conducted our experiments using Reuters-21578[5], 20

[4] Available at http://svmlight.joachims.org/svm_multiclass.html

[5] Available at http://disi.unitn.it/moschitti/corpora.htm

Newsgroups[6], and RCV1-v2/LYRL2004[7] which is widely used benchmark collections in the classification task.

3.5.1.1 Reuters-21578 dataset

In Reuters-21578, the ten top-sized categories of Apte' split are adopted, which splits data into a test set and a training set. Because the system is evaluated with 10-fold cross validation, we merged the training and testing documents together. The Reuters-21578 corpus contains 9976 documents.

3.5.1.2 20Newsgroups dataset

The second dataset that we used in this experiment is the 20 Newsgroups, which is a popular dataset to use against machine learning techniques such as TC and text clustering. It contains approximately 18,828 news articles across 20 different newsgroups. For convenience, we call the 20 categories Atheism, CompGraphics, CompOsMsWindowsMisc, CompSysIbmPcHardware, CompSysMacHardware, CompWindowsx, MiscForsale, RecAutos, RecMotorcycles, RecSportBaseBall, RecSportHockey, SciCrypt, SciElectronics, SciMed, SciSpace, SocReligionChristian, TalkPoliticsGuns, TalkPoliticsMideast, TalkPoliticsMisc, and TalkReligionMisc as "Ath," "CGra," "CMWM," "CSIPH," "CSMH," "CWin," "MFor," "RAuto," "RMot," "RSB," "RSH," "SCry," "SEle," "SMed," "SSpa," "SRChr," "TPG," "TPMid," "TPMisc," and "TRMi," respectively.

We employ the commonly used 10-fold cross validation technique in which the Reuters-21578 and 20 Newsgroups datasets are randomly divided into 10-fold. Each turn on one data fold is used for testing and the remaining folds are used for training. Table 1, 2 shows the description of only one of the possible sample datasets for the Reuters-21578 and 20 Newsgroups respectively.

[6] Available at http://people.csail.mit.edu/jrennie/20Newsgroups/

[7] Available at http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

Table 1. One sample Reuters-21578 dataset fold from 10-fold cross validation

Dataset	Category				
	acq	corn	crude	earn	grain
# Training	2129	214	521	3568	524
# Testing	236	23	57	396	58
	interest	money	ship	trade	wheat
# Training	431	646	258	438	255
# Testing	47	71	28	48	28

Table 2. One sample 20Newsgroups dataset fold from 10-fold cross validation

Dataset	Category				
	Ath	CGra	CWin	CSIPH	CSMH
# Training	719	876	882	883	865
# Testing	80	97	98	99	96
	CMWM	MFor	RAuto	RMot	RSB
# Training	888	875	891	895	895
# Testing	98	97	99	99	99
	RSH	SCry	SEle	SMed	SSpa
# Training	900	892	883	891	888
# Testing	99	99	98	99	99
	SRChr	TPG	TPMid	TPMi	TrMi
# Training	898	819	846	698	565
# Testing	99	91	94	77	63

3.5.1.3 RCV1-v2 dataset/LYRL2004

The RCV1 [22] dataset, RCV1-v2/LYRL2004 is adopted, which contains a total of 804414 documents with 103 categories from four parent topics. As single-label in concern in this study, therefore we extract all the documents which are labeled with at least once. We found that only approximate 23000 documents out of 804414 are labeled with at least once. Beside this, we therefore, considered a document which is labeled with two categories a parent with child category.

Then we removed the parent category and child category is assigned in order to produce single-label classification for each document. From RCV1-v2/LYRL2004, a single topic is assigned a total of 229221 documents which falls into 54 different categories. We keep the same split, the first 23149 documents as for training and the remainder 206072 documents is for testing. Table 3 shows the description of training and testing split over RCV1-v2 dataset.

Table 3. RCV1-v2 dataset training and testing split

Dataset	Category					
	C11	C12	C13	C14	C15	C16
# Training	729	11	317	163	3	28
# Testing	5839	68	2263	1557	31	333
	C17	C21	C22	C23	C24	C31
# Training	49	538	422	42	891	1112
# Testing	467	4702	2180	451	6711	10842
	C32	C33	C34	C41	C42	CCAT
# Training	45	772	7	9	5	28
# Testing	662	5808	66	175	25	414
	E11	E12	E13	E14	E21	E31
# Training	453	313	3	9	5	38
# Testing	3438	2409	18	89	60	274
	E51	E61	E71	ECAT	G15	GCAT
# Training	30	23	559	3	173	2449
# Testing	212	162	4505	56	1113	21474
	GCRIM	GDEF	GDIP	GDIS	GENT	GENV
# Training	836	63	968	210	155	60
# Testing	5697	488	8141	1881	1024	482
	GFAS	GHEA	GJOB	GODD	GPOL	GPRO
# Training	3	116	4	120	1182	132
# Testing	71	957	55	979	10172	550
	GREL	GSCI	GSPO	GTOUR	GVIO	GWEA
# Training	75	70	3030	8	1287	103
# Testing	320	1003	31732	61	9308	910
	GWELF	M11	M12	M13	M14	MCAT
# Training	10	3657	1382	13	434	12
# Testing	73	36521	15258	140	3440	157

3.5.2 Feature Selection by Threshold Setting

This study emphasizes on a novel class-indexing-based term weighting scheme to enhance the classification task. Therefore, we do not adopt feature selection techniques such as information gain, mutual information, chi-square test, and document frequency as feature selection criterion [1, 2, 26, 27, 36]. Nevertheless, we first normalize all the documents in the document space using several preprocessing steps, including converting uppercase letters in a token to lowercase, removing punctuation, eliminating word on the smart stop word list that contains 571 words, and reducing inflected words to their root form using stemming. After normalizing the documents, we perform a local feature (term) selection by term frequency in a certain category (TFC). In this approach, we rank and assess all features based on their appearance in a certain category c_k, and arrange them in descending order. In this way, we select only some of the features and remove others by setting common thresholds for each category in the two datasets. The idea behind using a threshold is to create a high-dimensional vector space (including numerous rare terms) and a comparatively low-dimensional vector space (removing rare terms) to judge the performance of the proposed TF.IDF.ICF and TF.IDF.ICS$_\delta$F approaches compared with the traditional TF.IDF approach in both high-dimensional and low-dimensional vector space. Setting the thresholds, we conducted experiments in two different sessions: one in which the high-dimensional vector space is considered and other in which the comparatively low-dimensional vector space is considered.

The high-dimensional and low-dimensional vector spaces are generated by setting the threshold $\rho = 3$ and $\rho = 10$, respectively, in each category of the Reuters-21578 and 20 Newsgroups datasets. In the high-dimensional vector space, the threshold $\rho = 3$ eliminates all the terms that appear for not more than two times in a certain class c_k of the two datasets. Therefore, we investigate the effects of the proposed and traditional term weighting schemes in this vector space with the centroid classifier. In contrast, the threshold $\rho = 10$ is assigned to the comparatively low-dimensional vector space and it eliminates all the terms that appear for not more than nine times in a certain category of the datasets. This action judges the effects of the proposed and traditional term weighting schemes in this vector space with the centroid classifier. With the thresholds of $\rho = 3$ and $\rho = 10$ set in the Reuters-21578 and 20 Newsgroups datasets, the total number of unique terms in term space $T_i = \{t_1, t_2, \dots t_n\}$ decreased from 8,974 to 8,286 and 19,634 to 12,377, respectively. Thus, we manage the vector space to eliminate a certain level of infrequent terms by setting the thresholds, in order to justify the effectiveness of proposed and traditional term weighting approaches in high-dimensional and comparatively low-dimensional vector space.

3.5.3 Results with High-Dimensional Vector Space

The high-dimensional vector space is generated by setting the threshold to $\rho = 3$ in the Reuters-21578 and 20 Newsgroups datasets. Tables 4, 5, and 6 show F_1 measure $F_1(C_k)$

of a certain category in the Reuters-21578 dataset over centroid, NB, and SVM classifier respectively. In these tables, in comparing with other weighting methods, TF.IDF.ICF and TF.IDF.ICS$_\delta$F is outperformed over other methods in centroid and SVM classifier respectively. The TF.ICF is showing its own superiority in NB classifier. In contrast, Tables 7, 8, and 9 show F_1 measure $F_1(C_k)$ of a certain category in the 20 Newsgroups dataset over centroid, NB, and SVM classifier respectively. The result shows that TF.IDF.ICS$_\delta$F shows its own superiority in Centroid and SVM methods. It achieves higher accuracy 19 of 20 in centroid and 20 of 20 categories in SVM over other methods. The TF.IDF is showing its own superiority in NB classifier.

Table 4. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 3 over the Centroid classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
acq	0.707	0.825	0.799	0.784	0.814
corn	0.076	0.453	0.245	0.205	0.455
crude	0.389	0.793	0.670	0.639	0.763
earn	0.799	0.891	0.852	0.765	0.872
grain	0.033	0.307	0.263	0.163	0.320
interest	0.439	0.659	0.527	0.546	0.617
money-fx	0.445	0.701	0.509	0.428	0.633
ship	0.305	0.675	0.625	0.587	0.667
trade	0.289	0.716	0.507	0.462	0.758
wheat	0.193	0.495	0.293	0.189	0.525
macro-F_1	0.367	0.651	0.532	0.477	0.642
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
acq	0.897	0.877	0.900	0.917	**0.930**
corn	0.432	0.480	0.428	**0.468**	0.391
crude	0.820	0.820	0.825	**0.829**	0.826
earn	0.936	0.912	0.933	0.946	**0.958**
grain	0.375	0.396	0.365	**0.408**	0.377
interest	0.702	0.685	0.698	0.712	**0.713**
money-fx	0.695	0.710	0.702	**0.704**	0.682
ship	0.732	0.730	0.738	0.758	**0.760**
trade	0.819	0.797	0.812	**0.836**	0.828
wheat	0.520	0.535	0.526	**0.540**	0.511
macro-F_1	0.693	0.694	0.692	**0.707**	0.697

Table 5. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 3 over the NB classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
acq	**0.799**	0.786	0.795	0.759	0.783
corn	0.237	0.249	0.243	0.104	0.254
crude	0.620	0.599	0.635	0.615	0.631
earn	0.901	0.907	**0.914**	0.827	0.897
grain	0.546	0.520	0.547	0.465	0.535
interest	0.516	0.529	0.531	0.500	0.501
money-fx	0.572	0.580	0.591	0.482	0.573
ship	0.552	0.577	0.572	0.521	0.608
trade	0.579	0.596	0.582	0.466	0.595
wheat	0.416	0.434	0.390	0.161	0.424
macro-F_1	0.574	0.578	0.580	0.490	0.580
	TF.IDF	TF.ICF	TF. ICS$_{\delta}$F	TF.IDF.ICF	TF.IDF.ICS$_{\delta}$F
acq	0.750	0.783	0.760	0.753	0.751
corn	0.242	**0.286**	0.255	0.206	0.215
crude	0.575	**0.645**	0.597	0.620	0.603
earn	0.886	0.905	0.885	0.886	0.879
grain	0.517	**0.582**	0.519	0.537	0.515
interest	0.486	**0.567**	0.513	0.501	0.487
money-fx	0.572	**0.614**	0.574	0.583	0.574
ship	0.558	**0.617**	0.578	0.484	0.481
trade	0.577	**0.606**	0.571	0.593	0.566
wheat	0.435	0.459	0.430	0.437	**0.525**
macro-F_1	0.560	**0.606**	0.568	0.560	0.560

Table 6. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 3 over the SVM classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
acq	0.812	0.880	0.839	0.714	0.758
corn	0.139	0.308	0.163	0.253	0.207
crude	0.654	0.768	0.709	0.778	0.584
earn	0.886	0.929	0.913	0.879	0.806
grain	0.394	0.443	0.368	0.403	0.386
interest	0.495	0.657	0.515	0.569	0.593
money-fx	0.598	0.738	0.605	0.449	0.631
ship	0.557	0.456	0.611	0.568	0.322
trade	0.539	0.743	0.572	0.589	0.652
wheat	0.056	0.265	0.082	0.021	0.216
macro-F_1	0.513	0.619	0.538	0.522	0.516
	TF.IDF	TF.ICF	TF. ICS$_{\delta}$F	TF.IDF.ICF	TF.IDF.ICS$_{\delta}$F
acq	0.907	0.855	0.900	0.942	**0.947**
corn	0.235	0.172	0.213	**0.243**	0.240
crude	0.803	0.728	0.775	0.849	**0.851**
earn	0.922	0.862	0.918	0.960	**0.975**
grain	0.408	0.450	0.400	0.472	**0.494**
interest	0.648	0.627	0.629	0.644	**0.685**
money-fx	0.749	0.678	0.741	0.763	**0.771**
ship	0.578	0.496	0.530	0.686	**0.714**
trade	0.744	0.723	0.752	0.783	**0.821**
wheat	0.278	0.259	0.203	**0.302**	0.269
macro-F_1	0.627	0.585	0.606	0.664	**0.677**

