THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Doctor of Engineering (Computer Engineering)
DEGREE

Computer Engineering	Computer Engineering
FIELD	DEPARTMENT

TITLE: Content-Adaptive Filter based Feature Selection

NAME: Mr. Part Pramokchon

THIS THESIS HAS BEEN ACCEPTED BY

THESIS ADVISOR
(Associate Professor Punpiti Piamsa-nga, D.Sc.)

COMMITTEE MEMBER
(Assistant Professor Pirawat Watanapongse, Ph.D.)

COMMITTEE MEMBER
(Assistant Professor Charay Lerdsudwihai, Ph.D.)

DEPARTMENT HEAD
(Associate Professor Anan Phonphoem, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON

DEAN
(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

CONTENT-ADAPTIVE FILTER BASED FEATURE SELECTION

PART PRAMOKCHON

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Engineering (Computer Engineering)
Graduate School, Kasetsart University
2014

Part Pramokchon 2014: Content-Adaptive Filter based Feature Selection. Doctor of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Associate Professor Punpiti Piamsa-nga, D.Sc. 92 pages.

In data representation approaches, a feature vector model using only some relevancy features, also known as optimal feature set, instead of all features yields better data analysis performance. Filter-based method is the most used feature selection method for highly dimensional data. However, there are three important drawbacks of filter-based methods. First, the optimal set depends on empirical based algorithm. Second, the ranking method ignore considering the redundancy between features which reduces the overall performance. Finally, feature scoring used for ranking feature often fails in the imbalanced class distribution.

In this paper, we proposed an algorithm to determine cutoff threshold to select an optimal feature subset using collaboration with statistical outlier detection. We present two statistic-based feature scores which are insensitive to the imbalanced class distribution based on normalization of the Document Frequency and evaluation of the difference between two sample means. We also introduced the method to reduce the effect of imbalance class dataset by separating the main classes into many subclasses and use them instead of the main class for computing feature score. Moreover, we proposed the algorithm to evaluate and eliminate redundant feature based on measuring correlation between features. Compared to feature sets selected by existing methods, the experimental results show that performance of a feature set selected by the proposed methods are comparably equal and better when it is tested on the standard dataset such as Reuter21578, RCV1v2, etc. and the optimal feature set is a compact size.

______________________________ ______________________________ ___ / ___ / ___

Student's signature Thesis Advisor's signature

ACKNOWLEDGEMENTS

First of all, I would like to gratefully thank to my advisor, Associate Professor Dr. Punpiti Piamsa-nga, for providing the basis and guidance that has enabled this work to emerge. He introduced me to the field of multimedia processing, advised me knowledge in research methodology, and worked hard in reviewing on my journal article.

I am also grateful to the other faculties of Department of Computer Engineering — Assistant Professor Dr. Pirawat Watanapogse, Assistant Professor Dr. Charay Lerdsudwichai, Assistant Professor Dr. Jittat Fakcharoenphol, Assistant Professor Dr. Chaiyaporn Jaikeaw, Associate Professor Dr. Anan Phonphoem, Associate Professor Dr. Kitsana Waiyamai, Assistant Professor Dr. Arnon Rungsawang who provided useful comments and gave me the opportunity on studying in the DCPE .

I am heartfully thankful to all my young friends of the MAD Laboratory, which provided a home with a friendly environment. Moreover, this research was formally supported by the Kasetsart University Research and Development Institute and the Graduate School of Kasetsart University.

Finally, most of all, my appreciations devote to my family for their love and encouragements on my success.

Part Pramokchon
June, 2014

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CONTENT-ADAPTIVE FILTER BASED FEATURE SELECTION

INTRODUCTION

Through the fast growth of the Internet and the wide availability of on-line text documents, the task of management text data becomes one of the principal problems. It is greatly important to explore efficient methods of text classification and clustering for automatic applications such as text filtering, text retrieval, and web-page classification. Text categorization is the process of automatically classification text documents into set of predefined categories based on the assessment of their contents. Text clustering is the task of grouping a set of text documents. The documents in the same group (called cluster) are more similar to each other than to those in other groups. Text categorization and clustering also has become one of the key techniques for analysis and management text data.

Recently, the researchers on text categorization have popularly focused on applying machine learning algorithms to classifier construction (Sebastiani, 2002;Yonghong and Jain, 1998). Many induction algorithms have learned the pattern in a text document collection, a modest number of training examples for each category, and can classify an unlabeled document. From a machine learning perspective, clustering is unsupervised learning to discover the hidden concepts of a set of patterns within the unlabeled dataset. It represents many data objects by few clusters, and hence, it models data by its clusters.

Most machine learning task with text document typically employ the Bag-of-Words (BOW) model, the vector space model in area of Information Retrieval, for representing text documents (Baeza-Yates and Ribeiro-Neto, 1999;Combarro *et al.*, 2005;Lee *et al.*, 2010;Nuntiyagul *et al.*, 2007;Sebastiani, 2002). In this model, a text document is represented as a feature vector. One dimension of the feature vector corresponds to one term which is a single word or a phrase, call as a feature, appearing in the text document collection.

The enormous number of actual features makes a major problem of text categorization (Shang *et al.*, 2007;Yang and Pedersen, 1997). It is the extremely high dimensionality of the feature space. For instance, a moderate-sized document collection may have tens or hundreds of thousands of term. The task of learning such document vector can become infeasible. Most of these features are irrelevant features which are not relative to text categorization (Combarro *et al.*, 2005). Furthermore, the effective features are frequently redundant since some features may produce the same effects to the classifiers. These irrelevant features and redundant feature will not only seriously slow down the categorization process, but also degrade the classification accuracy (Baeza-Yates and Ribeiro-Neto, 1999;Combarro *et al.*, 2005;Forman 2003;Luying *et al.*, 2005b;Makrehchi and Kamel, 2005;Yanjun *et al.*, 2008). Many researches proposed that using some best feature, a.k.a. optimal feature subset, instead of all features yields better performance both in terms of accuracy and time. The optimal feature subset consists of some relevancy features that can be used to detect the presence of a text category and to distinguish that category from all other categories.

Generally, Feature selection (FS) is an effective approach to reduce the dimensionality of feature space (Luying *et al.*, 2005b). The method explicitly choose a set of relevant feature by only selecting some best features from the set of all actual features that are most useful for compactly representing the text documents (Combarro. *et al.*, 2005;Shang *et al.*, 2007;Yang and Pedersen, 1997). Feature Selection select and extract an optimal feature subset from the original feature. Using the optimal feature subset can conserves storage and resources for the learning, classifying and clustering phase. Thus the computational efficiency can improved without sacrificing learning accuracy. Further, the effectiveness of classifier and clustering may be improved substantially, or equivalently (Aghdam *et al.*, 2009;Dai *et al.*, 2009;Forman, 2003;Shang *et al.*, 2007).

Feature selection methods can be classified as filter-based approaches or wrapper-based (Foithong *et al.*, 2012;Forman 2003;Ruiz *et al.*, 2005;Sebastiani, 2002;Yu and Liu, 2004). Filter-based methods selects features using “usefulness” of each individual feature for learning algorithm (Makrehchi and Kamel, 2011;Rangarajan and

Veerabhadrappa, 2010;Sebastiani, 2002;Shamsinejadbabki and Saraee, 2011;Yu and Liu, 2004). There are many feature scoring functions that can be used to determine the usefulness such as Document Frequency (DF), Information Gain (IG) and CHI-square (CHI) (Combarro *et al.*, 2005;Forman 2003;Lee and Chen, 2006;Yang and Pedersen, 1997), and etc. These feature scoring rely on various measures of the general characteristics of the training examples such as distance, information, dependency, and consistency (Luying, 2005b). Features are ranked and selected according to their score. Top-ranked features are selected as an optimal feature subset using a predefined cutoff threshold. The threshold for getting the optimal number mainly depends on the prior knowledge that can achieve better performance in some specific environments. There is also no theoretical support to determine the optimal cutoff threshold (Dai *et al.*, 2009;Rangarajan and Veerabhadrappa, 2010;Soucy and Mineau, 2003). On the other hand, the experimental methods are needed to find out the threshold (Dai, *et al.*, 2009).

Wrapper-based methods employ a search algorithm to determine an optimal set of features. However, exhaustive search is NP-hard. A popular search algorithms used in text classification is sequential forward selection (Ruiz *et al.*, 2005;Yu and Liu, 2004). It iteratively increases one feature at a time to merge into its collection until there is no significant improvement in classification accuracy. Wrapper-based methods generally outperform filter-based methods in that the optimal feature subset is usually appropriate to a chosen learning algorithm (Liu and Yu, 2005;Rangarajan and Veerabhadrappa, 2010;Ruiz *et al.*, 2005;Sebastiani, 2002;Somol *et al.*, 2010). However, the wrapper approach is usually not suitable when efficiency (in time or computer resources) is a constraint (Arauzo-Azofra *et al.*, 2008;Makrehchi and Kamel, 2011). On the another hand, filter-based method is fast and available and the selected features are classifier independent; therefore, it is easy to scale up for machine learning research with high-dimensional datasets

The ranking wrapper based with sequential forward selection (SFS) is the hybrid approach which integrates advantage of both filter and wrapper approaches. Firstly, it evaluates features by using feature scoring and ranking to find the best subset and then using a learning algorithm to find the final best subset (Duangsoithong and

Windeatt, 2009). The approach empirically uses the classification accuracy of a predetermined learning algorithm to measures the usefulness of feature subset from ranking manner (Luying *et al.*, 2005b;Sebastiani, 2002). Although it efficiently searches for optimal number of features better suited to the learning algorithm aiming to improve performance, but it also trends to be more computationally expensive to reiteratively run through the feature subset space (Aghdam *et al.*, 2009;Combarro *et al.*, 2005;Duangsoithong and Windeatt, 2009;Forman, 2003;Luying *et al.*, 2005b;Sebastiani, 2002). The predetermined threshold, which is computed by this approach, varies with various feature scoring and training data. Moreover, the selected optimal feature subset may not be suitable for other classifiers.

Despite the impressive achievements in the current field of feature selection, high-dimensional data often contains many redundant features. Some effective features may produce the same effects to the classifiers. For example, many terms in English or other human languages can be synonym; such terms with identical meanings are potentially “similar”. Existing Feature selection is incapable of removing redundant features because redundant features likely have similar rankings. As long as features are deemed relevant to the class, they will all be selected even though many of them are highly correlated to each other. Several feature selection approaches are used to reduce these redundant features based on the removing some highly similar word or the notion of grouping similar features into a much smaller number of feature-cluster, and use these clusters as features. However, the computational cost of the existing method is very intensive. Therefore the crucial problem in such procedures is how to efficiently quantify the similarity of features and effectively remove these redundant features.

The most favorable measures of feature score, such as Information Gain and Chi-square (Forman, 2003;Makrehchi and Kamel, 2011;Uchyigit and Clark, 2007;Yang and Pedersen, 1997), often fail to produce good performance when it is applied for classifying instances into the problem of the multi-class text classification with the class imbalance problem (also called as the skewed class distribution problem). The performance, especially the recall rate, of the classification is usually dropped due to the number of features and number of text documents in each class is drastically different in real life (Forman, 2003;He and Garcia, 2009;Makrehchi and Kamel, 2011;Zheng *et al.*, 2004). The feature ranking methods tend to rank features in large classes (majority), while those in small classes (minority), which are difficult to be learned, are rarely considered(Forman, 2003;He and Garcia, 2009;Makrehchi and Kamel, 2011;Zheng *et al.*, 2004). This not only leads to a low recall rate for the smaller classes but also reduces overall performance of the multi-class text classification.

Many recent researches have proposed many solutions for dealing with the imbalance data problem both for standard and for ensemble learning algorithms. They can be categorized into three groups (He and Garcia, 2009;López *et al.*, 2013). The first is data preprocessing including re-sampling in which the training datasets are modified in such a way to produce a more or less balanced class distribution that allow classifiers to perform in a similar manner to standard classification. The second is cost-sensitive learning which considers higher costs for the misclassification of examples of the minority class with respect to the majority class for trying to minimize higher cost errors. The third is algorithm modification which aims to the adaptation of standard learning methods to be more adjusted to class imbalance issues. Although, there is no single best approach to deal with imbalance problems, the re-sampling methods have been shown to be very successful in many researches (Jo and Japkowicz, 2004;Kihoon and Kwek, 2005;López *et al.*, 2013). However, the re-sampling methods may also introduce the problem of important concept missing, overfitting, and overgeneralization (He and Garcia, 2009).

In this thesis, several methods for handling these problems as mentioned earlier are proposed to obtain the optimal feature subset depended on content in the text document collection. Firstly, we introduce an idea of applying outlier detection to determine optimal cutoff threshold for improving efficiency of feature selection method. Outlier detection is a statistical approach that observes and indicates group of outlier, the observations which have behavior very different from the remainder of other observations in data collection (Aggarwal and Yu, 2005;Carter *et al.*, 2009;Hodge and Austin, 2004). Based on Outlier Detection, proposed feature selection algorithm explicitly selects and groups the feature outliers that are highly indicative of relevance for each category as an optimal feature subset (Ben-Gal, 2005;Seo, 2006).

Some supervised feature selection methods have been successfully used in text classification (Sebastiani, 2002;Yang and Pedersen, 1997). For data clustering, which class label is not available, many research projects introduce some unsupervised feature selection such as Document Frequency (DF), Term Contribution (TC), Term Variance (TV) and Mean Median (MM). Some researchers proposed to reduce redundancy and irrelevance for the feature selection algorithm, such as; however, class label is required and it cannot be used for data clustering directly.

Next, we proposed the efficient and effective feature selection algorithm for data clustering. The methods are filter-based and unsupervised approaches which do not need feature-class information. Furthermore, we integrate concept of evaluating feature redundancies into the proposed algorithms. In the propose algorithm, Unsupervised Fast Correlation based Feature Selection (UFCBF), the redundancy assessment is a feature correlation measurement based on the geometric view. The UFCBF algorithm is not an empirical process and no need expertise knowledge. UFCBF automatically predefines two theoretical-support thresholds by using a statistical coefficient of confident to identify redundant features and to remove them.

Two imbalanced-class insensitive scores are proposed for problem of imbalanced class distribution. We introduce the first novel feature scoring called Document Frequency Difference (DFD) which is based on the difference between the Z-score of the feature document frequency in one category and the other categories (Johnson and Wichern, 2002;Tamhane and Dunlop, 2000). For each category, a feature, whose document frequency in the category is higher than that in the other categories, is highly informative since it is relevant with respect to this category. Moreover, we apply the statistical t-test technique to compute the second feature score called t-Test score. This score is based on the idea that features may discriminate particularly well between two classes if occurrence frequencies from both classes are significantly different. Given two data sets (one specific class and others), which are characterized by means, standard deviations, data sizes and their population distributions are assumed to be normal, t-Test score can be used to determine whether number of data in each class are distinct or not. Because the t-Test function normalizes the mean difference value by the common standard deviation of the two class classes. Therefore, it can be estimated on the unequal classes unlike other feature scores.

Finally, we proposed an effective data-preprocessing technique to reduce the effect of imbalanced class distribution. Technically, large classes of training dataset are clustered by clustering algorithm into k subclasses and those k subclasses are used to be replaced with main class for computing feature scores. We use two most popular clustering algorithm, K-mean clustering (MacQueen, 1967) and hierarchical agglomerative clustering (Hastie *et al.*, 2008;Manning *et al.*, 2008). Unlike most of the resampling paradigm such as under-sampling, over-sampling etc. (He and Garcia, 2009), proposed score computing method avoids the problem of important concept missing, overfitting, and overgeneralization (He and Garcia, 2009) by considering both within-class and between-class relationships through the clustering method. The experimental results are quite promising. The K-mean clustering is faster algorithm but the method has to get an appropriate K value by expertise determination or iteratively empirical experiments or sophisticate approaches and its result is non-deterministic. The hierarchical agglomerative clustering (HAC) integrated with class-sensitive weighting for determination cutoff is slower than K-mean but the method does not require expertise knowledge or iterative empirical experiments to define the number of k clusters and easy for implementation; and its resulted clusters are deterministic.

Contributions

The contribution of research consists of:

1. The method for defining cutoff threshold for filter based feature selection.
2. Two feature scoring techniques for text classification.
3. Two data-preprocessing methods for feature scoring for text classification.
4. The unsupervised feature selection algorithms for text clustering.

All approaches are proposed in manner of filter-based methods which are non-iteration empirical processing, thus, can obtain automatically the optimal feature sub-set which is dynamic and adaptable to the content of text document in dataset.

Thesis Organization

The further document is detailed as follows:

1 OBJECTIVES describe about objectives of the research.

2 LITERATURE REVIEW presents and discusses the background of Text Categorization and Clustering; and some of the approaches adopted for Feature Section in the past are summarized

3 MATERIALS AND METHODS explains about

3.1 Dataset

3.2 Proposed methods associated with the feature selection and theirs characteristics.

3.2.1 Outlier based cut-off determination for feature ranking method

3.2.2 Document Frequency Differential Feature scoring

3.2.3 T-test Feature scoring

3.2.4 K-mean clustering based feature scoring method

3.2.5 Hierarchical agglomerative clustering based feature scoring method

3.2.6 Unsupervised Fast correlation-based feature selection for clustering

4 RESULT AND DISCUSSION explains the detail of our experiment method including the document collection, the classifier and the performance measurement.

5 CONCLUSION provides conclusions and further directions about the deep analyze in some interesting topics.

OBJECTIVES

To find efficient and effective methods to select the most appropriate feature set for text document analysis by considering the content within the dataset itself.

LITERATURE REVIEW

This section details about text categorization feature, feature selection and feature extraction techniques used in previous research. A summarized of some previous studies with different classifiers and its performance has detailed.

1. Text Categorization/Text Classification

Multi-class text categorization (classification) is an automatic process for assigning a category in a finite set of predefined categories to text documents. From the perspective of machine learning, it can be regarded as a supervised learning problem. Given a supervised training set of labeled text documents, the goal is to induce a classifier based on the assessment of their content to correctly label the class of a new unlabeled document (also known as *testing set*).

Text categorization is the task of assigning a Boolean value to each pair of document $d_j$and category c_j, $\langle d_j, c_k \rangle \in D \times C$, which$C = \{c_1, \dots, c_{|C|}\}$ be a set of categories and *D* is a set of unseen documents. A value of True (T) assigned to $\langle d_j, c_k \rangle$ indicates a decision to label $d_j$under c_k, while a value of False (F) indicates a decision not to label $d_j$under c_k. A classifier (a.k.a. rule or hypothesis, or model) for c_k is then a function$\Phi_k = D \rightarrow \{T, F\}$. The single labeling (a.k.a. non-overlapping categories) is the case which exactly one category must be assigned to each $d_j \in D$. The special case of single labeling is binary text categorization, in which each $d_j \in D$ must be assigned either to category $c_k$or to its complement$\overline{c_k}$. Since an algorithm for binary classification can also be used for multi-class classification. We will view the multi-class classification under $C = \{c_1, \dots, c_{|C|}\}$ as consisting of $|C|$ independent problems of binary classifying the documents in D under a given category c_k for $k = 1, \dots, |C|$.

1.1 Binary class Text Categorization

Definition 1 Let $c \in C = \{spam, legitimate\}$, U be a set of unseen incoming messages and a message $m \in U$. Email message classification is to construct a binary classifier, denoted by $\Phi(\cdot)$, for c such that:

$$\Phi(m) = \begin{cases} 1 & if\, d(m) > 0, \\ -1 & otherwise, \end{cases}$$

where $\Phi(m) = 1$ implies that m belongs to c (*i.e., spam*) and $\Phi(m) = -1$ implies that m does not belong to it (i.e., legitimate). $d(\cdot) \in \Re$ is a decision function. The goal of constructing a binary classifier, $\Phi(\cdot)$, is to correctly approximate the unknown target function $\breve{\Phi}(\cdot)$, so that $\Phi(\cdot)$ and $\breve{\Phi}(\cdot)$are as coincident as possible. Note that, every classifier Φ_i has its own decision function $d_i(\cdot)$. If there are many different classifiers, there will be many different decision functions.

1.2 Multi-class Text Classification

Multi-class Text categorization also is the task of assigning a Boolean value to each pair of document $d_j$and category c_j, $\langle d_j, c_k \rangle \in D \times C$, which$C = \{c_1, \dots, c_{|C|}\}$ be a set of categories and D is a set of unseen documents. A value of True (T) assigned to $\langle d_j, c_k \rangle$ indicates a decision to label $d_j$under c_k, while a value of False (F) indicates a decision not to label $d_j$under c_k. A classifier for c_k is then a function $\Phi_k = D \rightarrow \{T, F\}$. The single labeling (a.k.a. non-overlapping categories) is the case which exactly one category must be assigned to each $d_j \in D$ where $|C| > 2$.

2. Text Clustering

Due to the increase in the size of text datasets, human manual labeling has become extremely difficult and expensive. Therefore, automatic labeling has become indispensable step in data mining. Data clustering is one of the most popular data labeling techniques. In text clustering, there are given unlabeled text data, and it is the technique that aims to group similar text documents into one group classed a cluster.

Each cluster has maximum within-cluster similarity and minimum between-cluster similarity based on certain similarity index. In other words text clustering is the unsupervised classification of samples into clusters. Usually, neither cluster's description nor its quantification is given in advance unless a domain knowledge exists, which poses a great challenge in clustering analysis.

We can defined the goal in classical clustering as follows, given (i) a set of documents $D = \{d_1, \ldots, d_n\}$, (ii) a predefined number of clusters K, and (iii) an objective function that evaluates the quality of a clustering, we want to compute an assignment $\gamma: D \rightarrow \{1, \ldots, K\}$ that minimizes (or, in other cases, maximizes) the objective function. The objective function is often defined in terms of similarity or distance between documents. For example, the objective in K-means clustering (described below) is to minimize the average distance between documents and their centroids or, equivalently, to maximize the similarity between documents and their centroids.

3. A Formal Representation of Text Document

Text documents cannot be directly interpreted by a classifier or a clustering algorithm. Therefore, document preprocessing is necessary to maps a document d_j into a compact representation of its content needs to be uniformly applied to training, validation and test documents. As we have already mentioned in Section 1, the BOW model is the most widely adopted in Text document analysis. It consists in regarding a document as a sequence of terms assuming their independence and ignoring ordering and text structure. Every message is represented by a feature vector, called the document vector space, where each dimension corresponds to a term assuming term independence and ignoring ordering or textual structure. The term (a.k.a. feature) is typically a single word, a multiple keywords, or a phrase.

3.1 Text Feature weighting

Let $Tr = \{d_1, \ldots, d_{|Tr|}\}$ is the training set of text documents. Every text document d_j is usually represented as a feature weight vector $\bar{d}_j = [w_{1,j}, \ldots, w_{|F|,j}]$, where F is a set of original distinct features that occur at least once in at least one document of data collection. Feature weight is a weight associated with the pair$\langle f_i, d_j \rangle$, where$f_i \in F$. The feature weights $w_{i,j}$ can be either binary or non-binary representation. This weight quantifies the importance of the feature f_i for describing semantic content of the document d_j. Therefore, distinct features have varying relevance when they are used to describe contents of documents. If a feature $f_i$occurs in a document d_j, $w(f_i, d_j) > 0$. If a feature f_i does not occur in a document d_j, then $w(f_i, d_j) = 0$. This effect is captured through several different way of computing the occurrence values and assigned these numerical weights to each term of a document.

3.2 Term Frequency Inverse Document Frequency (TFIDF) weighting

In this paper we focus on the non-binary representation, which a weight $w_{i,j}$ is quantified by measuring the raw frequency of a feature f_i inside a document d_j. Such Term Frequency (TF) factor provides measurement of how well that feature (term) describes the document contents as the essential characteristic within the document. Furthermore, a weight $w_{i,j}$ is also quantified by measuring the inverse of the frequency of a feature f_i among the document in the data collection. This factor is usually referred as the Inverse Document Frequency (IDF) factor. The motivation for usage of an IDF is that features, which appear in many documents, are not very useful for distinguishing a relevant document from a non-relevant one.

Definition 2 Let $freq(f_i, d_j)$ be raw occurrence frequency of feature f_i in the document d_j (i.e., the number of times the feature f_i is mentioned in the text of the document d_j). Then TF factor, the normalized frequency $freq(f_i, d_j)$ of feature f_i in the document d_j, is given by$tf(f_i, d_j) = \frac{freq(f_i, d_j)}{\sum_i freq(f_i, d_j)}$, where $\sum_i freq(f_i, d_j)$ is the sum of number of occurrences of all features in the document d_j. If the feature f_i does not appear in the document d_j, then $tf(f_i, d_j) = 0$.

Further, let inverse document frequency for feature f_i, be given by $idf(f_i) = \frac{|Tr|}{DF(f_i)}$where $|Tr|$ is the total number of document in Tr and $DF(f_i)$ be the number of document in which the feature f_i occurs. Finally, the best known term-weighting schemes use weights which are given by $w(f_i, d_j) = tf(f_i, d_j) \times idf(f_i)$.

4. Problem of the High dimensionality of feature vector space

4.1 Curse of Dimensionality

One major difficulty in Text analysis is the high dimensionality of the vector space. Due to the rapid growth of the text document content and language usage changes overtime, there are many distinct features in original feature set F, which can be tens or hundreds of thousands features of even a moderate-sized text document collection. The size of original feature set F, $|F|$ is the dimensionality of the vector space. Obviously, the computational complexities of most learning algorithms especially depend on the dimension of the input space. Hence the learning task such document vectors become infeasible. In theory, the more features should provide the more discriminating power, however in practice, excessive features will not only significantly slow down the learning process but also degrade the accuracy. In other words, the bottleneck of the classification/clustering task is the poor efficiency and effectiveness caused by the high dimension of the feature space.

4.2 Irrelevant Relevant and Redundant Feature

Based on a review of previous definitions of feature relevance in data classification (Liu, H. and L. Yu, 2005;Yu, L. and H. Liu, 2004), whole features are classified into three disjoint categories, namely, relevant, and irrelevant features. Let F be a full set of features, f_i be a feature, and a feature subset $S_i = F - \{f_i\}$. We use $P(c|f_i)$ to denote the posterior probability of document class c_k given the value of feature f_i, $c_k \in C$, and c'_k denote the contrary class of c_k. These categories of relevance can be formalized as follows.

Definition 3 (Relevance). A feature f_i is relevant if and only if $P(c_k|f_i) > P(c_k'|f_i)$ and $P(c_k|S_i, f_i) > P(c_i|S_i)$

Definition 4 (Irrelevance). Let s_i be a subset of $S_i(s_i \subseteq S_i)$. A feature f_i is irrelevant if and only if $\forall s_i(P(c_k \mid s_i, f_i) = P(c_k \mid s_i))$

Relevance of a feature indicates that the feature is always necessary for an optimal subset; it cannot be removed without affecting the original conditional class distribution. Irrelevance indicates that the feature is not necessary at all. An optimal subset should include all relevant features, none of irrelevant features.

In the other research domains, high-dimensional data often contains many redundant features. Both theoretical analysis and empirical evidence show that along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms and thus should be eliminated as well. In many text categorization researches, the notion of feature redundancy was introduced as such semantic similarity between words. We now formally define feature redundancy.