Table 7. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 3 over the Centroid classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
Ath	0.444	0.413	0.727	0.792	0.633
CGra	0.353	0.516	0.532	0.443	0.577
CMWM	**0.409**	0.403	0.110	0.264	0.090
CSIPH	0.366	0.493	0.451	0.479	0.523
CSMH	0.332	0.624	0.590	0.536	0.636
CWin	0.432	0.536	0.621	0.464	0.549
MFor	0.330	0.637	0.368	0.362	0.637
RAuto	0.372	0.715	0.726	0.636	0.749
RMot	0.315	0.755	0.587	0.676	0.529
RSB	0.447	0.621	0.728	0.672	0.640
RSH	0.531	0.360	0.483	0.823	0.304
SCry	0.669	0.726	0.801	0.859	0.786
SEle	0.255	0.512	0.576	0.167	0.414
SMed	0.264	0.723	0.744	0.695	0.742
SSpa	0.530	0.728	0.774	0.739	0.786
SRChr	0.243	0.689	0.728	0.477	0.694
TPG	0.469	0.673	0.728	0.733	0.723
TPMid	0.619	0.773	0.676	0.863	0.806
TPMi	0.251	0.531	0.590	0.461	0.548
TrMi	0.174	0.391	0.502	0.444	0.376
macro-F$_1$	0.390	0.591	0.602	0.579	0.587
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
Ath	0.751	0.769	0.747	**0.822**	**0.822**
CGra	0.750	0.792	0.742	0.799	**0.806**
CMWM	0.139	0.142	0.143	0.143	0.175
CSIPH	0.680	0.701	0.659	0.718	**0.735**
CSMH	0.786	0.812	0.772	0.830	**0.834**
CWin	0.676	0.687	0.654	0.725	**0.756**
MFor	0.776	0.778	0.767	0.825	**0.828**
RAuto	0.884	0.899	0.875	**0.931**	**0.931**
RMot	0.929	0.940	0.923	0.962	**0.963**
RSB	0.848	0.933	0.835	**0.952**	**0.952**
RSH	0.797	0.927	0.766	**0.932**	**0.934**
SCry	0.878	0.926	0.875	0.923	**0.925**
SEle	0.767	0.691	0.761	0.832	**0.835**
SMed	0.907	0.934	0.908	0.946	**0.947**
SSpa	0.905	0.925	0.899	0.930	**0.930**
SRChr	0.841	0.850	0.834	**0.872**	**0.872**
TPG	0.839	0.861	0.837	0.869	**0.870**
TPMid	0.901	0.886	0.902	**0.938**	**0.938**
TPMi	0.725	0.770	0.723	**0.804**	**0.804**
TrMi	0.599	0.616	0.607	**0.645**	**0.645**
macro-F$_1$	0.769	0.792	0.761	0.820	**0.825**

Table 8. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 3 over the NB classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
Ath	**0.815**	0.791	0.803	0.781	0.810
CGra	0.802	0.780	0.818	0.772	0.760
CMWM	0.799	0.808	0.787	0.744	0.761
CSIPH	0.813	0.819	0.813	0.726	0.812
CSMH	0.815	0.839	0.852	0.778	0.836
CWin	0.822	0.849	0.824	0.803	0.812
MFor	0.824	0.800	0.839	0.687	0.802
RAuto	0.824	0.812	0.840	0.792	0.831
RMot	0.867	0.849	0.868	0.804	0.871
RSB	0.856	0.863	0.889	0.804	0.883
RSH	0.890	0.898	0.890	0.844	0.902
SCry	0.832	0.837	0.834	0.856	0.842
SEle	0.802	0.815	0.803	0.729	0.804
SMed	0.819	0.807	0.836	0.839	0.807
SSpa	0.788	0.763	0.785	**0.828**	0.769
SRChr	0.834	0.848	0.809	**0.835**	0.833
TPG	0.789	0.762	0.789	0.776	0.776
TPMid	0.756	0.776	0.780	**0.801**	0.775
TPMi	0.692	0.652	0.705	0.673	0.676
TrMi	0.760	0.730	0.761	0.381	0.731
macro-F$_1$	0.810	0.805	0.816	0.763	0.805
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
Ath	0.803	0.808	0.808	0.787	0.790
CGra	**0.822**	0.805	0.784	0.791	0.768
CMWM	**0.822**	0.821	0.797	0.795	0.788
CSIPH	**0.839**	0.837	0.820	0.809	0.805
CSMH	0.854	**0.868**	0.858	0.838	0.843
CWin	**0.864**	0.848	0.832	0.835	0.823
MFor	**0.838**	0.799	0.821	0.787	0.789
RAuto	**0.854**	0.848	0.842	0.833	0.832
RMot	0.875	0.878	0.861	0.870	0.863
RSB	0.891	0.892	0.896	0.886	0.871
RSH	**0.910**	0.893	0.903	0.890	0.889
SCry	**0.862**	0.861	0.858	0.854	0.847
SEle	**0.832**	0.828	0.822	0.813	0.804
SMed	**0.841**	0.828	0.826	0.819	0.800
SSpa	0.802	0.793	0.785	0.793	0.769
SRChr	0.830	0.802	0.808	0.810	0.797
TPG	0.792	**0.806**	0.801	0.794	0.789
TPMid	0.767	0.770	0.779	0.783	0.755
TPMi	0.692	0.703	**0.709**	0.699	0.683
TrMi	**0.742**	0.712	0.741	0.713	0.723
macro-F$_1$	**0.827**	0.820	0.818	0.810	0.801

Table 9. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 3 over the SVM classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
Ath	0.786	0.632	0.766	0.793	0.477
CGra	0.709	0.606	0.708	0.575	0.585
CMWM	0.649	0.667	0.614	0.632	0.195
CSIPH	0.651	0.606	0.623	0.523	0.455
CSMH	0.713	0.713	0.698	0.726	0.599
CWin	0.713	0.671	0.728	0.318	0.460
MFor	0.635	0.724	0.593	0.534	0.587
RAuto	0.770	0.766	0.806	0.633	0.608
RMot	0.802	0.832	0.837	0.904	0.637
RSB	0.833	0.737	0.844	0.910	0.617
RSH	0.835	0.799	0.830	0.938	0.680
SCry	0.885	0.702	0.884	0.954	0.763
SEle	0.640	0.536	0.671	0.258	0.298
SMed	0.822	0.808	0.840	0.881	0.397
SSpa	0.825	0.804	0.844	0.925	0.706
SRChr	0.761	0.725	0.774	0.705	0.673
TPG	0.764	0.755	0.813	0.858	0.465
TPMid	0.899	0.798	0.885	0.574	0.682
TPMi	0.693	0.632	0.740	0.764	0.397
TrMi	0.534	0.223	0.532	0.566	0.168
macro-F_1	0.746	0.687	0.752	0.699	0.522
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
Ath	0.794	0.812	0.790	0.889	**0.905**
CGra	0.827	0.823	0.826	0.891	**0.900**
CMWM	0.796	0.805	0.792	0.873	**0.873**
CSIPH	0.766	0.767	0.756	0.839	**0.856**
CSMH	0.841	0.844	0.839	0.899	**0.915**
CWin	0.834	0.842	0.837	0.912	**0.913**
MFor	0.828	0.811	0.827	0.867	**0.897**
RAuto	0.886	0.902	0.888	0.946	**0.950**
RMot	0.943	0.950	0.942	0.969	**0.978**
RSB	0.935	0.947	0.936	0.971	**0.977**
RSH	0.920	0.919	0.919	0.968	**0.979**
SCry	0.871	0.869	0.872	0.947	**0.963**
SEle	0.797	0.787	0.792	0.886	**0.908**
SMed	0.926	0.918	0.925	0.960	**0.962**
SSpa	0.900	0.921	0.905	0.951	**0.958**
SRChr	0.831	0.830	0.826	0.903	**0.935**
TPG	0.860	0.870	0.856	0.924	**0.946**
TPMid	0.897	0.882	0.898	0.969	**0.971**
TPMi	0.812	0.804	0.805	0.901	**0.924**
TrMi	0.493	0.446	0.465	0.718	**0.809**
macro-F_1	0.838	0.837	0.835	0.909	**0.926**

The above results show that the class-indexing-based TF.IDF.ICS$_\delta$F and TF.IDF.ICF, both term weighting approaches outperformed the conventional approaches. This is especially the case for the TF.IDF.ICF approach in which both IDF and the ICF functions give positive discrimination on rare terms and the high-dimensional vector space is considered with numerous rare terms in the term space. Therefore, by setting the threshold at $\rho = 3$, the TF.IDF.ICF and TF.IDF.ICS$_\delta$F term weighting approaches are showing its own superiority over the centroid and SVM classifier respectively.

3.5.4 Results with Comparatively Low-Dimensional Vector Space

The low-dimensional vector space is generated by setting the threshold to $\rho = 10$ in the two datasets, Reuters-21578 and 20 Newsgroups. Tables 10, 11, and 12 show F_1 measure $F_1(C_k)$ of a certain category in the Reuters-21578 dataset over centroid, NB, and SVM classifier respectively.

Table 10. Performance on F1 measure in the Reuters-21578 dataset with threshold $\rho = 10$ over the Centroid classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
acq	0.717	0.834	0.810	0.800	0.817
corn	0.083	0.444	0.281	0.252	0.445
crude	0.403	0.793	0.699	0.748	0.772
earn	0.853	0.898	0.861	0.871	0.873
grain	0.017	0.360	0.283	**0.479**	0.318
interest	0.363	0.641	0.527	0.547	0.633
money-fx	0.463	0.656	0.645	0.576	0.685
ship	0.276	0.639	0.624	0.751	0.668
trade	0.273	0.747	0.589	0.865	0.758
wheat	0.189	0.447	0.336	0.187	0.522
macro-F_1	0.364	0.646	0.565	0.608	0.649
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
acq	0.906	0.878	0.903	0.925	**0.934**
corn	0.508	0.509	0.462	**0.539**	0.477
crude	0.836	0.838	0.837	**0.849**	0.838
earn	0.938	0.911	0.933	0.950	**0.960**
grain	0.372	0.408	0.407	0.424	0.435
interest	0.788	0.767	0.790	0.856	**0.892**
money-fx	0.829	0.856	0.837	**0.898**	0.891
ship	0.763	0.759	0.761	0.783	**0.785**
trade	0.833	0.826	0.824	**0.887**	0.853
wheat	0.544	0.540	0.536	**0.559**	0.529
macro-F_1	0.732	0.729	0.729	**0.767**	0.759

Table 11. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 10 over the NB classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
acq	0.858	0.859	**0.880**	0.863	0.867
corn	**0.496**	0.448	0.392	0.117	0.491
crude	0.736	0.730	**0.781**	0.712	0.739
earn	0.932	0.937	**0.946**	0.892	0.931
grain	0.673	0.668	0.660	0.577	0.655
interest	0.664	0.729	0.679	0.605	0.714
money-fx	**0.818**	0.792	0.778	0.638	0.782
ship	0.794	0.790	0.794	0.637	0.783
trade	**0.758**	0.728	0.715	0.689	0.712
wheat	**0.620**	0.599	0.558	0.211	0.579
macro-F_1	0.735	0.728	0.718	0.594	0.725
	TF.IDF	TF.ICF	TF. ICS$_{\delta}$F	TF.IDF.ICF	TF.IDF.ICS$_{\delta}$F
acq	0.797	0.865	0.852	0.867	0.858
corn	0.420	0.453	0.435	0.448	0.454
crude	0.578	0.766	0.724	0.738	0.748
earn	0.876	0.939	0.933	0.932	0.931
grain	0.618	**0.696**	0.680	0.679	0.671
interest	0.630	**0.738**	0.701	0.668	0.708
money-fx	0.711	0.798	0.771	0.749	0.751
ship	0.720	**0.818**	0.741	0.727	0.735
trade	0.563	0.736	0.711	0.684	0.692
wheat	0.573	0.593	0.582	0.587	0.580
macro-F_1	0.649	**0.740**	0.713	0.708	0.713