Definition 5 (Redundancy). Given two synonym features f_i, f_j where $i \neq j$. There is a redundancy between any pair of features f_i, f_j if and only if $P(c_k \mid f_i) = P(c_k \mid f_j)$.

Several approaches are to reduce these redundant features based on the notion of grouping similar features into a much smaller number of feature-cluster, and use these clusters as features. The crucial problem in such procedures is how to quantify the similarity of feature. Not only features are difficult to similarity determination, but also it can be used changeably over time cause by concept drifting and spammer’s rule avoiding. Moreover, manual dictionary/thesaurus-based method for capture the new words and senses is costly if not impossible.

5. Dimension Reduction by Feature selection Approaches

Feature Selection is widely adopted approach for dimensionality reduction in supervised learning. It is based on selecting a set of relevant feature as an optimal subset from the original set of features. In Text Classification and Clustering, this will be formulated into the problem of identifying the most informative terms within a set of documents for classification. In the rest of the section, the details of feature selection for Text Classification are described.

Based on a review of definition of feature relevance (Yu and Liu, 2004), the terms can be divided into two disjoint types, namely, relevant, and irrelevant term. The relevant term is highly importance for the Text Classification and Clustering task and always necessary for an optimal subset; it cannot be removed without affecting the classification result. On the other hand, irrelevant term is not necessary at all. An optimal subset should include all relevant features, none of irrelevant features. Therefore, selecting relevant term and removing irrelevant term not only reduces the high dimensionality of the term vector space, but also provides a better data understanding, which improves the classification result.

This paragraph describes applying feature selection in machine learning. More over we demonstrate the framework of machine learning consist of learning process and object identify process. Next is the detail about types of feature selection. In the final we mention why the filter-based is suitable for applying in the multi-class text analysis

5.1 Feature scoring

Most Feature selection methods for text document use a feature scoring function to measure term relevance for classification task scoring rely on the general characteristics of the training examples such as distance, information, dependency, and consistency. Feature selection method score each term according to feature scoring, and then select the predefined number from the top-ranked terms.

5.1.1 Document Frequency (DF)

DF is a simply measures in how many documents the feature occurs. Since it can be computed without class labels, it may be computed over the entire test set as well. Selecting frequent words will improve the chances that the features will be present in future test cases.

$$DF(f_i) = \left|\left\{d_j \middle| f_i occur\ in\ d_j, d_j \in Tr\right\}\right|$$

5.1.2 Global Information Gain (IG)

Information gain is frequently employed as a feature usefulness criterion in the field of machine learning. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. Let $|C|$ denote the number of categories. The information gain of term $f_i$is defined as follow:

$$IG(f_i) = -\sum_{k=1}^{|c|} P(c_k) \cdot logP(c_k)$$

$$+P(f_i) \cdot \sum_{k=1}^{|c|} P(c_k|f_i) \cdot logP(c_k|f_i) + P(f_i') \cdot \sum_{k=1}^{|c|} P(c_k|f_i') \cdot logP(c_k|f_i')$$

where $P(c_k|f_i)$ is the conditional probability that the feature f_i occurs in category c_k, and $P(c_k|f_i')$ is the conditional probability that f_i does not occur in c_k. IG is commonly applied in text categorization which uses the class information to compute the score. After computing these feature scores, The Feature Ranking method rank features by their scores. The feature ranking method selects the optimal feature with respect to the feature scoring.

5.1.3 χ^2Statistic (CHI)

The χ^2statistic measures the lack of independence between f_i and c_k and can be compared to the χ^2 distribution with one degree of freedom to judge extremeness.

It is defined as follow:

$$\chi^2(f_i, c_k) = \frac{|Tr| \cdot [P(f_i, c_k) \cdot P(f_i', c_k') - P(f_i, c_k') \cdot P(f_i', c_k)]^2}{P(f_i) \cdot P(f_i') \cdot P(c_k) \cdot P(c_k')}$$

Besides DF and IG, CHI are specified "locally" to a specific category c_k; in order to assess the value of a feature f_i in a "global" category-independent sense, typically we can calculate one of two ways as follow:

$$\chi^2_{avg}(f_i) = \sum_{k=1}^{|C|} P(c_k)\chi^2(f_i, c_k),$$

$$\chi^2_{max}(f_i) = max_{i=1}^{|C|}\{\chi^2(f_i, c_k)\}$$

In (Yang and Pedersen, 1997), the results reported that CHI and IG more effective for multi-class classification result, and DF a better choice for efficiency and scalability if a small disgrace in effectiveness is acceptable.

5.1.4 Term Contribution (TC)

Term contribution takes the term weight into account. Because DF assumes that each feature is of same importance in different documents, it is easily biased by those common features which have high document frequency but uniform distribution over different classes. TC is proposed to deal with this problem. The result of text clustering is highly dependent on the documents similarity. So the contribution of a feature can be viewed as its contribution to the documents' similarity. The similarity between documents d_i and d_j is computed by dot product:

$$sim(d_i, d_j) = \sum_{f} w(f, d_i) \times w(f, d_j)$$

Where *w(t,d)* represents the *tf*idf* weight of feature *f* in document *d*. So we defined the contribution of a feature in a dataset as its overall contribution to the documents' similarities. The equation is

$$TC(f_k) = \sum_{i,j \cap i \neq j} w(f_k, d_i) \times w(f_k, d_j)$$

5.1.5 Term Variance (TV)

Term Variance is introduced to evaluate the quality of feature. That is to compute the variance of every term in all dataset. Methods like DF assume that each feature is of same importance in different document, it is easily biased by those common features which have high document frequency but uniform distribution over different classes. TV follows the idea of DF that the terms with low document frequency is not important and can solve the problem above at the same time. A feature appears in very few documents or has uniform distribution over documents will have a low TV value. The quality of the feature is measured as follows:

$$TV(f_i) = \sum_{j=1}^{|Tr|} \left[w(f_i, d_j) - \overline{w(f_i)} \right]^2$$

where |Tr| is the number of documents in training set in which f_i occurs at least once and $\overline{w(f_i)}$is the average weight value of feature f_i.

5.1.6 Mean Median (MM)

Mean Median is dispersion measures proposed to compute the absolute difference between the mean and median of f_i, given by

$$MM(f_i) = \left| \overline{w(f_i)} - median\left(w(f_i)\right) \right|$$

Although this is not a classical dispersion measure, the result in show that it adequate for FS for the data clustering.

5.2 Feature selection strategy

By nature of the algorithm, a filter-based method needs a threshold from empirical results as a cut-off for the number of relevant features. Many strategies have been proposed (Soucy and Mineau, 2003) for feature selection based mainly on observation or experience. On the other hand, experimental methods are needed to find this threshold (Dai *et al.*, 2009). In other words, finding the optimal size of the feature subset requires repeated experiments, and cannot be selected based solely on expert observation.

The ranking wrapper-based method with sequential forward selection (Blachnik *et al.*, 2009;Ruiz *et al.*, 2005) is one of the most common methods that empirically uses the classification accuracy of a predetermined learning algorithm to measures the goodness of the selected feature subset. This is a heuristic method that evaluates features using feature scoring to find the set of top-ranked terms and then using a learning algorithm to find the final number of top-ranked terms. This algorithm is shown as below.

Based on analysis in (Luying *et al.*, 2005b), which describes a typical strategy for selecting a subset of original features, the ranking wrapper algorithm is explained as follow. The algorithm consists of three basic steps: subset generation, subset evaluation, and stopping criterion. Subset generation iteratively produces candidate feature subsets for evaluation based on feature scoring and ranking. All features are ranked according to their scores. The candidate feature subset consists of a number of top-ranked features. Then sequential forward selection starts with an empty set and successively adds p terms at each loop. Each candidate subset is evaluated and compared with the previous best one according to a certain evaluation criterion. The wrapper-based method requires a predetermined classifier and uses its classification accuracy as the evaluation criterion. If the accuracy of the new subset turns out to be better, it replaces the previous best subset. The process of subset generation and evaluation is repeated until a given stopping criterion, such as accuracy drop or predefined number of features or use whole features, is satisfied. This is actually the greedy hill-climbing approach. The best feature set selected by this approach approximates the optimal feature subset. This ranking wrapper algorithm tends to give superior performance sets as it can search for the terms that are appropriate to the learning algorithm. Hence, this is one way to improve the performance of the classifier.

Algorithm 1 Ranking Wrapper Algorithm

```
Input : Original feature set ($T$)
        Training data ( $Tr$)
        Predetermined Classifier ($A$)
        Number of features to add in one step sequential forward ($p$)
Output : Optimal feature subset ($Opt$)
$|T| \leftarrow GetNumberOfTerm(T)$
Compute Score step:
for each feature $t_i$ in $T$ do
    $s_i \leftarrow ComputeScore(t_i, Tr)$
$ST \leftarrow SortedFeatureByScore(s_i)$
Selection step:
$Opt = \emptyset$
$BestClassification = 0$
for $k$ from $p$ to $|T|$    // Stopping Criterion
    $tmpOpt \leftarrow GetTerm(ST, k)$   //Subset Generation
    $TmpClassification \leftarrow WrapperClassification(Opt, Tr, A)$
    //Subset Evaluation
    if ($BestClassification < TmpClassification$) then
        $Opt = tmpOpt$
        $BestClassification \leftarrow TmpClassification$
    end if
    $k = k + p$
end for
return $Opt$;
```

6. Feature Selection Problems

6.1 Cut-off Threshold is hard to be predefined

This ranking wrapper-based method for feature selection has two problems. First, the wrapper-based method is computationally intensive, even though the greedy sequential search reduces the search space from $O(2^{|T|})$ to $O(|T|)$, but it tends to be more computationally expensive to iteratively run through the feature subset space (Foithong *et al.*, 2012;Ruiz *et al.*, 2005;Yu and Liu, 2004). Second, the sequential forward search is likely to stop at the local optima (Foithong *et al.*, 2012;Luying *et al.*, 2005b). Finally, the predetermined threshold, which is computed by this approach, varies with variance of feature scoring and training data (Dai *et al.*, 2009;Luying *et al.*, 2005b;Somol *et al.*, 2010;Yang and Pedersen, 1997). Hence our research attempts to develop new feature selection methods which can overcome these problems.

6.2 Feature Redundancy affects performance

Notations used in this paper are as follows: $F = \{f_1, \ldots, f_{|F|}\}$ be a set of original distinct features; $D = \{d_1, \ldots, d_{|D|}\}$ be a training dataset; A training sample d_j is represented as a feature weight vector, $\overrightarrow{d_j} = [w(f_1, d_j), \ldots, w(f_{|F|}, d_j)]$. This feature weight $w(f_i, d_j)$ quantifies the importance of the feature f_i for describing semantic content of d_j.

For example, text clustering prefers feature weight such as the Term Frequency or the Term Frequency Inverse Document Frequency (Alelyani *et al.*, 2013;Almeida *et al.*, 2009;Ferreira and Figueiredo, 2011;Lewis *et al.*, 2004;Liu *et al.*, 2003;Luying *et al.*, 2005a;Yanjun *et al.*, 2008). Feature redundancy can be represented in terms of feature correlation. It is widely accepted that the features are redundant to each other if their values are completely correlated (Ferreira and Figueiredo, 2012;Guyon *et al.*, 2003;Yu and Liu, 2004). The traditional filter-based feature selection is incapable of removing redundant features because redundant features likely have similar rankings. As long as features are deemed relevant to the class, they will all be selected even though many of them are highly correlated to each other. For high-dimensional data which may contain a large number of redundant features, this approach may produce results far from optimal.

Many research projects address some similarity/correlation/redundancy measures that have been used for feature selection, such as correlation coefficient (Bache and Lichman, 2013;Ferreira and Figueiredo, 2012;Yu and Liu, 2004), symmetrical uncertainty (Ferreira and Figueiredo, 2012;Yu and Liu, 2004) and absolute cosine (Ferreira and Figueiredo, 2012). In (Ferreira and Figueiredo, 2012) argue that using angles to measure similarity is better suitable for high-dimensional sparse data. Therefore, absolute cosine is used as geometric view in our proposed algorithm.

6.3 Feature score is sensitive to imbalanced class distribution

Class imbalance of dataset is a situation that number of entities in each class is drastically different referred to as a between-class imbalance shown in Figure 1 (He and Garcia, 2009; López *et al.*, 2013). Typically, classifiers are optimized for overall accuracy without taking distribution of class sizes into account. Thus, the larger classes are dominant by the higher frequencies of support data; and the smaller classes become statistically unimportant. Most of the researches on several classifications in imbalance domains have shown significantly effect of the between-class imbalance on the performance (He and Garcia, 2009; López *et al.*, 2013). However, there are several investigations which also suggest that other factors such as the presence of within-class imbalance, multiple concepts, overlapping, or lack of representative data can contribute to such performance depreciation as well ((He and Garcia, 2009;Jo and Japkowicz, 2004;López et al., 2013). These factors are illustrated in Figure 1.