Table 12. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 10 over the SVM classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
acq	0.827	0.867	0.853	0.791	0.758
corn	0.219	0.324	0.182	0.416	0.126
crude	0.705	0.755	0.783	0.865	0.592
earn	0.909	0.920	0.923	0.919	0.807
grain	0.404	0.353	0.405	0.554	0.420
interest	0.643	0.712	0.537	0.640	0.636
money-fx	0.737	0.793	0.741	0.690	0.671
ship	0.638	0.407	0.710	0.740	0.311
trade	0.638	0.722	0.681	0.889	0.677
wheat	0.151	0.229	0.229	0.068	0.319
macro-F_1	0.587	0.608	0.605	0.657	0.532
	TF.IDF	TF.ICF	TF. ICS$_{\delta}$F	TF.IDF.ICF	TF.IDF.ICS$_{\delta}$F
acq	0.915	0.887	0.912	0.960	**0.961**
corn	0.305	0.311	0.277	0.359	**0.419**
crude	0.824	0.778	0.792	0.876	**0.879**
earn	0.928	0.879	0.926	0.972	**0.981**
grain	0.465	0.450	0.467	0.573	**0.629**
interest	0.757	0.733	0.753	0.863	**0.907**
money-fx	0.874	0.814	0.866	0.911	**0.923**
ship	0.617	0.569	0.570	0.755	**0.788**
trade	0.780	0.744	0.783	0.864	**0.882**
wheat	0.294	0.272	0.268	0.374	**0.403**
macro-F_1	0.676	0.644	0.661	0.751	**0.777**

Table 13. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 10 over the Centroid classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
Ath	0.609	0.650	0.656	**0.839**	0.668
CGra	0.439	0.595	0.531	0.393	0.629
CMWM	0.158	0.109	0.097	**0.298**	0.182
CSIPH	0.432	0.538	0.509	0.328	0.559
CSMH	0.399	0.663	0.609	0.605	0.672
CWin	0.508	0.549	0.546	0.614	0.584
MFor	0.649	0.633	0.599	0.389	0.682
RAuto	0.451	0.777	0.773	0.717	0.796
RMot	0.688	0.650	0.650	0.759	0.663
RSB	0.639	0.621	0.629	0.799	0.657
RSH	0.245	0.238	0.346	0.900	0.343
SCry	0.728	0.814	0.813	0.913	0.823
SEle	0.507	0.489	0.465	0.228	0.548
SMed	0.679	0.798	0.786	0.790	0.827
SSpa	0.420	0.795	0.789	0.801	0.825
SRChr	0.701	0.712	0.713	0.641	0.756
TPG	0.647	0.756	0.744	0.807	0.786
TPMid	0.716	0.820	0.807	0.884	0.842
TPMi	0.523	0.567	0.567	0.588	0.626
TrMi	0.369	0.471	0.466	0.595	0.478
macro-F$_1$	0.525	0.612	0.605	0.644	0.647
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
Ath	0.761	0.775	0.761	0.635	0.829
CGra	0.745	0.797	0.744	0.606	**0.811**
CMWM	0.131	0.134	0.131	0.128	0.147
CSIPH	0.665	0.696	0.663	0.540	**0.718**
CSMH	0.778	0.809	0.778	0.674	**0.829**
CWin	0.662	0.683	0.663	0.568	**0.726**
MFor	0.779	0.788	0.779	0.515	**0.839**
RAuto	0.889	0.904	0.889	0.732	**0.939**
RMot	0.924	0.940	0.924	0.634	**0.962**
RSB	0.845	0.941	0.846	0.633	**0.957**
RSH	0.789	0.940	0.791	0.371	**0.936**
SCry	0.883	0.926	0.883	0.813	**0.928**
SEle	0.762	0.708	0.761	0.455	**0.828**
SMed	0.913	0.939	0.915	0.799	**0.954**
SSpa	0.910	0.927	0.910	0.771	**0.935**
SRChr	0.847	0.855	0.847	0.698	**0.876**
TPG	0.837	0.860	0.838	0.744	**0.869**
TPMid	0.905	0.888	0.905	0.839	**0.948**
TPMi	0.729	0.766	0.728	0.569	**0.813**
TrMi	0.620	0.625	0.619	0.450	**0.662**
macro-F$_1$	0.769	0.795	0.769	0.609	**0.825**

Table 14. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 10 over the NB classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
Ath	0.835	0.816	0.828	0.811	0.817
CGra	0.821	0.817	0.830	0.795	0.811
CMWM	0.819	0.831	0.817	0.719	0.822
CSIPH	0.835	0.840	0.833	0.684	0.839
CSMH	0.848	0.848	0.862	0.783	0.849
CWin	0.844	0.845	0.844	0.809	0.852
MFor	0.840	0.842	**0.850**	0.698	0.842
RAuto	0.836	0.836	0.851	0.795	0.848
RMot	0.885	0.871	0.884	0.781	0.880
RSB	0.888	0.889	0.903	0.809	0.897
RSH	0.903	0.892	0.907	0.852	0.905
SCry	0.852	0.851	0.847	0.881	0.844
SEle	0.820	0.827	0.827	0.712	0.831
SMed	0.846	0.836	0.864	0.866	0.859
SSpa	0.798	0.781	0.805	0.836	0.794
SRChr	0.847	0.841	0.837	0.860	0.831
TPG	0.793	0.781	0.797	0.798	0.791
TPMid	0.775	0.757	0.793	0.830	0.760
TPMi	0.702	0.690	0.723	0.684	0.700
TrMi	0.772	0.754	0.775	0.362	0.758
macro-F_1	0.828	0.822	0.834	0.768	0.827
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
Ath	0.761	0.807	0.825	0.820	**0.829**
CGra	0.745	**0.842**	0.814	0.815	0.811
CMWM	0.131	**0.851**	0.831	0.147	0.831
CSIPH	0.665	**0.864**	0.838	0.840	0.718
CSMH	0.778	**0.880**	0.859	0.861	0.829
CWin	0.662	**0.875**	0.853	0.863	0.726
MFor	0.779	0.844	0.846	0.833	0.839
RAuto	0.889	0.867	0.844	0.844	**0.939**
RMot	0.924	0.887	0.872	0.877	**0.962**
RSB	0.845	0.904	0.888	0.900	**0.957**
RSH	0.789	0.910	0.901	0.911	**0.936**
SCry	0.883	0.865	0.855	0.842	**0.928**
SEle	0.762	0.849	0.840	0.823	**0.828**
SMed	0.913	0.872	0.852	0.861	**0.954**
SSpa	0.910	0.833	0.809	0.794	**0.935**
SRChr	0.847	0.842	0.848	0.829	**0.876**
TPG	0.837	0.802	0.800	0.795	**0.869**
TPMid	0.905	0.792	0.784	0.761	**0.948**
TPMi	0.729	0.729	0.711	0.714	**0.813**
TrMi	0.620	0.754	0.762	0.745	**0.662**
macro-F_1	0.769	0.843	0.832	0.794	**0.859**

Table 15. Performance on F1 measure in the 20Newsgroups dataset with threshold $\rho = 10$ over the SVM classifier

Category	Term weighting scheme				
	TF.CC	TF.MI	TF.OR	TF.PB	TF.RF
Ath	0.785	0.712	0.736	0.837	0.428
CGra	0.736	0.713	0.771	0.573	0.631
CMWM	0.728	0.722	0.789	0.517	0.586
CSIPH	0.671	0.646	0.622	0.628	0.537
CSMH	0.762	0.754	0.751	0.789	0.584
CWin	0.717	0.722	0.790	0.609	0.628
MFor	0.777	0.742	0.754	0.581	0.631
RAuto	0.819	0.816	0.809	0.586	0.653
RMot	0.874	0.853	0.893	0.726	0.581
RSB	0.855	0.821	0.821	0.597	0.511
RSH	0.894	0.848	0.861	0.624	0.719
SCry	0.817	0.796	0.824	0.632	0.830
SEle	0.692	0.591	0.608	0.319	0.344
SMed	0.866	0.849	0.880	0.933	0.606
SSpa	0.860	0.840	0.827	0.927	0.683
SRChr	0.777	0.753	0.767	0.854	0.613
TPG	0.813	0.802	0.848	0.920	0.667
TPMid	0.864	0.803	0.857	0.966	0.823
TPMi	0.736	0.670	0.714	0.850	0.278
TrMi	0.468	0.268	0.408	0.645	0.213
macro-F_1	0.776	0.736	0.766	0.706	0.577
	TF.IDF	TF.ICF	TF. ICS$_\delta$F	TF.IDF.ICF	TF.IDF.ICS$_\delta$F
Ath	0.804	0.800	0.805	0.835	**0.911**
CGra	0.834	0.829	0.834	0.796	**0.900**
CMWM	0.809	0.803	0.802	0.780	**0.867**
CSIPH	0.756	0.775	0.761	0.768	**0.849**
CSMH	0.839	0.851	0.838	0.825	**0.905**
CWin	0.851	0.851	0.844	0.827	**0.907**
MFor	0.834	0.811	0.837	0.815	**0.905**
RAuto	0.893	0.900	0.896	0.875	**0.956**
RMot	0.948	0.946	0.942	0.925	**0.972**
RSB	0.946	0.945	0.944	0.945	**0.981**
RSH	0.923	0.922	0.921	0.958	**0.978**
SCry	0.883	0.896	0.877	0.924	**0.963**
SEle	0.813	0.798	0.791	0.795	**0.906**
SMed	0.934	0.943	0.932	0.908	**0.966**
SSpa	0.908	0.918	0.914	0.902	**0.958**
SRChr	0.832	0.835	0.829	0.866	**0.944**
TPG	0.865	0.877	0.864	0.870	**0.945**
TPMid	0.910	0.892	0.906	0.939	**0.978**
TPMi	0.818	0.824	0.821	0.832	**0.934**
TrMi	0.503	0.497	0.471	0.683	**0.832**
macro-F_1	0.845	0.846	0.841	0.853	**0.928**

In these tables, in comparing with other weighting methods, TF.IDF.ICF and TF.IDF.ICS$_{\delta}$F is outperformed over other methods in centroid and SVM classifier respectively. TF.ICF is showing its superiority in NB classifier. In contrast, Tables 13, 14, and 15 show F$_1$ measure F$_1$(C$_k$) of a certain category in the 20 Newsgroups dataset over centroid, NB, and SVM classifier respectively. The result shows that TF.IDF.ICS$_{\delta}$F shows its own superiority in among three classifiers. It achieves higher accuracy 18 of 20 categories in centroid, 14 of 20 in NB, and 20 of 20 categories in SVM over other term weighting approaches.

The above results show that the class-indexing-based TF.IDF.ICS$_{\delta}$F and TF.IDF.ICF approaches outperformed over the different term weighting approaches. This is especially the case for the TF.IDF.ICS$_{\delta}$F approach, which shows its superiority almost all the categories in the dataset.

3.5.4 Results with RCV1-v2 dataset

In this dataset, we do not introduce any thresholds because of unbalanced document distribution in the training process. Table 3 shows that majority of the categories have a small domain, some of them are only contains three to thirteen documents in a certain category. Therefore, the total number of unique terms in term space $T_i = \{t_{1,} t_{2, \ldots} t_n\}$ is 58,885, which includes many rare terms in the vector space. Figure 11 shows the performance based on micro-F1 over the centroid classifier. In this figure, the TF.IDF.ICF showing its own superiority over other weighing methods, which achieved the highest micro-F1 score (79.06). The TFIDF and TF.IDF.ICSdF are second and third rank with micro-F1 score is 78.87 and 78.63 respectively.

3.6 Overall Performances and Discussions

Figures 7, 8, 9, and 10 show the performance comparison with micro-F1on ten different term weighting methods over the Reuters-21578 and 20 Newsgroups datasets with setting threshold ρ = 3 and ρ = 10 using three different learning classifier. In figures 7, 8 for Reuters-21578 and figures 9, 10 for 20 newsgroups, the proposed TF.IDF.ICS$_{\delta}$F outperforms in SVM and centroid classifier to compare with other term weighting approaches.

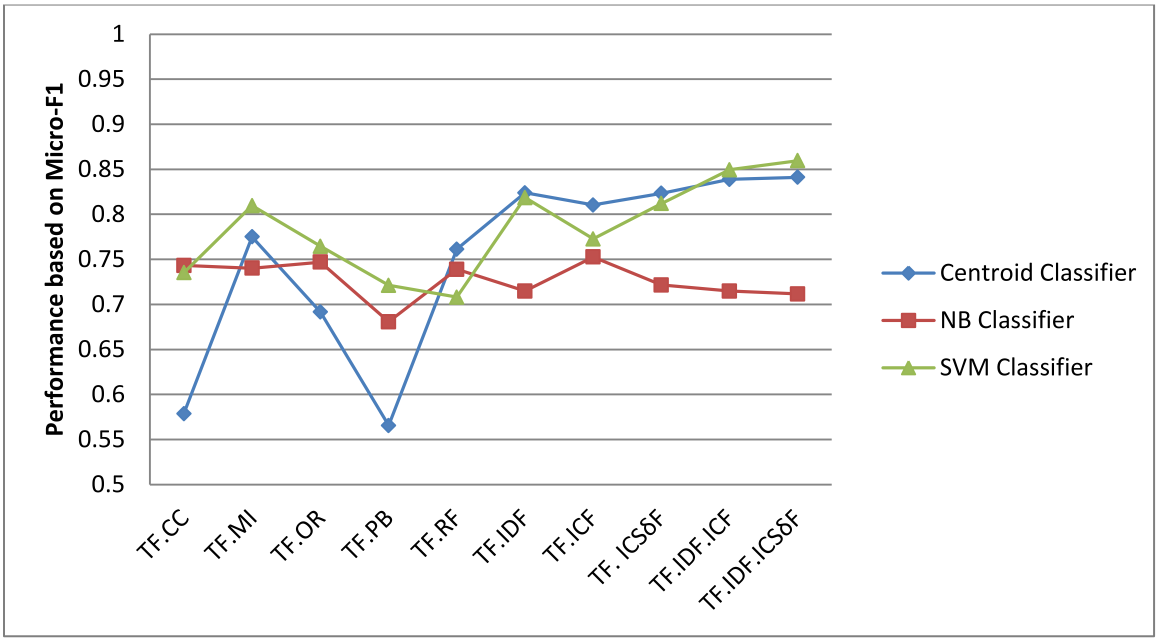


Figure 7. Performance comparison with micro-F_1 in the Reuters-21578 dataset with setting threshold using $\rho = 3$ over the Centroid, NB, and SVM classifier.

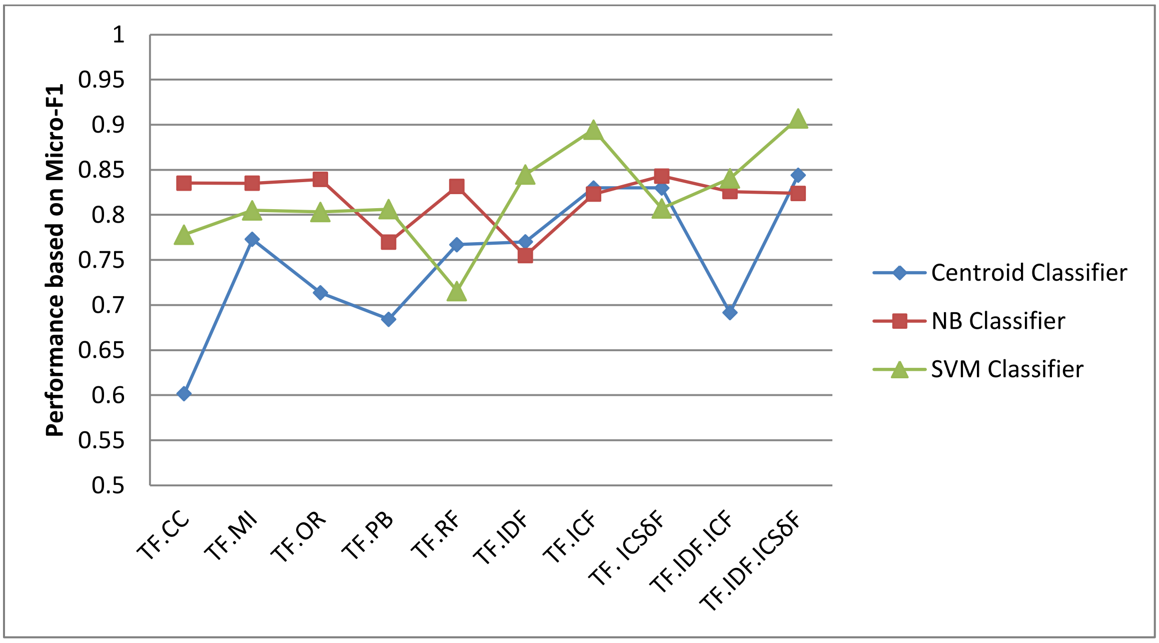


Figure 8. Performance comparison with micro-F_1 in the Reuters-21578 dataset with setting threshold using $\rho = 10$ over the Centroid, NB, and SVM classifier.

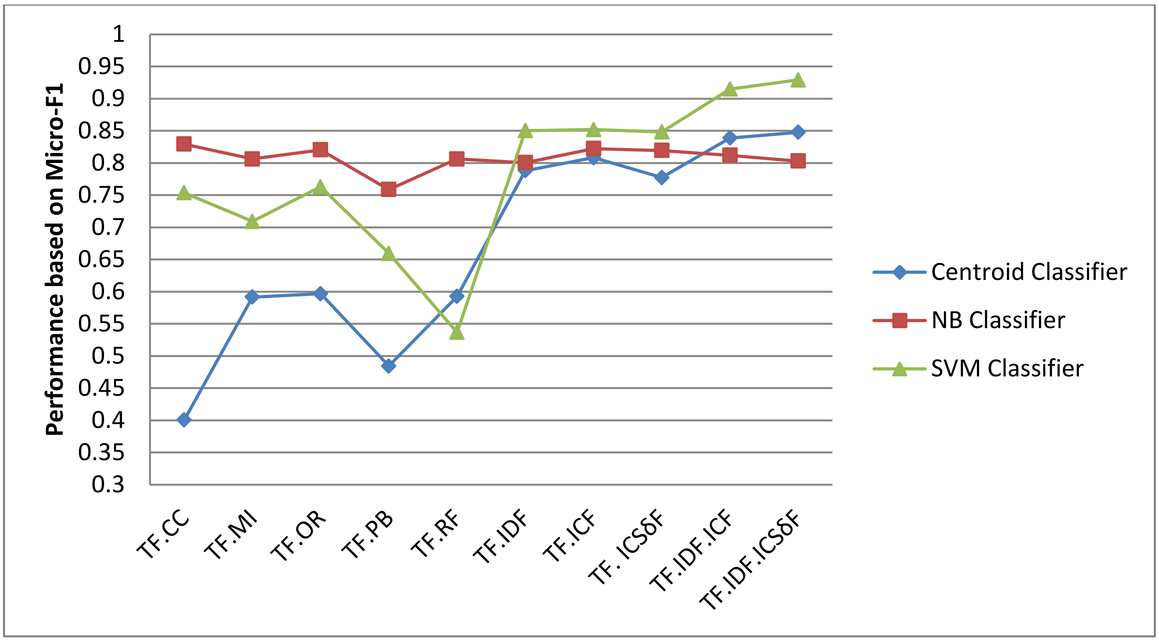


Figure 9. Performance comparison with micro-F_1 in the 20 Newsgroups dataset with setting threshold using ρ = 3 over the Centroid, NB, and SVM classifier.

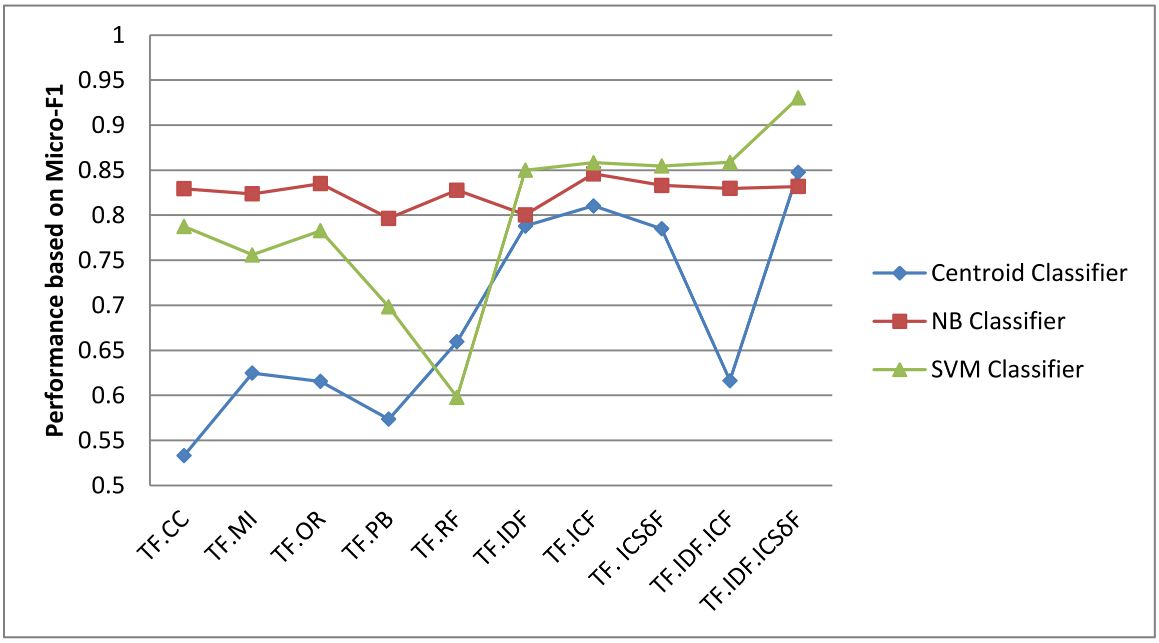


Figure 10. Performance comparison with micro-F_1 in the 20Newsgroups dataset with setting threshold using ρ = 10 over the Centroid, NB, and SVM classifier.

The results of the above experiments show that the proposed class-indexing-based TF.IDF.$ICS_{\delta}F$ term weighting method consistently outperforms in high-dimensional and in comparatively low-dimensional vector spaces over the other methods. Moreover, the TF.IDF.ICS$_\delta$F approach shows its superiority not only in overall, micro-F$_1$, and macro-F$_1$ but also in all the categories of the 20 Newsgroups and a majority of the Reuters-21578 datasets using SVM and centroid classifier. Another proposed class-indexing-based TF.IDF.ICF term weighting approach outperformed on RCV1-v2, which is very high-dimensional vector space with numerous rare terms are included in the VSM.

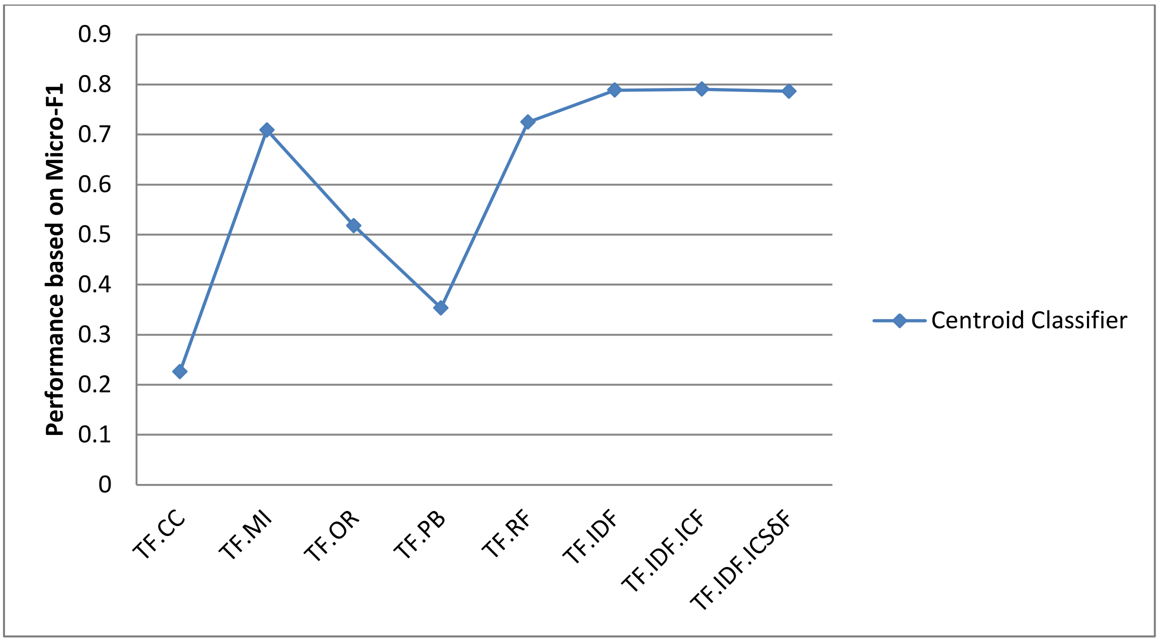


Figure 11. Performance comparison with micro-F$_1$ in the RCV1-v2 dataset over the Centroid classifier.

However, it is important to note that from the experiments result the combination of TF.IDF.ICS$_\delta$F and centroid or TF.IDF.ICS$_\delta$F and SVM, which can significantly improve the system performance. It is also noticeable that, the TF.IDF.ICS$_\delta$F approach is very effective when we reduce the vector space. Therefore, it is important to generate more informative terms in the class-indexing-based term weighting method in order to enhance the automatic indexing task. Our study shows that the TF.IDF.ICS$_\delta$F approach is a novel term weighting method that demonstrates a consistently higher performance over other term weighting approaches.

It is worth to mention that, in Reuter-21578, the results of some small categories like wheat and corn are comparatively poor along with other categories in this dataset. We, therefore, examined the Reuters-21578 dataset to explore the possible reason for providing poor accuracy. In high dimensional vector space, the category wheat and corn contains 1169(13.02%) and 1153(12.8%) terms respectively, to represent their own domain which we found very small in compare with other categories. And in comparatively low dimensional vector space, the category wheat and corn contains 515(6.2%) and 471(5.7%) terms respectively. These categories are semantically much correlated and most of the terms are overlapped with each other.