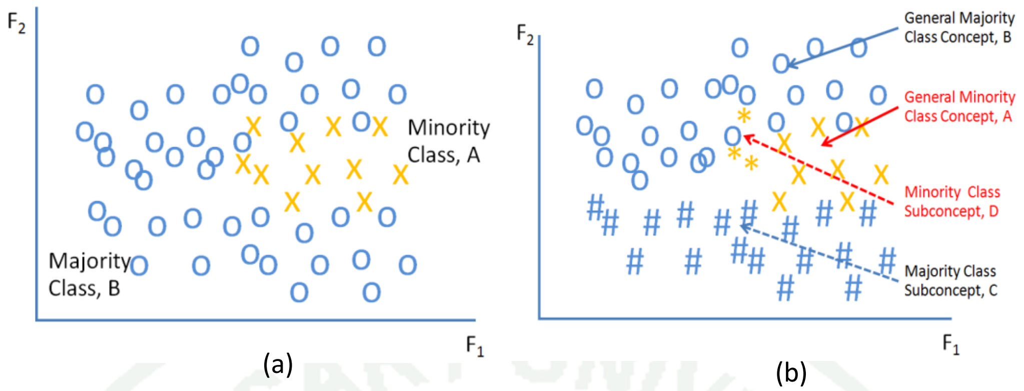
F2
Minority Class, A
Majority Class, B
F1
(a)
F2
General Majority Class Concept, B
General Minority Class Concept, A
Minority Class Subconcept, D
Majority Class Subconcept, C
F1
(b)

Figure 1 A high complexity data set with both between-class and within-class imbalance

Feature ranking methods also fail when applied to multiple classes with non-uniform class distributions (class skew) as well. The method pays more attention to the large classes to compensate for their weakness that tends to ignore small classes. Thus, the relevant features for large classes can be selected but features for small classes are ignored. In (Makrehchi and Kamel, 2011), it has been showed that Information Gain assigns higher scores to the features correlated with large classes and such that small classes are under-represented in top-rank features. Their experimental result also showed the overall accuracy is improved even though there is a low recall rate for small classes. The experimental results in (Forman, 2004) have also concluded that other scores have suffered to these kinds of problems as well.

There are many approaches proposed to handle with the class imbalance problem. These approaches can be categorized into three groups (He and Garcia, 2009; López *et al.*, 2013). First, the external approaches that process the data before learning task in order to decrease the effect of their class imbalance. Most of the approaches are re-sampling techniques which is a preprocessing step in order to balance the class distribution. These works have proved empirically that is a useful solution and the further advantage of these techniques is that they are independent of the underlying classifier. Second, the cost-sensitive learning which incorporates both the data and algorithmic level approaches assume higher misclassification costs for samples in the minority class and seek to minimize the high cost errors. The precise value of misclassification cost is necessarily achieved by domain experts, or can be learned via other approaches.

Third, ensemble-based method that is based on the combination between the ensemble learning algorithm and one of the previous techniques such as data approaches, existing learning, or cost-sensitive learning solutions. The basic idea of ensemble-based classifiers is to construct several classifiers from the original data and then aggregate their predictions with the unknown instances entered. The ensemble-based classifier, also known as multiple classifier system, can improve the performance of single classifiers by inducing several classifiers and combining them to obtain a new classifier that outperforms every one of them.

All groups described above have different particular drawbacks which make them appropriate for a given application. The ensemble methodologies have shown very accurate result, but their learning time may be high and the output model can be difficult to understand by the final user. Cost-sensitive approaches have also shown very precise result, but the necessity of defining an optimal cost-matrix imposes hard restrictions to their use. Finally, the preprocessing algorithms have shown their robustness and obtained very good global results, and therefore they can be viewed as a standard approach for imbalanced datasets (López *et al.*, 2013).

However, the work in (Forman, 2003;Zheng et al., 2004) argues that feature selection should be relatively more important than classification algorithms in highly imbalanced situations, and report that feature selection can significantly improve the performance of classification of imbalanced data. Therefore, instead of balancing the training data, enhancement selecting the most useful features according to the imbalanced data provides an alternative to handle the imbalanced data problem.

6.4 Supervised feature selection cannot be applied to data clustering directly

There are two types of filter-based feature selections, supervised and unsupervised methods.. For supervised methods, class label, which information to identify class of a data entity, must be provided. Some supervised feature selection methods have been successfully used in text classification (Sebastiani, 2002;Yang and Pedersen, 1997), such as Information Gain (IG) and x2 Statistics (CHI). However, if class label is not available, many research projects introduce some unsupervised feature selection such as document frequency (DF) (Liu *et al.*, 2003;Sebastiani, 2002;Yang and Pedersen, 1997), term contribution (TC) (Almeida *et al.*, 2009;Liu *et al.*, 2003;Luying *et al.*, 2005a;Zonghu *et al.*, 2011), term variance (TV) (Ferreira and Figueiredo, 2012;Luying *et al.*, 2005a;Zonghu *et al.*, 2011) and Mean Median (MM) (Ferreira and Figueiredo, 2012). Some researchers proposed to reduce redundancy and irrelevance for the feature selection algorithm, such as (Yu and Liu, 2004); however, that methods is still supervised filter-based algorithm. In other words, class label is required and it cannot be used for data clustering directly.

6.5 Proposed method to tackle the problems

To tackle these problems, we formulated statistical techniques for our relevancy signature extraction and selection issues:

1. "Outlier based Feature Selection Algorithm" issue is based on outlier detection that proposed feature selection algorithm explicitly selects and groups the feature outliers that are highly indicative of relevance for each category as an optimal feature subset. Our proposed method uses a coefficient of confident to select relevant features, thus the algorithm is non-iteration empirical processing which can obtain automatically the optimal feature subset which is dynamic and adaptable to the content of text document in dataset.

2. "Unsupervised Fast correlation-based feature selection for clustering" (Pramokchon and Piamsa-nga, 2014a) issue is based on integrating concept of evaluating feature redundancies into the proposed algorithm. The redundancy assessment is a feature similarity measurement based on the geometric view of features. Our proposed method also uses a coefficient of confident to remove redundant features.

3. "New Feature Scoring based on Statistical value and Document Frequencies" (Pramokchon and Piamsa-nga, 2014b) issue is based on the difference between the Z-score of the feature document frequency in one category and the other categories. For each category, a feature, whose document frequency in the category is higher

than that in the other categories, is highly informative since it is relevant with respect to this category.

4. “A statistics-based feature score for classifying class-imbalanced data” issue is based on statistical t-Test technique which evaluates the difference of mean of feature occurrence frequencies between one class and other classes as a usefulness measure of individual feature. Unlike other feature score, t-Test can be used to determine whether two class sizes are equal or not. It assumed that both distributions are normal and both variances are unequal. Moreover, its function normalizes the mean difference value by the common standard deviation of the two classes. Therefore, the t-Test score is insensitive to the problem of imbalanced class distribution.

5. “K-mean clustering based feature scoring method” (Pramokchon and Piamsa-nga, 2014c) issue is based on a technique to reduce the effect of imbalanced class distribution by creating intermediate subclasses by clustering features in each large class and then use these subclasses to be analyzed along with small main classes. Technically, large classes of training dataset are clustered by K-mean clustering into k subclasses and those k subclasses are used to be replaced with their main class for computing feature scores. Unlike most of the re-sampling paradigm (under-sampling, over-sampling, and hybrids), our proposed method avoids the problem of important concept missing, overfitting, and overgeneralization by considering both within-class and between-class relationships based on the clustering method.

6. “Hierarchical agglomerative clustering based feature scoring method” issue is introduced in order to enhance the idea of applying clustering as the data processing before classification, in this work, we propose method based on Hierarchical Agglomerative Clustering (HAC) (Hastie *et al.*, 2008;Hsiao and Chang, 2008;Manning *et al.*, 2008) integrated with class-sensitive weighting for determination cutoff for imbalanced text classification problems. The HAC method does not require a predefined number of clusters as input. For partitioning data into clusters, the method measures the dissimilarity between instances and only requires a cutoff threshold computed based on proposed class-sensitive weighting. The proposed method does not

require expertise knowledge or iterative empirical experiments to define the number of K clusters and easy for implementation; and its resulted clusters are deterministic.

MATERIALS AND METHODS

Materials

1. Computer Systems

1.1 Hardware High Performance Notebook 1 system

1.2 Hardware High Performance Server 1 system

1.3 Software
- MATLAB
- Perl
- EXCEL
- VBA
- PSPP
- WEKA

2. Dataset

There are some publicly available standard text document datasets that can be used for text categorization. In our experiment, we use the datasets as below:

2.1 Reuters 25178

The Reuters-21578 dataset is consisting of stories from Reuter's news agency which classified under categories related to economics. The dataset has been most widely used for the experimental works in Text Categorization so far. In Reuters-21578 dataset, we adopt the top 10-categories; 6343 documents in training set and 2548 document in test set. The distribution of the category is unbalance. The maximum category has 2787 documents, about 43.94% of training set. The minimum category has 79 documents, about 1.24% of training set. Table 1 presents the10 most frequent categories along with the number of training and test examples in each.

2.2 RCV1v2

The RCV1v2 dataset is used to evaluate our algorithm (Lewis *et al.*, 2004). Table 2 shows details of dataset characteristics.   It is organized in four "Topic Codes" hierarchical groups: CCAT, ECAT, GCAT, and MCAT. Each class also has different defined subclasses, which we used it as ground truth. Totally, dataset contains 19,806 training documents and 16,886 testing documents. Table 2 shows sizes of main classes and number of their ground truth subclasses. The largest class (CCAT) is almost five times as large as the smallest class (ECAT). Table 2 depicts distribution of ground truth subclasses. Note that size distributions of different subclasses are likely similar.

Table 1 Number of training/testing documents

Category	Number of Training Documents	Number of Testing Documents
Earn	2787	1084
Acquisition	1592	711
Money-fx	351	130
Trade	345	112
Crude	343	151
Interest	343	133
Ship	179	87
Corn	176	56
Wheat	148	49
Grain	79	35

Table 2 Dataset characteristics.

	Number of	Number of documents	
Class	**subclasses**	**Training**	**testing**
CCAT	18	8410	6993
ECAT	10	1690	1467
GCAT	23	4959	4580
MCAT	4	4747	3846

2.3 UCI repository dataset

Four benchmarks (UCI) labeled dataset (Bache and Lichman. 2013), consist of Isolet, Leukemia, Madelon, and Ovarian, were selected in experiments to test the performance of proposed algorithm. They are described in Table 3.We selected these datasets as they are either well understood in terms of feature relevance or they contain a huge number of features with comparatively much smaller sample size and are thus good candidates for feature selection.

Table 3 UCI Dataset characteristics.

Dataset	Number of classes	Number of documents	Number of features
Isolet	26	1559	617
Leukemia	2	73	7129
Madelon	2	2600	500
Ovarian	2	54	1536

Methods

1. Outlier based Feature Selection Algorithm

An outlier is defined as a data point that is very different from the rest of the data (Aggarwal and Yu, 2005). Such a data point often contains useful information on abnormal behavior in the system that is characterized by the data. The outlier identification problem is translated to a problem of identifying those data points that lie in a so-called outlier region.

Figure 2 shows 50 data points of 50 random samples that have a log-normal distribution (mean=1, SD=1). If the data points which are beyond a predetermined cutoff are considered outliers, three data points which have larger values than the upper cutoff are considered as outliers in the 50 random samples. If this data set is feature score, these three outliers are features that contain more useful information that contributes to the classification result.

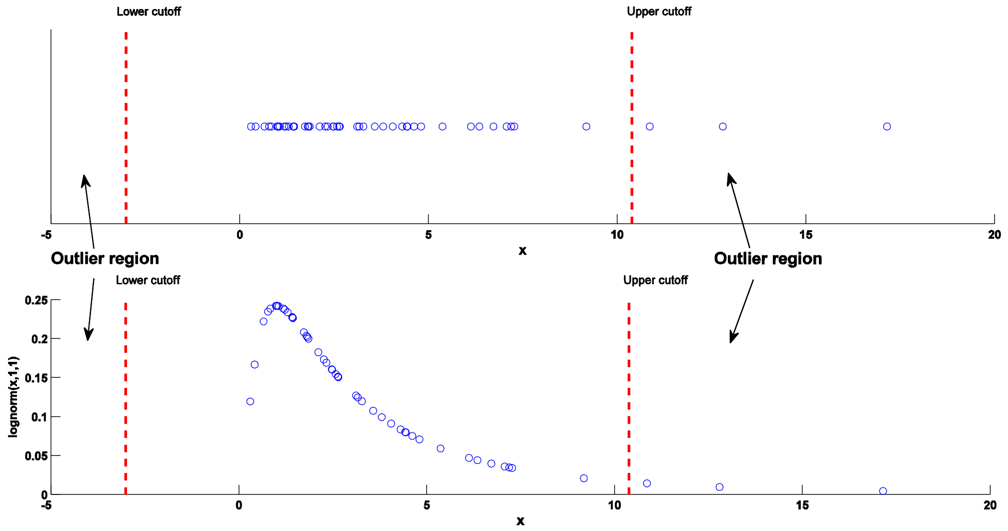

Figure 2 Outlier identification using outlier region

Our proposed method uses outliers to identify the relevant features; some outliers whose scores are higher than other features. These features are selected for inclusion in the feature subset. Under this definition the cut-off threshold is formulated to determine the outlier region using the concept of univariate outlier identification. For all features f_i in the original feature set T, $\boldsymbol{x(f_i)}$ denotes the feature score (such as DF, IG or CHI) as presented in Section 5.1.

Definition 6 Outlier region. For any confidence coefficient, given a significance level α, $0 < \alpha < 1$, the α–outlier region of the distribution is

$$out(\alpha, \mu, \sigma^2) = \left\{x(f_i)\colon |x(f_i) - \mu| > z_{1-\alpha/2} \cdot \sigma\right\},$$

where $z_{1-\alpha/2}$ is the $1 - {}^{\alpha}/_{2}$ quantize of the $N(0,1)$, μ is the average and σ^2 is the variance of the features $(\boldsymbol{f_i}), f_i \in T$.

Definition 7 Outlier. A feature f_i is an α–outlier if and only if $x(f_i) \in out(\alpha, \mu, \sigma^2)$.

Based on above definition of α–outlier, relevant features are defined as follows:

Definition 8 Relevant Feature. A feature t_i is a relevant feature if and only if f_i is an α–outlier and $x(f_i) > \mu$. Hence, t_i is included in the optimal feature subset (Opt).