3.7 Summary

This chapter presented an overview of the core algorithm of class-indexing concepts of text classification that are used and built upon within this thesis. First, the prototype class-indexing and how to determine the category mapping of a certain term of a certain category were discussed. A system example was introduced to get a clear idea of proposed system. Several classifiers were introduced to build a machine learning based classification system. Next, a brief evaluation method was discussed using different dataset. A high-dimensional and comparatively low-dimensional vector space was considered judging their effect in the classification task.

Chapter 4

Combined Term Weighting Schemes

This chapter presents a brief introduction of building combination of different term weighting approaches which is used in this work to address a new weighting scheme in information retrieval system, especially on automatic text classification. In section 4.1 gives an overview of related work. In section 4.2, system architecture of combined term weighting schemes are discussed. Section 4.3 shows experiment results of combined term weighting approaches in compare with other methods. Finally, section 4.4 shows overall performance and discussions.

4.1 Related Work

Sohrab, Fattah, and Fuji [53] discussed about different text features, including sentence position, sentence centrality, sentence resemblance to the title, sentence inclusion of named entity, sentence inclusion of numerical data, and sentence relative length to enhance automatic text summarization. In this approach, first judge the effect of individual feature parameter score with different compression ratio on summarization performance. Therefore, the sum of all normalized feature parameter is constructed to address summarization task. The experiment's results showed that the sum of all normalized feature parameter approach outperforms the individual feature parameter. Fattah and Fuji [54] discussed about different models, including Genetic Algorithm (GA), Mathematical Regression (MR), Feed Forward Neural network (FFNN), Probabilistic Neural Network (PNN), and Gaussian Mixture Model (GMM) to combine with the sum of all normalized feature parameter. The experiment's results showed that the results of different models with the sum of all normalized feature parameter are promising to enhance automatic text summarization.

4.2 Combined-Term-Weighting-Scheme

The Combined-Term-Weighting-Scheme (CTWS) is another criterion of weighting a term, where combining all possible weighting approaches together and generate a new weighting scheme. In this approach, we take the summation of feature parameters associated with the document under consideration to calculate its score value from the following equation. Therefore, the CTWS score for a certain term in a certain document with respect to a certain category is given as

$$CTWS(t_i, d_j, c_k) = w_1.TF.CC + w_2.TF.MI + w_3.TF.OR + w_4.TF.PB + w_5.TF.RF + w_6.TF.IDF + w_7.TF.ICF + w_8.ICS_\delta F + w_9.TF.IDF.ICF + w_{10}.TF.IDF.ICS_\delta F, \quad (39)$$

In Equation, for a certain term t_i, a weighted *CTWS* score function is exploited to integrate the ten feature scores; where w_i indicates the weight of t_i. To calculate the weight (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}), we therefore introduce five different approaches, including *CTWS* with Summation (*CTWS-Sum*), CTWS with average (CTWS-Avg.), CTWS with Mathematical Regression Model (CTWS-MR), CTWS with Genetic Algorithm Model (CTWS-GA), and CTWS with Feed Forward Neural Network (CTWS-FFNN). It is worth to mention that, the global weight w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, and w_{10} are calculated from the vector space of TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.ICS$_\delta$F, TF.IDF.ICF, and TF.IDF.ICS$_\delta$F respectively. Finding the global weight of a certain vector space with respect to a certain weighting method is far more complicated task. In this *CTWS* scheme, we introduce summation-based, average-based, MR-based, GA-based, and FFNN-based approaches to generate global weight from a certain vector space using different term weighting approaches.

4.2.1 CTWS-Sum Approach

We assume that, the output of global weight values (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}) are between 0 to 1, where 1 is a ‘perfect score’ and 0 is perfectly lousy score. Therefore, the weight between 0 to 1 for a certain global weight w_i ($i = 1, 2, ...10$) is further incorporated with certain term weighting scheme respecting a certain dataset. As such, it must be bounded by,

$$0 \leq w_i \leq 1, \text{ where } i=1, 2, \ldots .10$$

In CTWS-Sum approach, we assume that, the weight values of (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}) = 1. We apply Eq. 31 after using the defined weights from w_i. Therefore, the CTWS-Sum score is given as

$$CTWS - Sum(t_i, d_j, c_k) = TF.CC + TF.MI + TF.OR + TF.PB + TF.RF + TF.IDF + TF.ICF + ICS_\delta F + TF.IDF.ICF + TF.IDF.ICS_\delta F, \quad (40)$$

4.2.2 CTWS-Avg. Approach

In this approach, we estimate the global text feature weight $W = \{w_1, w_2, \ldots w_{10}\}$ using CTWS-Avg. model. The VSM, each document d_j is considered to be a vector in the term space. To calculate the global weight of a certain dataset, first we compute the document weight from document vector d_j ($j = 1, 2 \ldots M$),

$$\text{Document Weight}, d_j = \frac{1}{N} \sum_{k=1}^{N} t_k, \quad (41)$$

Where, N is the number of terms t_k ($k=1, 2, 3, \ldots N$) in a document d_j. Next, calculate the global weight as follows:

$$\text{Global weight}, W = \frac{1}{M} \sum_{j=1}^{M} d_j, \quad (42)$$

Where, M is the total number of documents in a dataset. Therefore, we compute the ten different global weight using ten different term weighting approaches for a certain dataset. We apply Eq. 39 after using the defined weights from $W = \{w_1, w_2, \ldots w_{10}\}$. The numeric representation of combined term weighting scheme based on average weighting approach for a certain term, represented as

$$CTWS - Avg.(t_i, d_j, c_k) = w_1.TF.CC + w_2.TF.MI + w_3.TF.OR + w_4.TF.PB + w_5.TF.RF + w_6.TF.IDF + w_7.TF.ICF + w_8.ICS_\delta F + w_9.TF.IDF.ICF + w_{10}.TF.IDF.ICS_\delta F, \quad (43)$$

4.2.3 CTWS-MR Approach

In this approach, the global weight of TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.ICS$_\delta$F, TF.IDF.ICF, and TF.IDF.ICS$_\delta$F is calculated using Mathematical Regression (MR) model.

4.2.3.1 Mathematical Regression Model

The mathematical regression (MR) model is exploited to obtain an appropriate set of feature weights using Reuters-21578 and 20Newsgroups dataset. Mathematical regression is a good model to estimate the text feature weights. In this model a mathematical relates output to input. In matrix notation, we can represent regression as follows:

$$\begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ Y_m \end{bmatrix} = \begin{bmatrix} X_{01} & X_{02} & X_{03} & \cdots & \cdots & X_{010} \\ X_{11} & X_{12} & X_{13} & \vdots & \vdots & X_{110} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{m1} & X_{m2} & X_{m3} & \cdots & \cdots & X_{m10} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_{10} \end{bmatrix}, \qquad (44)$$

Where [Y] is the output vector, [X] is the input matrix (feature parameter), [W] is the linear statistical model of the system (the weights w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10} in Eq. 44). m is the total number of terms in the training corpus. We apply Eq. 39 after using the defined weights from MR execution.

4.2.4 CTWS-GA Approach

In this approach, the global weight of TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.ICS$_\delta$F, TF.IDF.ICF, and TF.IDF.ICS$_\delta$F is calculated using genetic algorithm (GA) model.

4.2.4.1 Genetic Algorithm Model

The genetic algorithm (GA) is exploited to obtain an appropriate set of feature weights using Reuters-21578 and 20Newsgroups dataset. A chromosome is represented as the combination of all feature weights in the form of (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}). 1500 genomes for each generation were produced. In this experiment, 150 generations are evaluated to obtain steady of feature weights. We apply Eq. 39 after using the defined weights from GA execution. Therefore, the numeric representation of combined term weighting scheme based on genetic algorithm for a certain term, represented as

$$CTWS - GA(t_i, d_j, c_k) = w_1.TF.CC + w_2.TF.MI + w_3.TF.OR + w_4.TF.PB + w_5.TF.RF + w_6.TF.IDF + w_7.TF.ICF + w_8.ICS_\delta F + w_9.TF.IDF.ICF + w_{10}.TF.IDF.ICS_\delta F, \qquad (45)$$

4.2.5 CTWS-FFNN Approach

In this approach, the global weight of TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.ICS$_\delta$F, TF.IDF.ICF, and TF.IDF.ICS$_\delta$F is calculated using feed forward neural network (FFNN).

4.2.5.1 Feed Forward Neural Network

The feed forward neural network (FFNN) is exploited to obtain an appropriate global weight using different weighting schemes based on Reuters-21578 and 20Newsgroups dataset. The layered structure of the neural network that we used is illustrated in Figure 12. We have used 10 input units for Reuters-21578 and 20Newsgroups; 10 hidden units and 1 output unit to represent the neural network. The input unit represents the weight of a certain term using a certain weighting scheme as described in chapter 2 and 3.

All the input features are represented by the feature vector $\vec{x}$. The output of the hidden layer is given by:

$$O_j^{(1)} = f\left(\sum_{k=1}^{N} W_{jk}^{(1)} X_k\right), \qquad (46)$$

Where W_{jk} is the weight associated with the line between the input unit k and the hidden unit j. f is a sigmoidal function given by

$$f(z) = \frac{1}{1+\exp(-z)}, \qquad (47)$$

The output of the output layer given by

$$O_i^{(2)} = f\left(\sum_{k=1}^{M} W_{ij}^{(2)} O_j^{(1)}\right), \tag{48}$$

Where W_{ij} is the weight associated with the line between the hidden unit j and the output unit i. From equation 46 and 48

$$O_i^{(2)} = f\left(\sum_{k=1}^{N} W_{ij}^{(2)} f\left(\sum_{k=1}^{N} W_{jk}^{(1)} X_k\right)\right), \tag{49}$$

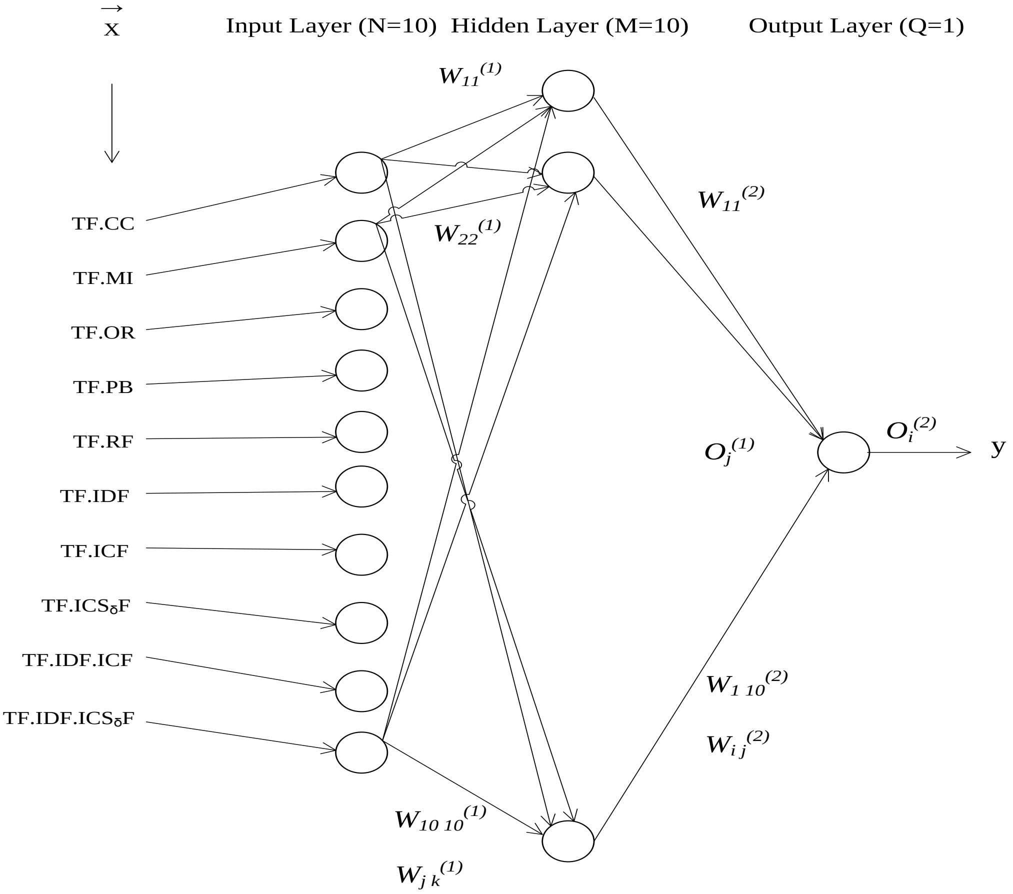


Figure 12. The feed forward neural network structure

The output $O_i^{(2)}$ further represents global weight $W = \{w_1, w_2, \ldots w_{10}\}$ using a certain weighting scheme. The global weight *w1, w2, w3, w4, w5, w6, w7, w8, w9,* and *w10* represents the term weighting scheme of TF.CC, TF.MI, TF.OR, TF.PB, TF.RF, TF.IDF, TF.ICF, TF.ICS$_\delta$F, TF.IDF.ICF, and TF.IDF.ICS$_\delta$F respectively.

Therefore, the numeric representation of combined term weighting scheme based on feed forward neural network for a certain term, represented as

$$CTWS - FFNN(t_i, d_j, c_k) = w_1.TF.CC + w_2.TF.MI + w_3.TF.OR + w_4.TF.PB + w_5.TF.RF + w_6.TF.IDF + w_7.TF.ICF + w_8.ICS_\delta F + w_9.TF.IDF.ICF + w_{10}.TF.IDF.ICS_\delta F, \quad (50)$$

4.3 Experiments and Evaluations

In this section, we provide empirical evidence for the effectiveness of the proposed CTWS over different term weighting approaches. In this experiments, we use Reuters-21578 and 20Newsgroups dataset with 10-fold cross validation method.

4.3.1 The Results with Reuters-21578 dataset

In the Reuters-21578 dataset, Tables 16, 17, and 18 show F_1 measure $F_1(C_k)$ of a certain category over NB, centroid, and SVM classifier respectively. In table 16, CTWS-FFNN weighting method showing its superiority in compare with others. It achieves higher accuracy 7 of 10 in NB classifier. In table 17 and 18, CTWS-Sum weighting approach achieves higher accuracy 10 of 10 in centroid and SVM classifier in compare with other term weighting method.

Table 16. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 10 over the NB classifier

Category	Term weighting scheme				
	CTWS-Sum	CTWS-Avg.	CTWS-GA	CTWS-FFNN	CTWS-MR
acq	0.872419	0.871136	0.871344	**0.877273**	0.876241
corn	**0.576577**	0.472222	0.403061	0.533333	0.378995
crude	0.749392	0.747164	0.745036	**0.817518**	0.773055
earn	0.937171	0.934515	0.93835	**0.942928**	0.937676
grain	0.673863	0.682737	0.661786	**0.696971**	0.667692
interest	**0.680272**	0.675172	0.643206	0.603175	0.677702
money-fx	0.755556	0.750892	0.785311	**0.792208**	0.759207
ship	**0.735751**	0.706282	0.735675	0.685714	0.736475
trade	0.707843	0.712062	0.733333	**0.788991**	0.709291
wheat	0.576792	0.585198	0.529946	**0.615385**	0.548223
macro-F_1	0.726563	0.713738	0.704705	**0.735349**	0.706456

Table 17. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 10 over the Centroid classifier

Category	Term weighting scheme				
	CTWS-Sum	CTWS-Avg.	CTWS-GA	CTWS-FFNN	CTWS-MR
acq	**0.902981**	0.844764	0.798652	0.841671	0.818542
corn	**0.462428**	0.159875	0.104651	0.146707	0.138138
crude	**0.836879**	0.682645	0.689723	0.661303	0.742382
earn	**0.933048**	0.898932	0.857966	0.897891	0.866657
grain	**0.406781**	0.134986	0.162857	0.119534	0.154286
interest	**0.790146**	0.579618	0.534261	0.563886	0.591881
money-fx	**0.836808**	0.655841	0.647059	0.635308	0.658892
ship	**0.760871**	0.451481	0.450539	0.436364	0.451081
trade	**0.824074**	0.612691	0.557924	0.586283	0.619048
wheat	**0.536451**	0.463329	0.379267	0.462094	0.459771
macro-F_1	**0.729046**	0.548416	0.518291	0.535104	0.550068

Table 18. Performance on F1 measure in the Reuters-21578 dataset with threshold ρ = 10 over the SVM classifier

Category	Term weighting scheme				
	CTWS-Sum	CTWS-Avg.	CTWS-GA	CTWS-FFNN	CTWS-MR
acq	**0.960942**	0.956665	0.843126	0.907267	0.904041
corn	**0.458427**	0.406948	0.181319	0.215613	0.248555
crude	**0.888293**	0.883094	0.792549	0.858121	0.857633
earn	**0.983363**	0.981104	0.925142	0.930238	0.951213
grain	**0.649957**	0.633117	0.415411	0.495379	0.512963
interest	**0.905782**	**0.905782**	0.552707	0.674541	0.706186
money-fx	**0.922969**	0.920056	0.758621	0.848981	0.846205
ship	**0.802013**	0.789116	0.613687	0.599078	0.634146
trade	**0.910345**	0.898722	0.674298	0.737643	0.801951
wheat	**0.490722**	0.383372	0.072848	0.275689	0.222222
macro-F_1	**0.797281**	0.775797	0.582969	0.654255	0.668511

4.3.2 The Results with 20Newsgroups dataset

In the 20Newsgroup dataset, Tables 19, 20, and 21 show F_1 measure $F_1(C_k)$ of a certain category over centroid, NB, and SVM classifier respectively. In table 19, CTWS-Sum weighting method showing its superiority in compare with others. It achieves higher accuracy 16 of 20 in centroid classifier. In table 20, CTWS-Sum weighting approach achieves higher accuracy 20 of 20 in NB classifier in compare with other term weighting method. In table 21, CTWS-Sum weighting approach achieves higher accuracy 20 of 20 in SVM classifier in compare with other term weighting methods.

Table 19. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 10 over the Centroid classifier

Category	Term weighting scheme				
	CTWS-Sum	CTWS-Avg.	CTWS-GA	CTWS-FFNN	CTWS-MR
Ath	**0.78273**	0.72235	0.68207	0.69295	0.70746
CGra	**0.74034**	0.70453	0.57389	0.61227	0.63330
CMWM	**0.68198**	0.12110	0.10195	0.11361	0.11101
CSIPH	**0.66993**	0.62448	0.53591	0.56118	0.58897
CSMH	**0.79977**	0.75062	0.65588	0.70348	0.72102
CWin	**0.74681**	0.63486	0.57432	0.59334	0.61002
MFor	0.57023	**0.66630**	0.59044	0.58468	0.61613
RAuto	0.84575	**0.85540**	0.79795	0.81489	0.83649
RMot	**0.91383**	0.84107	0.71023	0.76078	0.79161
RSB	**0.89990**	0.81661	0.67170	0.72719	0.76169
RSH	**0.92683**	0.76776	0.47801	0.61164	0.69324
SCry	**0.88367**	0.87336	0.82853	0.84080	0.85168
SEle	**0.63730**	0.62715	0.48125	0.50731	0.56624
SMed	0.89178	**0.89568**	0.82583	0.84711	0.86495
SSpa	**0.90302**	0.88181	0.81901	0.83557	0.85163
SRChr	0.80629	**0.81199**	0.73892	0.75603	0.78138
TPG	**0.85390**	0.82279	0.76605	0.78309	0.80193
TPMid	**0.89486**	0.87641	0.82309	0.83926	0.86296
TPMi	**0.70975**	0.68942	0.59334	0.61701	0.64823
TrMi	**0.61309**	0.56738	0.49429	0.52181	0.53866
macro-F_1	**0.78859**	0.72755	0.63713	0.66620	0.69193

Table 20. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 10 over the NB classifier

Category	Term weighting scheme				
	CTWS-Sum	CTWS-Avg.	CTWS-GA	CTWS-FFNN	CTWS-MR
Ath	**0.986850**	0.821429	0.880934	0.810644	0.885918
CGra	**0.927207**	0.829907	0.878233	0.821000	0.881455
CMWM	**0.936631**	0.823293	0.869165	0.826411	0.890370
CSIPH	**0.933812**	0.824918	0.875000	0.831332	0.884503
CSMH	**0.960207**	0.853947	0.902678	0.862539	0.915680
CWin	**0.946356**	0.853977	0.891704	0.851019	0.907096
MFor	**0.942517**	0.842388	0.873563	0.835429	0.894297
RAuto	**0.979282**	0.851687	0.885057	0.852041	0.888071
RMot	**0.987915**	0.876066	0.918605	0.877358	0.926503
RSB	**0.989950**	0.901682	0.932312	0.888071	0.939283
RSH	**0.990476**	0.908858	0.928525	0.901031	0.935123
SCry	**0.987418**	0.873379	0.905028	0.852621	0.900501
SEle	**0.959225**	0.825890	0.874515	0.835897	0.884615
SMed	**0.987830**	0.860858	0.895847	0.859206	0.912281
SSpa	**0.978184**	0.817961	0.858716	0.808594	0.892608
SRChr	**0.980626**	0.832714	0.881734	0.827046	0.899857
TPG	**0.987939**	0.805930	0.856566	0.802174	0.872783
TPMid	**0.981550**	0.781371	0.860511	0.779419	0.876571
TPMi	**0.983269**	0.737809	0.786136	0.730248	0.803056
TrMi	**0.969841**	0.752345	0.826462	0.749429	0.851397
macro-F_1	**0.969854**	0.833820	0.879065	0.830075	0.892098

Table 21. Performance on F1 measure in the 20Newsgroups dataset with threshold ρ = 10 over the SVM classifier

Category	Term weighting scheme				
	CTWS-Sum	CTWS-Avg.	CTWS-GA	CTWS-FFNN	CTWS-MR
Ath	**0.927665**	0.9120810	0.7003610	0.7555560	0.7989490
CGra	**0.875764**	0.8642860	0.6688590	0.7445110	0.7848780
CMWM	**0.868094**	0.8517790	0.6332360	0.7337530	0.7689340
CSIPH	**0.849186**	0.8433350	0.5868470	0.6773690	0.7201690
CSMH	**0.899795**	0.8873020	0.7193280	0.7768330	0.8081840
CWin	**0.883131**	0.8722610	0.7121210	0.7869360	0.8118040
MFor	**0.892171**	0.8833500	0.6813560	0.7067850	0.7588500
RAuto	**0.946626**	0.9388160	0.7765500	0.8474250	0.8660890
RMot	**0.970247**	0.9643390	0.7468990	0.9006080	0.9187840
RSB	**0.977444**	0.9758550	0.7705290	0.8566880	0.8960500
RSH	**0.976072**	0.9756100	0.8165880	0.8664900	0.8987630
SCry	**0.966145**	0.9622930	0.8017350	0.8493030	0.8783410
SEle	**0.905757**	0.8936600	0.5400890	0.6670680	0.7425290
SMed	**0.954637**	0.9473680	0.8243310	0.8705210	0.9100740
SSpa	**0.956655**	0.9486410	0.7958800	0.8558140	0.8894210
SRChr	**0.936337**	0.9286060	0.7590050	0.7824580	0.8156280
TPG	**0.944899**	0.9381110	0.7872450	0.8242610	0.8500770
TPMid	**0.975532**	0.9707600	0.8045980	0.8497850	0.8975750
TPMi	**0.91478**	0.9144380	0.6469280	0.7407960	0.8065900
TrMi	**0.879538**	0.8445950	0.3147460	0.4167640	0.4850570
macro-F_1	**0.925024**	0.9158740	0.7043620	0.7754860	0.8153370

4.4 Overall Performances and Discussions

Figures 13 and 14 show the performance comparison with micro-F1on 15 different term weighting methods over the Reuters-21578 and 20 Newsgroups datasets with setting threshold ρ = 10 using three different learning classifier. In figure 13, for Reuters-21578 the CTWS-Sum showing promising results in compare with other weighting scheme. In figure 14, for 20 newsgroups, the CTWS-Sum method showing its superiority in NB classifier.

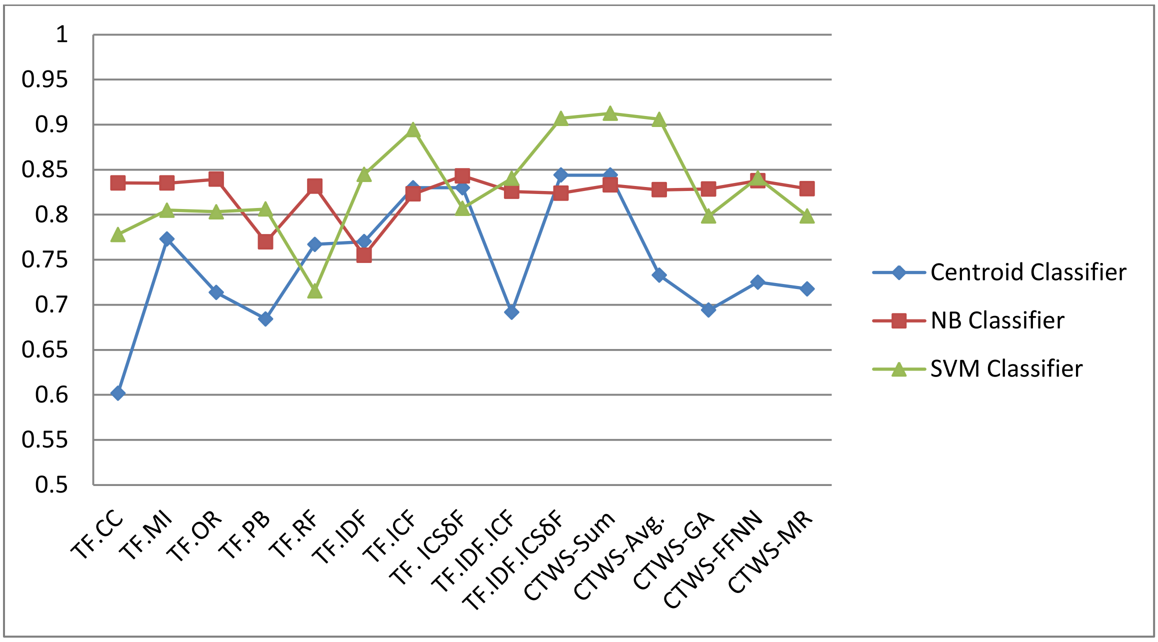


Figure 13. Performance comparison with micro-F_1 in the Reuters-21578 dataset with setting threshold using ρ = 10 over the Centroid, NB, and SVM classifier.