We now describe a feature selection algorithm. There are two steps shown in Algorithm2. First, feature score between each feature and the target categories is calculated (lines 1-2). In the selection step (lines 3-9), we deal with the computation of score of features. To obtain the outlier region, we use a predefined confidence coefficient as a parameter for computing the outlier cut-offs. We scan each feature in T, the relevant features as in definition 3 then are selected as the optimal feature set.

```
Algorithm 2  Relevant feature selection algorithm
Input : Original feature set (T), Predefined significant level(α)
Output :    Optimal feature subset ($Opt$)
Compute score step:
    for each feature $t_i$ in T do
        Compute a $x(t_i)$ for $t_i$;
Selection step:
    compute $\alpha$–outlier region;
    $Opt \leftarrow \emptyset$
    for each feature $t_i$ in T do
        if ($t_i$ is accepted in Definition 3) then {
            $Opt \leftarrow Opt \cup \{t_i\}$;
        }
    return $Opt$;
```

Our method selects many features that are effective for text categorization and simply reduces the size of the original features. The outlier feature statistics selection chooses features from a pool of original terms T which requires complexity of only $O(T)$ to scan each feature in T. Thus, the selection processing time varies linearly in the number of original features. This indicates that our selection approach requires lower time complexity than traditional selection methods. Moreover, this approach can greatly reduce the feature vector size without repeated experimental investigations to find feature candidates. Note that, to determine the number of the best features, the wrapper method requires iterations of performance evaluations where the cost varies according to the complexity of the classifier.

2. Feature Scoring based on Statistical value and Document Frequencies

In text classification, most scoring functions for feature selection algorithm are based on frequencies of feature occurrences in the training set. The basic notion that a feature whose occurrence frequency in a document of a specific category (positive category c_k) is higher than that of the other category (negative category $\overline{c_k}$) is desirable since it contains higher information and helps distinguishing between the two categories. However, comparing only the occurrence frequencies of one category and other categories will be limited since the size of the training set in both categories are in fact highly unequal. In ref. (Makrehchi and Kamel, 2011) argue that some feature scores may fail because they trend to rank features in the larger class, while those features in smaller class are difficult to be selected.

This leads us to propose a new feature selection score called “document frequency difference” (DFD), which is based on the difference between standard (Z) values of the document frequencies of a feature for both positive and negative categories. The standard score indicates how many standard deviations an observation is above or below the mean; therefore, we can compare document frequency of a feature even if both category sizes are drastically unequal. Moreover, this score is the family of Document Frequency that are very popular methods because of their simplicity and also acceptable in the Information Retrieval community (Makrehchi and Kamel, 2011).

We first introduce the notation of standard document frequency.

Definition 9 (Standard Document Frequency). The standard document frequency of f_i in c_k is given by

$$Z_{df}(f_i, c_k) = \frac{DF(f_i, c_k) - \overline{DF(f_\iota, c_k)}}{S_{df}(F, c_k)},$$

and the standard document frequency of f_i in $\bar{c}_k$ is

$$Z_{df}(f_i, c'_k) = \frac{DF(f_i, \mathrm{c}'_{\mathrm{k}}) - \overline{DF(f_\iota, \mathrm{c}'_{\mathrm{k}})}}{S_{df}(F, \mathrm{c}'_{\mathrm{k}})},$$

where $DF(f_i, c_k)$ and $DF(f_i, \mathrm{c}'_{\mathrm{k}})$ are document frequencies of a feature f_i in c_k and c'_{k} respectively; $\overline{DF(f_\iota, c_k)}$ and $\overline{DF(f_\iota, \mathrm{c}'_{\mathrm{k}})}$ are means of document frequencies for c_k and in c'_{k}, respectively, and $S_{df}(F, c_k)$ and $S_{df}(F, \mathrm{c}'_{\mathrm{k}})$ are standard deviations of the samples of document frequencies for c_k and c'_{k}.

Note that $\overline{DF(f_i, c_k)} = \frac{1}{|F|}\sum_{i=1}^{|F|} DF(f_i, c_k)$; $\overline{DF(f_i, \bar{c}_k)} = \frac{1}{|F|}\sum_{i=1}^{|F|} DF(f_i, \mathrm{c}'_\mathrm{k})$; $S_{df}(F, c_k) = \sqrt{\frac{1}{|F|}\sum_{i=1}^{|F|}\left(DF(f_i, c_k) - \overline{DF(f_i, c_k)}\,\right)^2}$; and $S_{df}(F, \mathrm{c}'_\mathrm{k}) = \sqrt{\frac{1}{|F|}\sum_{i=1}^{|F|}\left(DF(f_i, \mathrm{c}'_\mathrm{k}) - \overline{DF(f_i, \mathrm{c}'_\mathrm{k})}\,\right)^2}$.

We can then define a score for a feature by calculating the difference between its two standard document frequencies.

Definition 10 (Local Document Frequency Difference). For a specific category c_k, the document frequency difference of feature f_i is given by $DFD(\, f_i, c_k) = Z_{df}(f_i, c_k) - Z_{df}(f_i, \mathrm{c}'_\mathrm{k})$.

Intuitively the document frequency difference measures the importance of a feature for one category against others. By avoiding bias from imbalanced class distribution, document frequency difference tends to favor features whose document frequencies for one category are much higher than those for other categories (In this case, positive DFD value indicates relevance for the category; and negative DFD indicates irrelevance for the category). If a feature is relevant for both categories, its DFD score is small and close to zero.

Like CHI, DFD is specified with respect to a specific category c_k; in order to assess the value of a feature t_i in a global category-independent sense, we calculate DFD as follows:

Definition 11 (Global Document Frequency Difference – DFD). For a set of category of interest C, the document frequency difference of feature f_i is given by $DFD(f_i) = max_{i=1}^{|C|}\{DFD(f_i, c_k)\}$

We compared the computational complexity of the proposed scoring function with the standard metrics DF, IG, and CHI. The time complexity of our scoring function is of the same order as standard metrics which are linear in the size of original feature set T.

3. A statistics-based feature score for classifying class-imbalanced data

A feature is representative for a specific class if number of documents from that class contain it more than one from other classes. If we can evaluate the difference of occurrence frequency of features between two a specific class and other classes, we are able to use this difference as the feature score.

The statistical t-test technique is a commonly used method for statistical evaluation of the difference between two sample means (Tamhane and Dunlop, 2000). Given two datasets, t-test can be used to determine whether both dataset sizes are drastically unequal through analysis means, standard deviations and the assumption that both distributions are normal and both variances are unequal.

As we mentioned in previous section, the occurrence frequencies of features in a document are represented by the TF-IDF weight. Therefore, we applied the t-test technique to determine the significant difference of mean of the TF-IDF weight of a feature between a specific class and other classes. The basic idea is that a feature, whose mean of TF-IDF weight among the documents in a specific class is significantly higher than that of other class, is a highly discriminative feature since it contains higher information about a specific class.

This is a proposed t-test score which is defined as follows:

$$tscore(f_i, c_k) = \frac{\overline{w}(f_i, c_k) - \overline{w}(f_i, c_k')}{S}$$

where $\overline{w}(f_i, c_k)$ is the sample mean of TF-IDF weight of feature f_i of document in a specific class $c_k \in C$, $\overline{w}(f_i, c_k')$ is the sample mean of TF-IDF weight of feature f_i of document in the other class, $c_k' = C - c_k$. S is an estimator of the common standard deviation of the two classes which can be calculated as follows:

$$S = \sqrt{\frac{S^2(f_i, c_k)}{N_{c_k}} + \frac{S^2(f_i, c'_k)}{N_{c'_k}}}$$

$S^2(f_i, c_k)$ is the standard deviation of TF-IDF weight of f_i in a specific class c_k and $S^2(f_i, c'_k)$ is the standard deviation of weight of f_i in the other class.

If a feature f_i has high t-test score, there is a statistically significant difference in the occurrence frequency in a specific class c_k when it is compared with that in other classes. Therefore, this feature has higher discriminating power.

For multi-class domain, finding some highly-relevant features of each class and combining them as an optimal feature set can decrease the destructive impact of imbalance class distribution (Makrehchi and Kamel, 2011). The t-test score is locally specified with respect to a specific class c_k. In order to globally assess the value of a feature f_i in the multi-class domain, we calculate three t-test scores in three alternate ways to combine the class-specific score defined as follow:

$$tscore_{max}(f_i) = max_{k=1}^{|C|} tscore(f_i, c_k)$$

$$tscore_{avg}(f_i) = \frac{\sum_{k=1}^{|C|} tscore(f_i, c_k)}{|C|}$$

$$tscore_{wavg}(f_i) = \sum_{k=1}^{|C|} P(c_k) tscore(ft_i, c_k) \square$$

The Max-tscore, $tscore_{max}(f_i)$, aims to assess the maximum significance of feature occurring in one class against other classes. The Weighted Averaged-tscore, $tscore_{wavg}$, is basically average of F1 of each class where its weight varies to its size. On the other hand, the Averaged-tscore, $tscore_{avg}$, uses equal weights of all classes, regardless of how many documents belong to it.

4. K-mean clustering based feature scoring method

4.1 K-mean clustering

The K-means clustering is an automatic and unsupervised approach to separate a dataset into k clusters. A data entity belongs to a cluster if that entity is the nearest to that cluster centroid. Given a set of document data $(d_1, \dots, d_{|D|})$, where each data is a vector, K-means clustering aims to partition the |D| observations into k clusters (k ≤ |D|) $S = \{s_1, \dots, s_k\}$ so as to minimize the within-cluster sum of squares (WCSS): $arg\ min_S \sum_{j=1}^{k} \sum_{x_i \in s_j} \left\| x_i - \mu_j \right\|^2$,where μ_j is the arithmetic mean of the data in S_j. The most common algorithm uses an iterative refinement technique (MacQueen, 1967). The process of algorithm starts by selecting k initial cluster centers. Then, the centers, called the centroid, are iteratively updated by finding the means of the points closest to each center. The process is done when there is no more updating needed on the centroids. In order to perform the K-mean clustering, the distance between the documents and the centroids should be calculated. Many research projects address some similarity/correlation/redundancy measures that have been used for feature selection, such as correlation coefficient (Bache and Lichman, 2013;Ferreira and Figueiredo, 2012;Yu and Liu, 2004), symmetrical uncertainty (Ferreira and Figueiredo, 2012;Yu and Liu, 2004) and absolute cosine (Ferreira and Figueiredo, 2012). In (Ferreira and Figueiredo, 2012) argue that using angles to measure similarity is better suitable for high-dimensional sparse data. Therefore, absolute cosine is used as geometric view in our proposed algorithm.

$$cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \cdot \|d_j\|}$$

High cosine value shows that two documents are likely similar because they share most of their words. The cosine value is equal to 1 when two documents are identical, and the value 0 means that the documents are totally dissimilar and have nothing in common.

4.2 K-mean clustering based feature selection method

We proposed to separate data in large classes into several subclasses according to their proximity, all the subclasses, then, are used for feature scoring measure instead of the main classes. The proposed algorithm is showed in Algorithm 3 (Cluster-based feature scoring measure algorithm).

Algorithm 3 Cluster-based feature scoring measure algorithm

```
Input:  Original feature set ($F$)
              Training data ( $\mathrm{Tr} = \{\langle \mathrm{d}_1, \mathrm{c}_\mathrm{i} \rangle, \ldots, \langle \mathrm{d}_{|\mathrm{Tr}|}, \mathrm{c}_{|\mathrm{Tr}|} \rangle\}$)
              Main class $(\mathrm{C} = \{\mathrm{c}_1, \ldots, \mathrm{c}_{|\mathrm{C}|}\})$
              Predefined number of cluster $(\mathrm{K} = \{\mathrm{k}_1, \ldots, \mathrm{k}_{|\mathrm{C}|}\})$
Output:       Feature set ranked by score ($FScore$)
1:      for all classes in C do
2:            cluster all data $\{\langle \mathrm{d}, \mathrm{c}_\mathrm{j} \rangle\}$ within the class cj into
                    $\{\langle \mathrm{d}, \mathrm{c}_\mathrm{j}^\mathrm{t} \rangle\}\ t = 1, \ldots, k_j$ using K-Means clustering
3:      $\mathrm{Tr}' = \{\langle \mathrm{d}_1, \mathrm{c}_1^\mathrm{t} \rangle, \ldots, \langle \mathrm{d}_{|\mathrm{Tr}|}, \mathrm{c}_{|\mathrm{Tr}|}^\mathrm{t} \rangle\}$          //new training set
4:      for each feature $f_i$ in $F$ do
5:            $s_i \leftarrow ComputeScore(f_i, Tr')$
6:      $FScore \leftarrow SortedFeatureByScore(s_i)$
7:      Return FScore
```

In Algorithm 3, given a set of training data with known main classes, $Tr = \{\langle d_i, c_j \rangle\}\ i = 1, \ldots, |Tr|, j = 1, \ldots, |C|$, the proposed method starts by separately clustering each class cj into kj clusters using the K-means clustering algorithm (lines 1-2). Then, in line 3, we have a new training set $Tr' = \{\langle d_i, c_j^t \rangle\}\ , i = 1, \ldots, |Tr|, \mathrm{j} = 1, \ldots, |\mathrm{C}|, t = 1, \ldots, |K|$ where t is the label of a specific cluster (subclass) of main class C and $|K| = \sum_{j=1}^{|C|} k_j$. Then in lines 4-5, feature score s_i of Tr' is measured with associated subclasses instead of the main classes. Feature results are sorted by feature score (line 6) and then feature set which is ranked by feature score is returned in line 7.