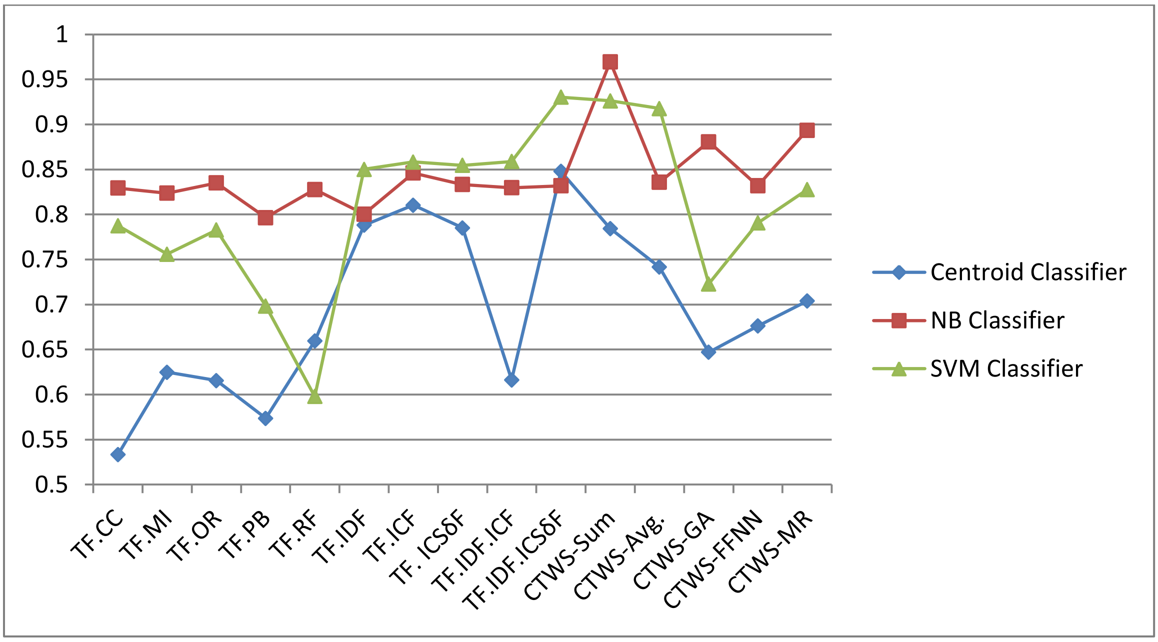


Figure 14. Performance comparison with micro-F_1 in the 20Newsgroups dataset with setting threshold using ρ = 10 over the Centroid, NB, and SVM classifier.

4.5 Summary

This chapter presented an overview of the combined term weighting scheme which incorporated with 10 different weighting approaches of automatic text classification. First, we discussed the combination of different models with CTWS. Next, experiments and evaluations were discussed.

Chapter 5

Class-Semantic-Indexing

This chapter presents a class-semantic-indexing which integrating category indexing and semantic indexing. Section 5.1 gives an overview of related work on semantic indexing. Section 5.2 presents class-semantic-indexing based term weighting method and in section 5.3 shows the proposed system prototype to navigate a certain term through corpus and dictionary based approach.

5.1 Related Work

In the indexing process [29, 38], semantic term weighting [5, 13, 25, 39] is another criterion of weighting a term, where term weighting is related to a term’s meaning and to the discriminative supremacy of a term that appears in a document or a group of documents, respectively. Leo, Chen, and Xiong [25] proposed a semantic term weighting by exploiting the semantics of categories and indexing term using WordNet. In this approach, they replaced the IDF function with a semantic weight (SW) and the experiment’s results, which were based on overall system performance, showed that the proposed TF.SW scheme that outperformed TF.IDF. However, the amount of training data was small and they were not able to show the improvement in the performance of a certain category in the dataset. Nevertheless, it is possible to address a limited number of terms in a term index by semantic term weighting, but with large number of terms in the term index, it is difficult to provide the appropriate semantic knowledge of a term based on categories. Recently, Youngjoong and Jungyun [15] proposed a new classification algorithm with feature projection techniques based on unsupervised learning. The TF.IDF is used as a feature weighting criterion. The classification approach is based on voting score incorporated with three different modules: voting ratio, modified weight, and adopted χ^2 feature selection method. In this approach, they project the elements using binary approach; if the category for an element is equal to a certain category, the output value is 1 otherwise 0. In the second part, where category frequency is the number of categories between any of the two features which is co-occurred in a certain category is computed. Therefore, they adopted χ^2 feature selection method to get the final voting score. The experiment results show that this approach is promising when the numbers of labeled training documents are up to 3000 approximately. The system failed to improve the performances in comparatively large number of labeled training documents in three different datasets.

5.2 Class-Semantic-Indexing based term weighting

Measuring the semantic indexing of concepts is an intriguing problem in Natural Language Processing. Various approaches that attempt to approximate human judgment of semantic indexing, have been tried by researchers. Semantic analysis method utilizes lexical hierarchies in the language dictionary or co-occurrence patterns of words in a large corpus of texts to decide the semantic similarity or semantic association between unknown words. In this section we look at the combination of Corpus-based and WordNet-based measures of class-semantic-indexing that we propose to compute weight of a certain term.

5.2.1 Category Mapping based on WordNet

In the Computational process class-space-density frequency for a certain term t_i which is discussed on section 3. In the next process, the sense or concept of that term t_i is taken into account based on WordNet. The creators of WordNet refer to it as an electronic lexical database [8]. Every node consists of a set of words, each representing the real world concept associated with that node. For example, the concept of a car may be represented by the set of words {car, auto, automobile, motorcar}. Such a set, in WordNet terminology, is known as a synset. A synset also has associated with it a short definition or description of the real world concept known as a gloss. In the category mapping process, class-space-density is computed from a certain dataset for a certain term t_i in respect to a certain gloss to judge how relevant that gloss term is in the dataset. We therefore arrange the glosses in a descending order and take the highest top gloss density weight and merge the corpus based class-space-density approach of certain term to address automatic text classification.

In a simple way, first we compute the weight of a certain term from a corpus, then compute the semantic weight based on glosses and merge together for that term.

5.3 Prototype Class-Semantic-Indexing System

A simple overview of the proposed class-semantic-indexing is shown in Figure 15 where the proposed class-semantic-indexing is incorporated with term, document and class index with corpus-based and Wordnet-dictionary-based combinational approaches. The nodes of term, document and class index contain two fields: data and link. While provide the dataset as an input to assign scores for lexical terms, the data field of term index contains a term with two different weights.

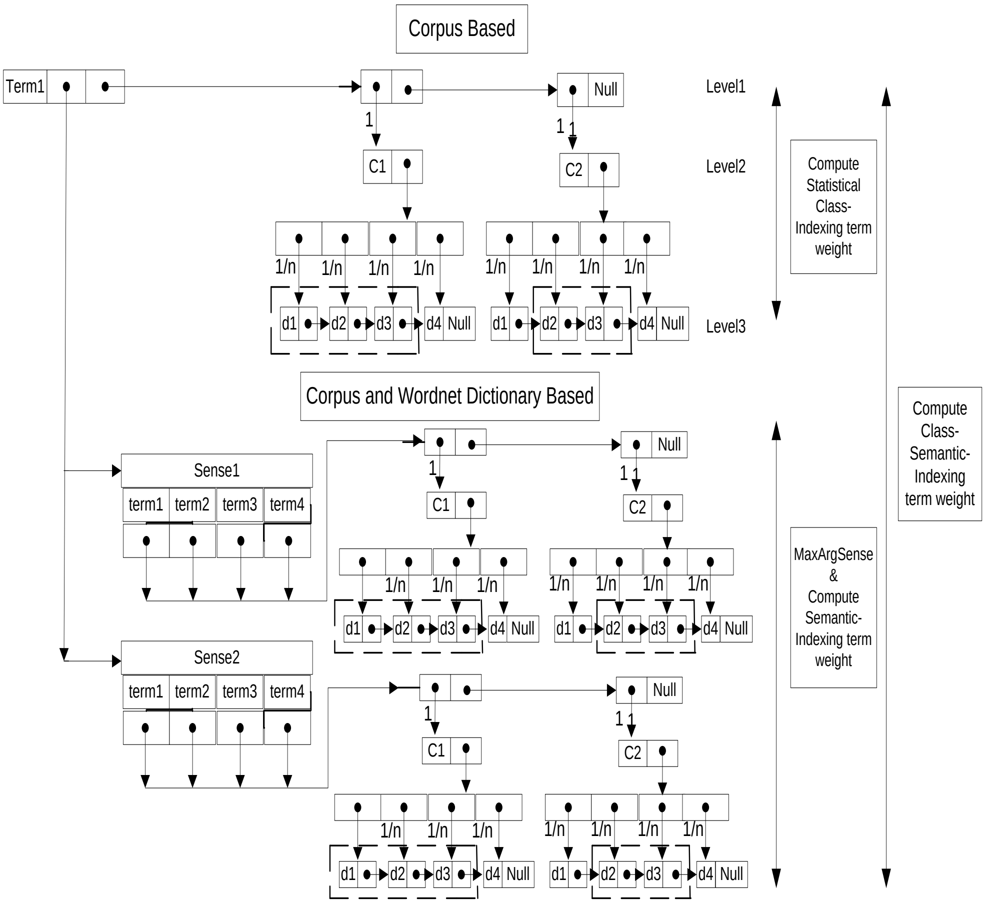

Corpus Based
Term1
Null
Level1
1
1 1
C1
C2
Level2
Compute Statistical Class-Indexing term weight
1/n
d1
d2
d3
d4
Null
Level3
Corpus and Wordnet Dictionary Based
Compute Class-Semantic-Indexing term weight
Sense1
term1
term2
term3
term4
MaxArgSense & Compute Semantic-Indexing term weight
Sense2

Figure 15. Architecture of Class-Semantic-Indexing

One is statistical corpus based term weight and another is Wordnet dictionary based term weight. To compute a certain term from a certain corpus from Level1 to Level 3 that is discussed in section 3. In the Wordnet dictionary based approach, first we determine the top rank sense from a set of senses for a certain term. We therefore, compute the weight for that sense through document and class index; and combine the weight with class-indexing based term weight and get a new class-semantic-weight for classification task.

5.4 Summary

This chapter presented an overview of new class-semantic-indexing based indexing system that are used and built upon within this thesis. First related studies were discussed to build a new indexing system. Then, the prototype class-semantic-indexing and how to determine the category mapping of a certain term of a certain category from corpus and Wordnet were discussed.

Chapter 6

Conclusion and Future work

This last chapter contains two sections. Section 6.1 presents conclusion the classification systems. Finally, in section 6.2 will give idea and plans for future work.

6.1Conclusion

In this study, we investigated the effectiveness of proposed class-indexing-based TF.IDF.ICS$_{\delta}$F and TF.IDF.ICF approaches with other different term weighing approaches using a centroid, Naïve Bayes, and SVM classifier to address the ATC task. The proposed term weighting approaches are effective in enhancing the classification task. The experiments were conducted using the Reuters-21578, 20 Newsgroups and RCV1-v2 datasets as benchmark collections. First, we proposed a new class-based indexing method to replace the traditional document indexing task. Then, we implemented a class-indexing-based TF.IDF.ICF term weighting approach that led to a new class-indexing-based TF.IDF.ICS$_{\delta}$F term weighing approach that emphasizes on addressing class space density rather than class space. The proposed class-indexing-based term weighting approaches outperformed the six different term weighting approaches. In particular, the TF.IDF.ICS$_{\delta}$F approach consistently outperformed in SVM and centroid classifier over other term weighting approaches. Since SVM is considered one of the most robust and achieved great success in TC, therefore, the combination of TF.IDF.ICS$_{\delta}$F and SVM can significantly improve the performance of TC and in other applications.