Separating the main classes into the subclass before computing feature score not only reduces the influence of large classes but also increases chance that important but rare features of small classes can be selected. In order to avoid the impact of class separating with small classes, we assign a small kj for small classes and large kj for large classes. The kj value is obtained by following formula:

$$k_j = round\left(\frac{|Tr|_j}{|Tr|} \times |K|\right)$$

where |K| is an empirical parameter of the proposed algorithm, which requires the experiment to find out the best value, represented the summation of resulted subclass and |Tr|j is a number of text document in the class cj.

5. Hierarchical agglomerative clustering based feature scoring method

We use the hierarchical agglomerative clustering to separate data from both larger and smaller class into several subclasses which are used instead of the main classes for feature scoring and the ideal is demonstrated in Figure 3.

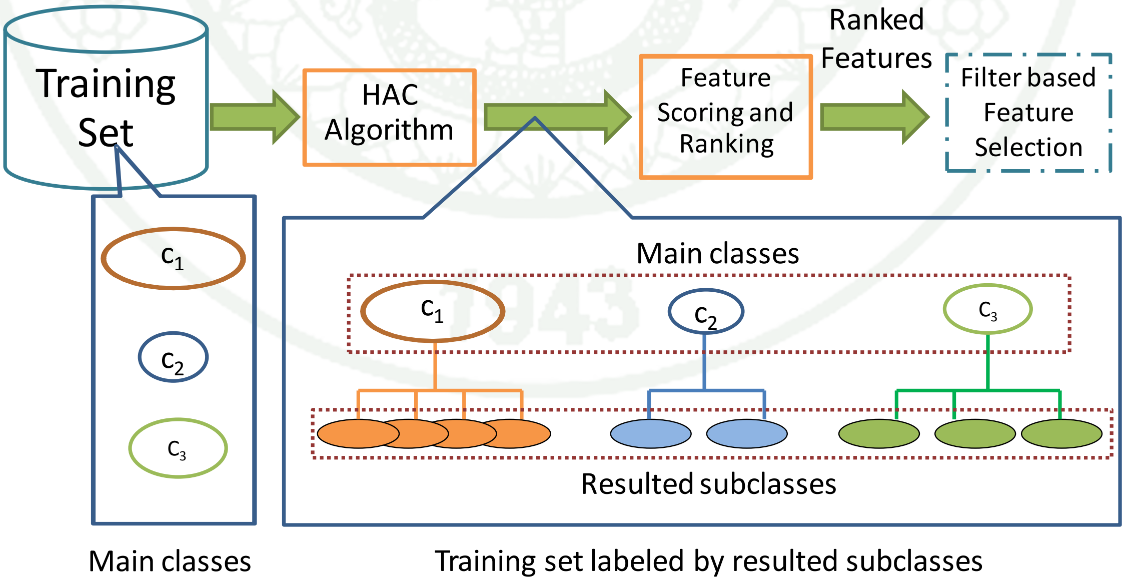

Figure 3 Concept of Hierarchical Agglomerative Clustering based feature scoring method.

5.1 Hierarchical Agglomerative Clustering (HAC)

Text clustering is about to combine similar text documents into a group (cluster). Some classes of text are large enough to be further clustered and then all data can be organized as hierarchical cluster. Hierarchical Agglomerative (bottom-up) clustering (Hastie *et al.*, 2008;Manning *et al.*, 2008) is about to consider each document as a singleton cluster at the beginning and then successively merge (or agglomerate) pairs of clusters until all documents have been merged into a single binary tree.

An example of document clustering is visualized by a dendrogram shown in Figure 4. In the dendrogram, element on the x-axis is document and distance marks that two elements (either document or cluster) are merged in on the y-axis. Thus, a binary tree can be plotted so that the height of each node is proportional to the values of the intergroup distances between its two child nodes. Possible clustering results depend on cut-off distance. In Figure 4., there are three examples of clustering: at height 0.84 into 3 clusters, at 0.72 into 4 clusters, and at 0.61 into 6 clusters. The higher cut-off value, the smaller number of clusters is.

In order to determine the best cut-offs, a measure of distance between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of documents), and a linkage criterion which specifies the distance of sets as a function of the pair-wised distance between two clusters. Many research projects argue that using angles to measure similarity is more suitable for document clustering which has highly-dimensional sparse data (Ferreira and Figueiredo, 2012). Therefore, absolute cosine is used as geometric view in our study.

HAC merges pairs of elements that are in close proximity into binary clusters using the linkage function. There are tree popular linkage criterion (Hastie *et al.*, 2008;Manning *et al.*, 2008), such as the single-link, complete-link, and UPGMA schemes. However, we use the complete-link clustering in our work because the approach can be more efficient than the other approaches for large dimensional data

(Hsiao and Chang, 2008). The overall complexity of the algorithm is $O(N^2 \log N)$ which N is the number of document in the dataset.

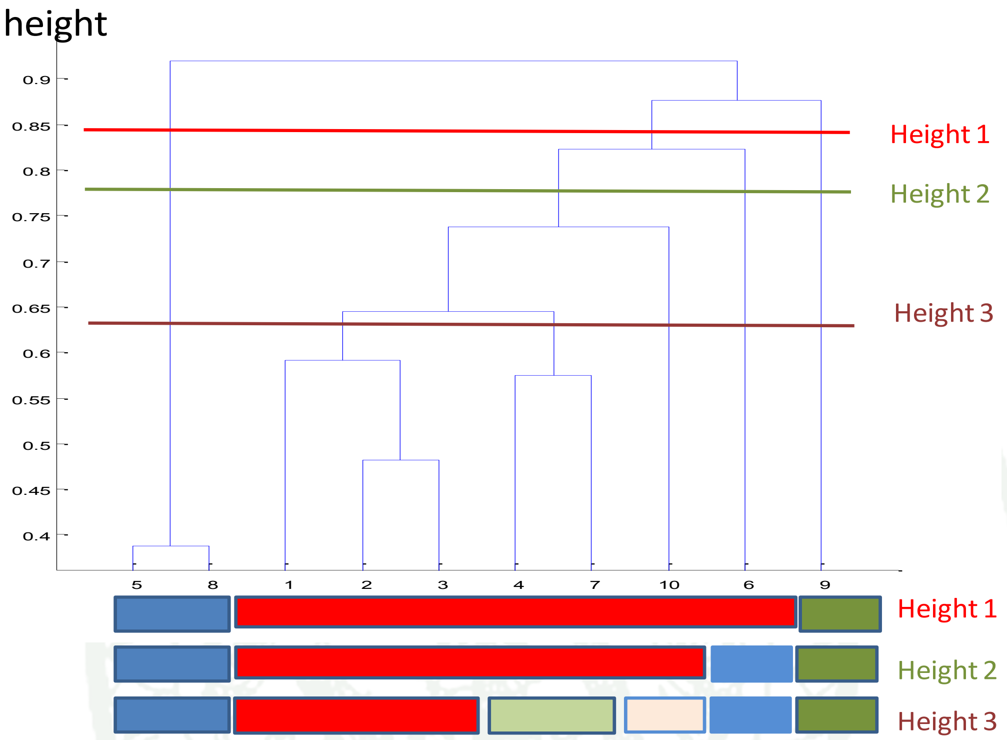

Figure 4 An example of text clustering visualized by dendrogram

The HAC algorithm is inferior in term of time complexity compared to the other clustering algorithms; however, it is deterministic algorithm which does not require a predefined number of clusters. The HAC algorithm is processed in one time while the *K*-mean clustering has to empirically repeat until the best number of clusters is reached, that is more exhaustive task.

5.2 Feature Scoring Method

The proposed algorithm is shown in Algorithm 1 (Cluster-based feature scoring measure algorithm). In Algorithm 1, given a set of training data with known main classes, $\mathrm{Tr} = \{\langle d_i, c_j \rangle\}\ i = 1, \dots, |\mathrm{Tr}|, j = 1, \dots, |C|$, the proposed method starts by separately clustering each class c_j into k_j clusters using the Hierarchical Agglomerative Clustering algorithm (lines 1-2). Then, in line 3, resulting data become $\mathrm{Tr}' = \{\langle \mathrm{d_i}, \mathrm{c_j^t} \rangle\}, \mathrm{i} = 1, \ldots, |\mathrm{Tr}|, \mathrm{j} = 1, \ldots, |\mathrm{C}|, \mathrm{t} = 1, \ldots, |\mathrm{K}|$ and $|\mathrm{K}| = \sum_{\mathrm{j}=1}^{|\mathrm{C}|} \mathrm{k_j}$ where t is the label of a specific cluster (subclass) of main class *C*. Then, we have new training sets Tr', which are passed as input to measure the feature score with considering subclasses instead of the main classes (line 4-5): after looping for computing scores of each feature, then used it for ranking all features, decreasingly (line 6). The result, which is a ranked feature by its scoring, is returned (line 7).

```
Algorithm 4 HAC-based feature scoring measure Algorithm
Input:  Original feature set ($F$)
            Training data ($Tr = \{\langle d_1, c_{|j|} \rangle, \ldots, \langle d_{|Tr|}, c_{|Tr|} \rangle\}$)
            Main class ($\mathrm{C} = \{\mathrm{c_1}, \ldots, \mathrm{c_{|C|}}\}$)
            Predefined number of cluster ($\mathrm{K} = \{\mathrm{k_1}, \ldots, \mathrm{k_{|C|}}\}$)
Output:     Feature set ranked by score ($FScore$)
1:      for all classes in C do
2:          cluster all data $\{\langle \mathrm{d}, \mathrm{c_j} \rangle\}$ within the class cj into
                $\{\langle \mathrm{d}, \mathrm{c_j^t} \rangle\}$ $t = 1, \ldots, k_j$ using HAC using proposed cutoff
3:      $\mathrm{Tr}' = \{\langle \mathrm{d_1}, \mathrm{c_1^t} \rangle, \ldots, \langle \mathrm{d_{|Tr|}}, \mathrm{c_{|Tr|}^t} \rangle\}$          //new training set
4:      for each feature $f_i$ in $F$ do
5:          $s_i \leftarrow ComputeScore(t_i, Tr')$
6:      $FScore \leftarrow RankdFeatureByScore(s_i)$
7:      Return FScore
```

Proposed method considers relevance between each feature and each class in the whole dataset by using the resulted subclasses to re-labeling each document without data over- or under-sampling process. Therefore, unlike the re-sampling paradigm, this method does not face problems of important concept missing, overfitting, and overgeneralization etc.

5.3 Class-sensitive weighting for cutoff threshold

After creating the hierarchy, we need a cutting point to partition data into clusters. We propose a method for determining an appropriate clustering cut-off based on normalized height of nodes in the cluster tree. It is about to compare the height of each node with the heights of neighboring nodes below it in the tree. A node that is approximately the same height as the nodes below it indicates that there is no distinct between objects joined at this level of the hierarchy. These nodes are to exhibit a high level of consistency, because the distance between the objects being joined is approximately the same as the distances between the objects they contain. On the other hand, a node whose height differs noticeably from the height of the nodes below it indicates that the objects joined at this level in the cluster tree are much farther apart from each other than their components were when they were joined. This node is said to be inconsistency with the nodes below it, except the leaf node's inconsistency which is zero. The relative consistency of each node in the hierarchical cluster tree can be quantified and expressed as the inconsistency coefficient. This value compares the height of a node in a cluster hierarchy with the average height of nodes below it.

The inconsistent coefficient of each cluster node $n_i$in the hierarchy is given by

$$\mathrm{I}(\mathrm{n_i}) = \frac{\mathrm{h}(\mathrm{n_i}) - \mu\big(\mathrm{h}(\mathrm{n_i})\big)}{\sigma\big(\mathrm{h}(\mathrm{n_i})\big)}$$

where $\mathrm{h}(\mathrm{n_i})$ is height of the node $\mathrm{n_i}$, $\mu\big(\mathrm{h}(\mathrm{n_i})\big)$ is mean of the heights of all the nodes included in the calculation ($\mathrm{n_i}$ and other node below it), and $\sigma\big(\mathrm{h}(\mathrm{n_i})\big)$ is the standard deviation of the height of all the node included in the calculation. Nodes, which join different clusters, have a high inconsistency coefficient; and nodes, which join same clusters, have a low inconsistency coefficient.

Next, we proposed the computing cut-off threshold, $\mathrm{T_{cut}}$, based on the statistical method of outlier detection defined as

$$T_{cut} = \mu_I + \sigma_I \cdot \left(Z_{1-\frac{\alpha}{2}} + w_{c_j}\right)$$

where α is a confidence coefficient which is only one statistical parameter in the proposed method, $Z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantize of the $N(0,1)$, μ_I and σ_I are mean and standard deviation of all cluster node in hierarchy, respectively and w_{c_j} is the class-sensitive weight of each main class (described below). Outlier detection is a statistical approach that indicates the observations which have very different behavior from others in data collection (Hodge and Austin, 2004). The cluster node n_i, that has inconsistency coefficient value $I(n_i)$ more than T_{cut}, this node is inconsistent when compared with its child nodes in the hierarchy, then its child nodes are grouped in the same cluster. The lower cut-off value means that the class will be divided into many subclasses while the higher cut-off value will make the clustering divides the class into the less number of subclasses.

Separating the main classes into the subclass before computing feature score not only reduces the influence of large classes but also increases chance that important but rare features of small classes can be selected. In order to avoid the impact of class separating with small classes, we assign a more w_{c_j} value for small classes and small w_{c_j} value for large classes. The w_{c_j} value of main class,$c_j \in C$, is obtained by following formula:

$$w_{c_j} = \frac{1}{\left(|c_j|/min\{|c_j||c_j \in C\}\right)^{1/2}}$$

where $|c_j|$ is a number of text document in the class c_j, $min\{|c_j||c_j \in C\}$ represented the size of the smallest class in training dataset. Actually, w_{c_j} value of the smallest class is 1, the values of the other classes are descending with respect to their size and the weight of largest classes is the lowest valu

6.   Unsupervised Fast correlation-based feature selection for clustering

The proposed method composed of two approaches: measuring feature relevance by feature score to keep highly relevance features; and measuring feature redundancy by feature correlation to identify redundant features. Then, we define a policy to eliminate less important features. The algorithm is listed in Algorithm 5 (Unsupervised fast correlation-based filter algorithm.). In Algorithm 5, scores of each feature are computed and then used it for sorting (lines 2 and 3); each feature will be compared with others in order to find relevancy (lines 4-12); feature redundancy is measured by computing feature similarity (Ferreira and Figueiredo, 2012) (line 8); each feature is then considered to be removed from the output (lines 13-17); and after looping for every feature, the result, which is a feature set that each has high relevance and low similarity among themselves, is returned (line 20),

```
Algorithm 5  Unsupervised Fast Correlation-Based Filter Algorithm
Input :  Original feature set ($F$)
         Training data ( $D$)
         Threshold parameter (α)
Output : Optimal feature subset ($Opt$)
1:   for each feature fi in $F$ do    // Compute score all feature
2:       $s(f_i) \leftarrow Score(f_i, D)$
3:   $ST \leftarrow SortedFeatureByScoreDes(s(f_i), F)$  // set of sorting feature in F by score descending
4:   $f_j \leftarrow GetFirstElement(ST)$
5:   do begin
6:       $F' \leftarrow GetAllNextFeature(f_j, ST)$
7:       for each feature fi in F'
8:           $sim(f_i, f_j) \leftarrow ComputeSimilarity(f_i, f_j)$
9:       $thres_{redundant}(f_j) \leftarrow ComputeRedudantThreshold(\alpha)$
10:      $thres_{remove}(f_j) \leftarrow ComputeRemoveThreshold(\alpha)$
11:      $ST' \leftarrow SortFeatureByScoreAs(s(f_i), F')$ //set of sorting feature in F' by score ascending
12:      $cr \leftarrow CumulativeRelevance(ST')$
13:      for each feature fi in ST'                // redundant identify and remove
14:          if ($sim(f_i, f_j) > thres_{redundant}$)
15:              if ($cr - s(f_i) > thres_{remove}$)
```

```
                remove fi from ST
        $f_j \leftarrow GetNextElement(f_j, ST)$
    end until (fj = NULL)
    Opt = ST
    return $Opt$
```

In practical, many feature selection methods suffer the problem of selecting appropriate thresholds for both redundant feature identification and redundancy elimination. Thus, we proposed the statistics based method to compute threshold to identify redundant feature of feature f_j, is defined as

$$Thres_{redundant} = \overline{sim(f_j)} + Z_{1-\frac{\alpha}{2}} \cdot \sigma_{f_j}$$

where α is a confidence coefficient, $Z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantize of the $N(0,1)$, $\overline{sim(f_j)}$ and $\sigma_{f_i}$are average and standard deviation, respectively, of similarity between features f_i and f_j, $f_i \in ST'$. The feature f_i, that has similarity value $\left(sim(f_i, f_j)\right)$ more than $\mathrm{thres}_{\mathrm{redundant}}$, is identified as redundant feature of feature f_j. Next, we introduce a criterion in strategy for redundant feature elimination. The decision to remove a feature depends on a cumulative relevance (cr) measure (Ferreira and Figueiredo, 2012).

$$cr = \sum_{i=1}^{|ST'|} s(f_i)$$

The cr value is used to calculate summation of relevance score of feature in subset ST'. Then, we propose removing the features $f_i$where $cr - s(f_i)$ is more than threshold, $Thres_{remove} = \left(1-\frac{\alpha}{2}\right) \cdot cr$. This means that f_i is redundant and removing f_i affects cr value a little, thus f_i can be removed from the feature set. On the other hand, some features f_i have $s(f_i)$ that make $cr - s(f_i) < thres_{remove}$, it means that the feature is redundant but it is influent to the cumulative relevance of feature set; thus, it should be kept in the selected feature set. Finally, we can select highly-relevant features and remove highly-redundant feature for data clustering.

RESULTS AND DISSICUSSION

Results

1. Evaluation Method

To measure the performance of a feature selection algorithm, the results obtained must be represented in some measure to provide means for comparison with other researches. Usually precision, recall and F1-measure are used to evaluate the feature selection algorithms. They are showed in the following equations:

Precision, $P_k = \frac{TP_k}{TP_k + FP_k}$

Recall, $R_k = \frac{TP_k}{TP_k + FN_k}$

We also use F1 ($F1_j$) as the local performance measure, which is a combination of precision P_j and recall R_j.

F1-measure, $F1_k = \frac{2 \times P_k \times R_k}{(P_k + R_k)}$

To exhibit the performance of the proposed method on the smaller class, the F1 measures over the multiple classes are summarized using the global average measure defined as follows:

Micro-average Precision, $Micro - P = \frac{\sum_{k=1}^{|C|} TP_k}{\sum_{k=1}^{|C|} (TP_k + FP_k)}$

Micro-average Recall, $Micro - R = \frac{\sum_{k=1}^{|C|} TP_k}{\sum_{k=1}^{|C|} (TP_k + FN_k)}$

Macro-average Precision, $Macro - P = \frac{\sum_{k=1}^{|C|} P_k}{|C|}$

Macro-average Recall, $Macro - R = \frac{\sum_{k=1}^{|C|} R_k}{|C|}$

Micro-average F1-measure, $Micro - F1 = \frac{2 \times micro - P \times micro - R}{(micro - P + micro - R)}$

Macro-average F1-measure, $Macro - F1 = \frac{\sum_{k=1}^{|C|} F1_k}{|C|}$

2. Outlier based Feature Selection Algorithm with DFD Feature Scoring based on Statistical value and Document Frequencies

The proposed selection algorithm is evaluated by comparing the effectiveness of the selected feature subset with ranking wrapper-based feature selection in Algorithm 1 as the baseline. We applied four feature scoring schemes, CHI, IG, Document Frequency (DF) and the proposed DFD, to determine feature subset on training documents with different cut-off number of features ranging from 100 to 2000. At each cut-off number, the performance of the classifier is estimated by four-fold cross-validation. To eliminate random variation, the micro-average F1 was averaged over the five runs. We use Student's paired two-tailed t-test in order to evaluate the statistical significance (at the significant level 0.05) of the difference between the previous best subset and the candidate subset.

Figure 5 shows the micro-averaged F1, also averaged over five runs for each feature score as we change the cut-off number of features. The baseline algorithm process is repeated until the stopping criterion (number of feature = 2000) is satisfied, the results show that the best number of selected features that suited the classifier for CHI, IG, DF and DFD scores are 600, 1300, 1400 and 1200, respectively. With these results, we also confirm that the selected subset of features using the ranking wrapper-based algorithm respect to not only type of learning machine but also type of feature scores.

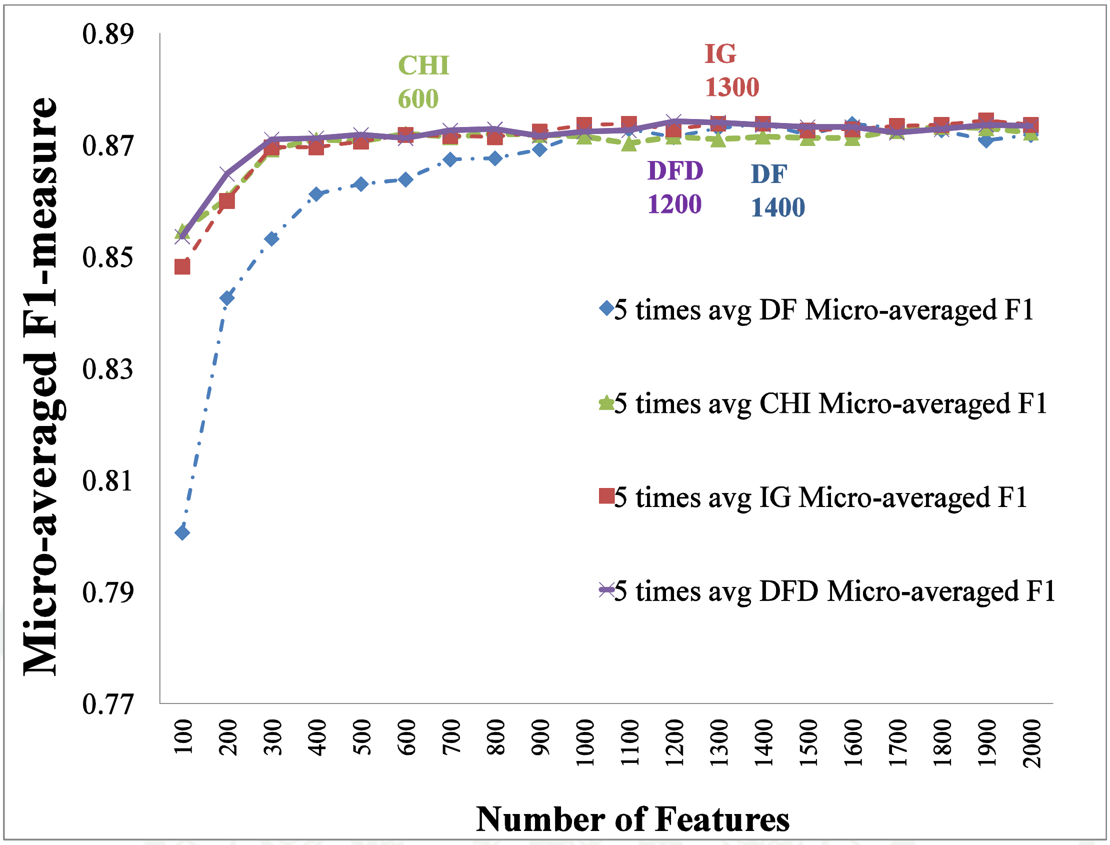

Figure 5 Micro-averaged F1 for each feature scores, as ranking wrapper algorithm varies the number of features

We investigated the effectiveness of the selected feature subset for multi-class text categorization in order to compare the performance of the four feature scores. The selected feature subsets from each score are used to train the classifier; then the classification performance is evaluated on testing documents. The 10-run micro-averaged and macro-averaged F1 are used for comparing the performances for different feature scores.

We compute Student's independent two-tailed t-test in order to evaluate the statistical significance of the difference between the two averaged F1 values: the one from the DFD score and the one from CHI, IG, and DF score. The results are shown in Table 4. The p-Val is the probability associated with an independent two-tailed t-Test. The symbols "+" and "-" identify statistically significant (at the 0.05 level) wins and losses of the proposed method over the baseline methods and no symbol means no statistically significant difference. We can see that IG obtained higher micro-averaged F1 than other scores (CHI, DF) due to the influence of the large-size categories because IG often assigns higher scores to the terms correlated with large classes such that small classes are not selected in top rank features (Makrehchi, M. and M. S. Kamel, 2011). However, the lowest macro-averaged F1 value shows that IG failed when it is applied to multiple classes with highly imbalance category sizes. We notice that DFD obtained micro-averaged F1 which is the largest among these four feature scores in the classifier and its macro-averaged F1 is higher than IG as well. As we can see in Table 4, the results obtained from the DFD are statistically better than from the IG. For micro-averaged F1, DFD is better than CHI and DF. However, for macro-averaged F1, DFD is statistically similar to DF and slightly smaller than CHI.

Table 4 Comparing Micro-average F1 and Macro-average F1 of selected features using DFD score and three standard feature scores

Average F1-measure	**DFD (1200f)**	**CHI (600f)**		**IG (1300f)**		**DF (1400f)**	
		measure	**p-Val**	**Measure**	**p-Val**	**measure**	**p-Val**
Micro	87.88	86.42 (W)	0.00+	87.72 (W)	0.00+	87.24 (W)	0.02+
Macro	78.95	79.18 (L)	0.02-	78.62 (W)	0.00+	79.07 (D)	0.23

We then used the proposed outlier-based feature selection in Algorithm 2 to select a feature subset with these four feature scores. Unlike the baseline algorithm, the outlier-based algorithm is non-iterative process and we set merely a parameter, value of the significance level $\alpha = 0.05$ with confidence level of 95%, to identify the optimal cut-off. Numbers of features selected by the outlier-based algorithm for CHI, IG, DF, and DFD are 314, 472, 514, and 358, respectively. The result shows that the outlier-based algorithm selects a smaller feature subset than the feature subset selected using the ranking wrapper-based algorithm.

Although the size of outlier-based feature set is smaller, we further investigate the effectiveness of those selected feature subsets for multi-class text categorization.

The selected feature subsets are used to train the classifier; then the classification performance is evaluated on testing documents. To compare the algorithms, the F1-measure is used for comparing the performances for different feature scores. Table 5 shows the F1-measures of features selected by each algorithm on the top-10 categories for CHI, IG, DF, and DFD. The F1 values in each table are 10-run averages of F1 values. We also compute Student's independent two-tailed t-test in order to evaluate the statistical significance of the difference between the two averaged F1 values: the one from the outlier-based method and the one from ranking wrapper-based method.

The last row in each table summarizes over all categories the wins/losses/ties in F1-measure (at significance level 0.05) comparing baseline feature subsets with the subset selected by the outlier-based method. Table 5 shows that the outlier-based method with CHI has a higher number of wins than the baseline, especially in small-size categories. The number of wins from both proposed algorithm and the baseline algorithm with IG are similar in large, moderate, and small size categories. The outlier algorithm with DF clearly outperforms the baseline algorithm on top-10 categories. Finally, the proposed algorithm with DFD achieves similar results in large size categories as the baseline algorithm and higher F1 measures in moderate size categories involve smallest size category, grain.