From the experiment results, it is clear that the TF.IDF.ICS$_{\delta}$F approach is very promising in domain based applications like text classification, information retrieval, information extraction, emotion recognition, topic identification, and many other applications in machine learning workbench. The TF.IDF.ICS$_{\delta}$F approach may further use to compute the weight of basic eight emotions[8] classification for intelligent robot.

[8] Joy, Love, Expectation, Surprise, Anxiety, Sorrow, Anger, and Hate

From the experiment results, it is noticeable that the TF.IDF.ICSdF approach is very effective when we reduce the vector space. Therefore, the proposed TF.IDF.ICS$_{\delta}$F approach can overcome the problem of high dimensionality. It gives positive discrimination both in rare or in frequent terms.

From the experiment results, it is also noticeable that the CTWS especially CTWS-Sum approach is very effective to providing a fair evaluation on three different classifiers that has been build up in this experiment. The TF.IDF.ICS$_{\delta}$F approach is very promising in centroid and SVM classifier but provide flexible performance on different dataset using NB classifier. The CTWS-Sum approach is very promising not only in NB classifier but also showing its consistency in centroid and SVM classifier. Therefore the CTWS is another criterion to enhance automatic text classification task. Although, the proposed TF.IDF.ICF, TF.IDF.ICS$_{\delta}$F and CTWS consistently performed well in single-label classification task, however, in its current form needs to apply on multi-label classification task; to judge whether the proposed method can significantly perform well or not to handle multi-label classification task.

6.2 Future Work

Future extension of this proposed methodology aim to improve the performance of text classification to introduce hybrid-indexing by exploiting semantic indexing technique. The classification performance can be further improved by combining class-indexing and semantic indexing. In future work, these researches can be extended to include machine learning based multi-label automatic text classification, emotion recognition of intelligent emotional robot from different emotion categories, multi-document summarized based automatic text classification, and Japanese-English-Bengali machine transliteration.

6.2.1 Machine Learning based multi-label ATC

In future work, beside single-label classification, we will conduct our experiments using class-semantic-indexing with multi-label classification. In the vector space model, a certain term in a certain category has different kind of meanings where a statistical based indexing approach cannot distinguish properly. Therefore, a semantic indexing based on Wordnet (an electronic lexical database, where a certain term consist of several concepts) is needed with statistical based approach to enhance the performance of multi-label text classification. The current research on text classification based on single-label classification which deals with the learning of instances that are associated a single label from a set of distinct labels. This research can be further applied with multi-label classification while instances/documents in multi-label classification are associated with a set of labels.

6.2.2 Emotion Recognition of Emotional Robot

Nowadays robotics is considering one of the key research areas in machine learning based approach. Especially the most intelligent emotional robot[9] which has eight different kinds of emotions: Joy, Love, Expectation, Surprise, Anxiety, Sorrow, Anger, and Hate. In this approach, we will compute these eight emotions with class-semantic-indexing to generate/perform multi-expressions like Love-Surprise, Surprise-Anxiety, and Anxiety-Sorrow etc. with a single command. Presently, the emotional Robot has three language resources: Japanese, English, and Chinese; which may extend Bengali textual emotion recognition in weblogs. The proposed TF.IDF.ICS$_{\delta}$F approach based on class-indexing based automatic text classification is very promising in domain based applications like text classification, information retrieval, information extraction, emotion recognition, topic detection and many other applications in machine learning workbench. The TF.IDF.ICS$_{\delta}$F approach may further use to compute the weight of basic eight emotions classification for intelligent emotional robot.

6.2.3 Japanese-English-Bengali Machine Transliteration

Parallel corpora can be thought of as a critical resource. Unfortunately, they are not readily available in the necessary quantities (especially Japanese-Bengali). It is a must to create parallel corpora for different language pairs since parallel corpora are very important tools to construct a good machine translation system and to make any natural language processing research for cross language information retrieval.

[9] http://a1-www.is.tokushima-u.ac.jp/member/ren/Projects/ren-robot/index-avatar.html

6.2.4 Multi document summarized based ATC

A major characteristics or difficulty of Text classification is the high dimensionality of the feature space. This vector space model includes with sparse matrix (a large number of elements with zero values) which may decrease the system performance. Nowadays in the field of information retrieval it is a primary demand to decrease the vector space and get more effective results. Therefore, in multi-document summarized based approach, first we compress the documents and retrieve useful information based on machine learning based automatic text summarization. We therefore apply the term weighting based on useful retrieval information and then classify the documents using classification system.

Bibliography

[1] J. Chen, H. Huang, S. Tian, Y. Qua, Feature selection for text classification with Naïve Bayes, Expert Systems with Applications 36 (2009) 5432–5435.

[2] T. F. Covoes, E. R. Hruschka, Towards improving cluster-based feature selectionwith a simplified silhouette filter, Information Sciences, 181 (2011) 3766-3782.

[3] F. Debole, F. Sebastiani, Supervised term weighting for automated text categorization, In Proceedings of the 2003 ACM symposium on applied computing, Melbourne, Florida, USA, 2003, pp. 784–788.

[4] S. Flora, T. Agus, Experiments in term weighting for novelty mining, Expert Systems with Applications, 38 (2011) 14094–14101.

[5] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Goncalves, Word co-occurrence features for text classification, Information Systems, 36 (2011) 843-858.

[6] N. Fuhr, C. Buckley, A probabilistic learning approach for document indexing, ACM Transactions on Information Systems, 9 (1991) 223–248.

[7] S. Godbole, S. Sarawagi, S. Chakrabarti, Scaling multi-class support vector machine using inter-class confusion, In Proceedings of the 8th ACM international conference on knowledge discovery and data mining, ACM Press, New Orleans, 2002, pp. 513–518.

[8] Y. Guo, Z. Shao, N. Hua, Automatic text categorization based on content analysis with cognitive situation models, Information Sciences, 180 (2010) 613-630.

[9] E. H. Han, G. Karypis, Centroid-based document classification: analysis and experimental results, In Principles of Data Mining and Knowledge Discovery, (2000) 424–431.

[10] T. Joachims, Text categorization with support vector machines: learning with many relevant features, In Proceedings of 10th European conference on machine learning, Springer Verlag, Heidelberg, Germany, 1998, pp. 137–142.

[11] T. Joachims, A statistical learning model of text classification for support vector machines, In Proceedings of the 24th ACM international conference on research and development in information retrieval, 2001.

[12] M. G. H. Jose, Text representation for automatic text categorization, 2003. Available at: http://www.esi.uem.es/~jmgomez/tutorials/eacl03/slides.pdf.

[13] B. Kang, S. Lee, Document indexing: A concept-based approach to term weight estimation, Information Processing and Management, 41(5) (2005) 1065–1080.

[14] S. Kansheng, H. Jie. L. Hai-tao, Z. Nai-tong, S. Wen-tao, Efficient text classification method based on improved term reduction and term weighting, The Journal of China Universities of Posts and Telecommunications, 18 (2011) 131-135.

[15] Y. Ko, J. Seo, Text classification from unlabeled documents with bootstrapping and feature projection techniques, Information Processing and management, 45 (2009) 70-83.

[16] M. Lan, C.L. Su, Y. Lu, Supervised and traditional term weighting methods for automatic text categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31 (2009) 721-735.

[17] J. H. Lee, Combining multiple evidence from different properties of weighting schemes, In Proceedings of the 18th International Conference on Research and Development in Information Retrieval, 1995, pp. 180–188.

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

[19] D. D. Lewis, Text representation for intelligent text retrieval: A classification-oriented view, In S. J. Paul (Eds.), Text-based intelligent systems: current research and practice in information extraction and retrieval, Lawrence Erlbaum Associates, Inc., Publishers, Hillsdale, New Jersey, USA, 1992a, pp. 179–197.

[20] D. D. Lewis, M. Ringuette, A Comparison of two learning algorithms for text categorization, In Proceedings of the third annual symposium on document analysis and information retrieval. Las Vegas, NV, USA, 1994, pp. 81–93.

[21] D. D. Lewis, R. E. Schapire, J. P. Callan, R. Papka, Training algorithms for linear text classifiers, In Proceedings of 19th ACM International Conference on Research and Development in Information Retrieval, 1996, pp. 298–30.

[22] D. D. Lewis, Y. Yang, T. Rose, and F. Li, RCV1: A new Benchmark Collection for text categorization Research, Journal of Machine Learning research, 5 (2004) 361-397.

[23] W. Li, D. Miao, W. Wang, Two-level hierarchical combination method for text classification, Expert systems with applications, 38 (2011) 2030-2039.

[24] Y. Liu, H. Loh, A. Sun, Imbalanced text classification: A term weighting approach, Expert Systems with Applications, 36 (2009) 690–701.

[25] Q. Luo, E. Chen, H. Xiong, A semantic term weighting scheme for text classification, Expert Systems with Applications, 38 (2011) 12708–12716.

[26] S. Maldonado, R. e Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Information Sciences, 181 (2011) 115-128.

[27] H. Ogura, H. Amano, M. Kondo, Comparison of metrics for feature selection in imbalanced text classification, Expert Systems with Applications, 38 (2011) 4978–4989.

[28] F. Peng, A. McCallum, Information extraction from research papers using conditional random fields, Information Processing and Management, 42 (2006) 963–979.

[29] G. Salton, A theory of indexing, Bristol, UK, 1975.

[30] G. Salton, A. Wong, C.S. Yang, A Vector Space Model for Automatic Indexing, Association of Computing Machines, 18 (1975) 613–620.

[31] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing and Management, 24 (1988) 513–523.

[32] G. Salton, M. J. McGill, Introduction to modern information retrieval, New York, 1983.

[33] F. Sebastiani, Machine learning in automated text categorization, ACM computing Surveys, 34 (2002) 1-47.

[34] G. Salton, C. S. Yang, C. T. Yu, Contribution to the theory of indexing, Proceedings of International Federation for information Processing Congress 74, Stockholm, American Elsevier, New York, 1973.

[35] H. Schutze, D. Hull, J. O. Pedersen, A comparison of classifiers and document representations for the routing problem. In Proceedings of the 18th ACM International Conference on Research and Development in Information Retrieval, 1995, pp. 229–237.

[36] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Systems with Applications, 33 (2007) 1–5.

[37] K. Sparck Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation, 28 (1972) 11–21.

[38] K. Sparck Jones, Index term weighting, Information Storage and Retrieval, 9 (1973) 619–633.

[39] A. Tagarelli, Exploring dictionary-based semantic relatedness in labeled tree data, Information Sciences, 220 (2012) 244-268.

[40] S. Tan, An improved centroid classifier for text categorization, Expert Systems with Applications, 35 (2008) 279–285.

[41] T. Theeramnukkong, V. Lertnattee, Improving centroid-based text classification using term distribution-based weighting system and clustering, International Symposium on Communications and Information terchnologies, 2001, 33-36.

[42] T. Theeramnukkong, V. Lertnattee, Effect of term distributions on centroid-based text categorization, Information Sciences 158 (2004) 89-115.

[43] K. Tzeras, S. Hartmann, Automatic indexing based on Bayesian inference networks, In Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, Pittsburgh, PA, 1993, pp. 22–34.

[44] C. H. Wan, L. H. Lee, R. Rajkumar, D. Isa, A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine, Expert Systems with Applications, 39 (2012) 11880-11888.

[45] X. Wu, V. Kumar(et. al), Top 10 algorithms in data mining, Knowledge and Information Systems, 14 (2008) 1-37.

[46] R. Xia, C. Zong, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification, Information Sciences, 181 (2011) 1138-1152.

[47] Y. Yang, An evaluation of statistical approaches to text categorization, Journal of Information Retrieval, 1 (1999) 67–88.

[48] Y. Yang, C. G. Chute, An example-based mapping method for text categorization and retrieval, ACM Transactions on Information Systems, 12(3) (1994) 252–277.

[49] Y. Yang, J. O. Pedersen, A Comparative study on feature selection in text categorization, In Proceedings of ICML-97, 14th international conference on machine learning, Nashville, TN, 1997, pp. 412–420.

[50] W. Zhang, F. Gao, An Improvement to Naïve Bayes for Text Classification, Procedia Engineering, 15 (2011) 2160-2164.

[51] R. R. Korfhage. Information Storage and Retrieval. John Wiley and Sons, 1997.

[52] Singhal, Amit . Modern Information Retrieval: A Brief Overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering (2001) 24 (4): 35–43

[53] Mohammad Golam Sohrab, Mohamed Abdel Fattah, Fuji Ren, The Best Feature Parameter and HMM for Text Summarization, Research in Computing Science, Vol.38, pp. 152-161, April, 2008

[54] Mohamed Abdel Fattah, Fuji Ren, GA, MR, FFNN, PNN, GMM based models for automatic text summarization, Computer Speech and Language, Vol. 23(1)(2009)pp. 126-144