Table 6 compares the micro-averaged F1 and macro-averaged F1 on top-10 categories for four score schemes between proposed method and baseline methods. The number in bold indicates that value from the outlier-based method is higher than value from the ranking wrapper method. For CHI, the outlier-based algorithm shows higher micro- and macro-averaged F1 than the ranking wrapper-based algorithm with statistical significance. From the definition of CHI, it is argued that the outlier-based algorithm selects better features, with higher relevance to the target categories, than the baseline algorithm. For IG, the outlier-based algorithm achieves lower micro-F1 than the baseline algorithm due to the influence of F1-measure of the largest category "earn". However, the outlier-based and baseline algorithms achieve no statistically significant difference in macro-F1. It seems that both algorithms with IG score are similar in average performance across different categories, without regarding to the size of training sets. For DF, the outlier-based and baseline algorithms obtained similar micro averaged F1's but the macro averaged F1 of the outlier-based is higher than of the baseline algorithm. Finally for DFD score, the outlier-based achieves macro averaged F1 higher than the baseline algorithm and similar micro averaged F1 as well. The experimental results show that the proposed outlier-based algorithm selects a more compact feature subset than the baseline algorithm while achieving similar or higher F1-values for different feature scores with statistical significance. In addition, DF with the outlier-based algorithm achieves highest both micro- and macro- F1 but its number of selected features is larger. DFD and CHI with the outlier-based seems to be better balance between performance and feature set size. Finally, IG with the outlier-based produces the lowest performance. This conforms to the suggestion in (Makrehchi, M. and M. S. Kamel, 2011) that IG with outlier-based feature selection is not recommended for the Reuters. We can argue that DFD and CHI are more suitable.

Table 5 Comparing F1-measures between feature subsets selected by the outlier-base method and the ranking wrapper-based method for CHI, IG, DF and DFD

Category	*CHI*			*IG*			*DF*			*DFD*		
	Outlier	**RW-based**		**Outlier**	**RW-based**		**Outlier**	**RW-based**		**Outlier**	**RW-based**	
	F1	**F1**	**p-Val**	**F1**	**F1**	**p-Val**	**F1**	**F1**	**p-Val**	**F1**	**F1**	**p-Val**
Earn	**95.63**	90.63	0.00+	90.69	**96.40**	0.00-	95.60	93.59	0.10	94.33	96.40	0.06
Acquisition	86.86	**90.58**	0.00-	**90.53**	86.98	0.00+	87.03	**89.01**	0.02-	88.44	87.12	0.06
Money-fx	66.22	**66.76**	0.00-	65.62	65.74	0.45	65.59	**67.07**	0.00-	**66.85**	65.59	0.00+
Trade	83.54	**84.74**	0.00-	83.90	84.16	0.75	**84.49**	83.11	0.00+	**85.61**	83.98	0.00+
Crude	70.30	**72.95**	0.00-	**72.68**	71.05	0.00+	72.06	72.10	0.61	**73.33**	71.76	0.00+
Interest	**77.77**	77.07	0.00+	78.61	**79.28**	0.00-	**81.08**	80.02	0.00+	78.22	**80.04**	0.00-
Ship	**86.72**	86.33	0.00+	83.95	**86.70**	0.00-	**88.60**	86.05	0.00+	86.40	**87.70**	0.00-
Corn	**77.38**	73.8	0.00+	**74.55**	70.91	0.00+	**74.44**	73.29	0.01+	71.48	72.26	0.12
Wheat	**84.00**	83.37	0.00+	81.81	**84.20**	0.00-	**86.00**	82.24	0.00+	80.51	**83.01**	0.00-
Grain	66.27	65.61	0.14	**62.17**	60.81	0.01+	**67.87**	64.17	0.00+	**67.89**	61.66	0.00+
Win/Lose/Tie	-	5/4/1		-	4/4/2		-	6/2/2		-	4/3/3	

Table 6 Micro-F1 and Macro-F1 comparison for four scores between proposed method and baseline method

Average F1-measure	**CHI**			**IG**			**DF**			**DFD**		
	proposed	**baseline**	**p-Val**	**proposed**	**baseline**	**p-Val**	**proposed**	**baseline**	**p-Val**	**proposed**	**baseline**	**p-Val**
Micro	**87.44**	86.42	0.00+	86.26	87.72	0.00-	87.83	87.24	0.05	87.44	87.88	0.07
Macro	**79.47**	79.18	0.01+	78.45	78.62	0.10	**80.28**	79.07	0.00+	**79.31**	78.95	0.02+

3. A statistics-based feature score for classifying class-imbalanced data

In this section, we compare performance of three proposed scores with various scores based on the experimental results reported in (Yang *et al.*, 2012). Those scores are Comprehensively Measure Feature Selection (CMFS), Information Gain (IG), Chi-square statistic (CHI), Improved Gini Index (GINI), Document Frequency (DF), Orthogonal Centroid Feature Selection (OCFS) and DIA association Factor (DIA). Figure 6 and Figure 7 show the micro F1 measure results when Support Vector Machines and Naïve Bayes are used on Reuters-21578 dataset, respectively.

Figure 6 indicates that the micro F1 performance of SVM based on Max-tscore and Averaged-tscore are superior to other feature scores. The results of Weighted Averaged-tscore are the worst because it is an average of tscore that weight of each class varies to its size. However, Averaged-tscore has not been so.

It can be seen from Figure 7 that the micro F1 performance of NB used Max-tscore outperforms that based on the other feature scores as well. However, Averaged-tscore gives results slightly lower than other scores and Weighted Averaged-tscore produces quite low classification performance.

Note that Max-tscore produces the best F1 results in both classifiers due to the score aims to maximize significance of feature relevance in one class against other classes rather than using average of the feature score for all classes. So, the score is suitable for situation where the class sizes are quite different. The micro F1 of SVM based on Max-tscore is the highest (94.20) when the number of the selected features is 400, and the micro F1 of NB based on Max-tscore reaches the highest (88.80) when the number of the selected features is 800. In the next section, we will discussion effectiveness of Max-tscore compared with others.

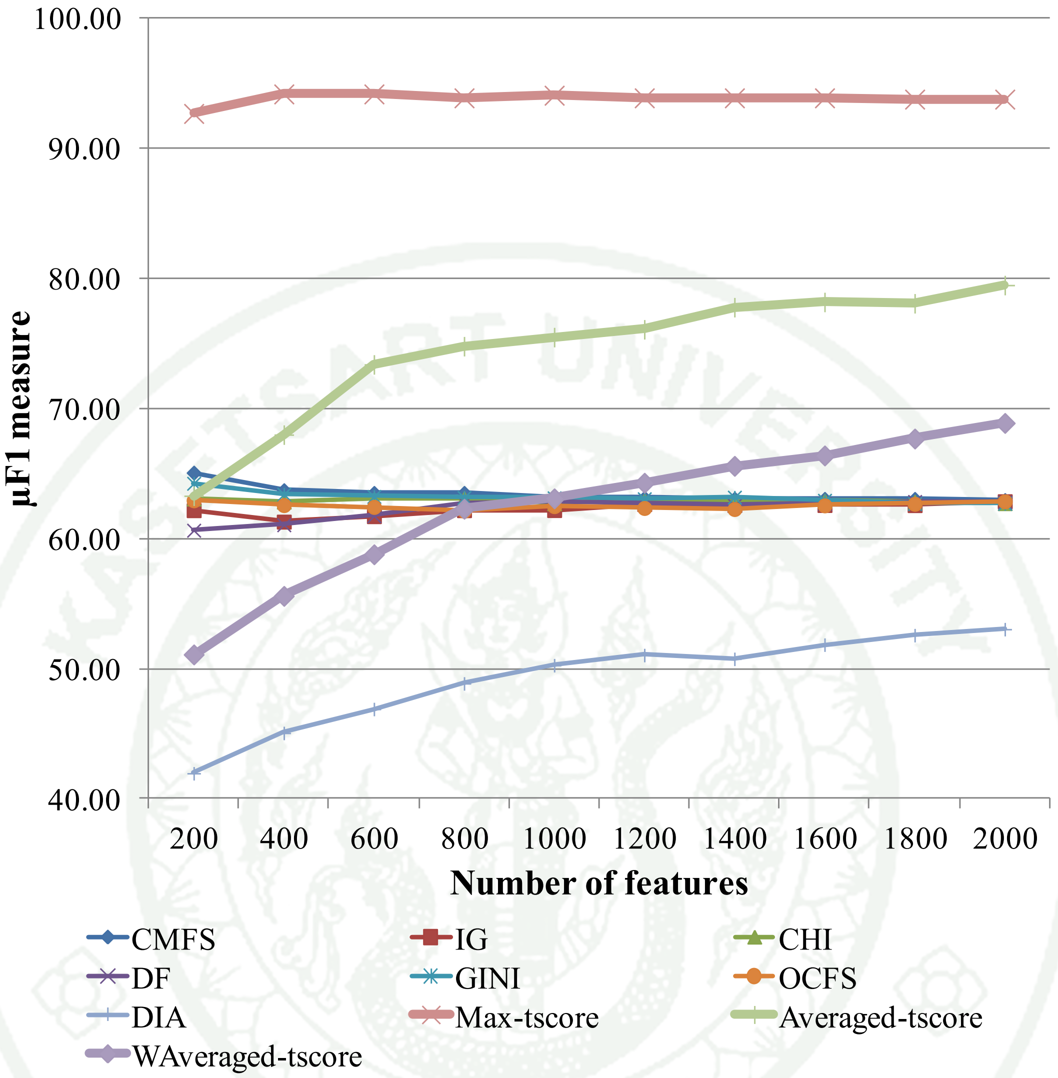

Figure 6 Micro F1 measure result using SVM on Reuters-21578

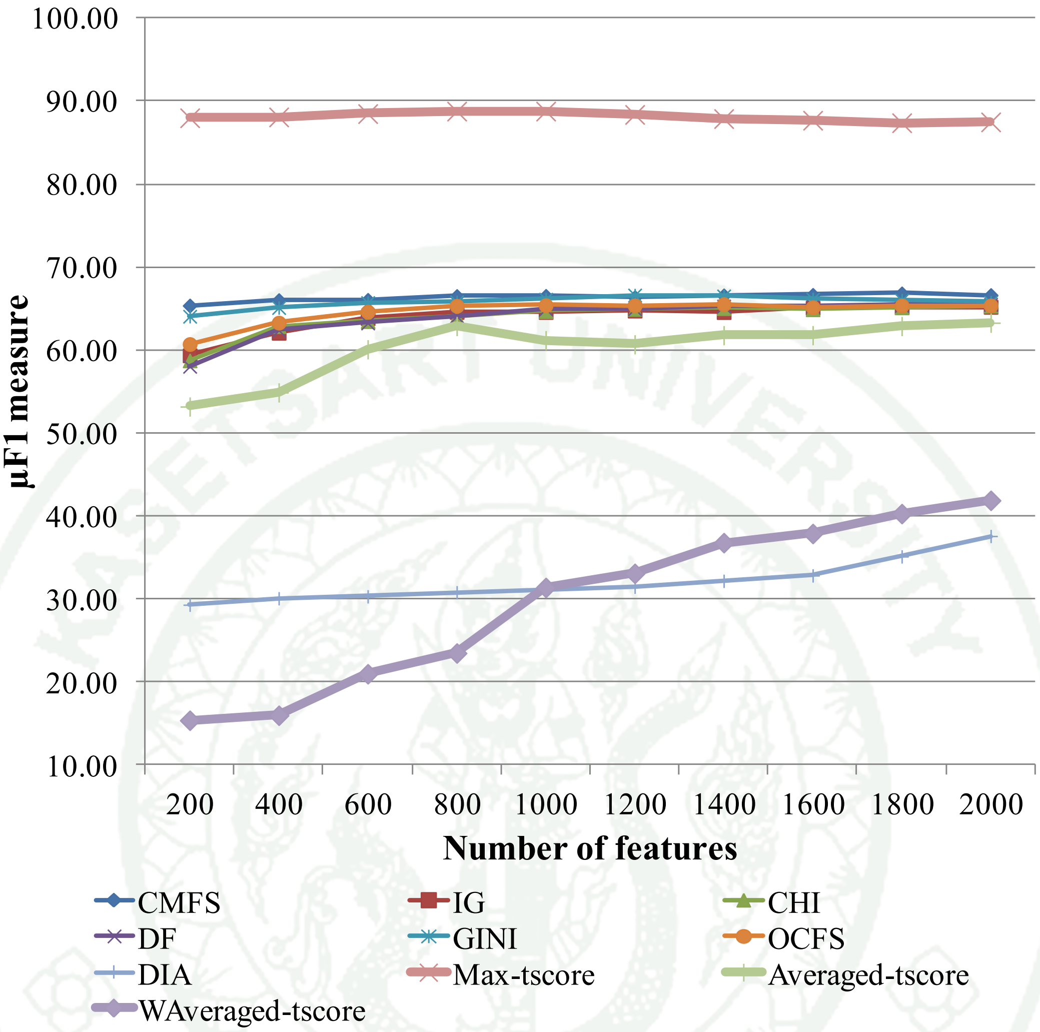

Figure 7 The micro F1 measure result using Naïve Bayes on Reuters-21578 Quality of selected features

Figure 8 shows that F1 measure of top-10 classes of Reuter-21578 dataset when various feature scores are used in order to consider the effect on the classification majority and minority classes. We compare the best result of Max-tscore with other scores based on the experimental results which are mentioned in (Yang, J. *et al.*, 2012). Note that CMFS score has previously reported which is the best performance score at all classes. However, our experiment shows that the Max-tscore clearly outperforms all feature scores under all classes especially very small size classes (corn, wheat and grain). We conclude that our score increase the F1 measure of the minority class without sacrificing the F1 measure of the majority class.

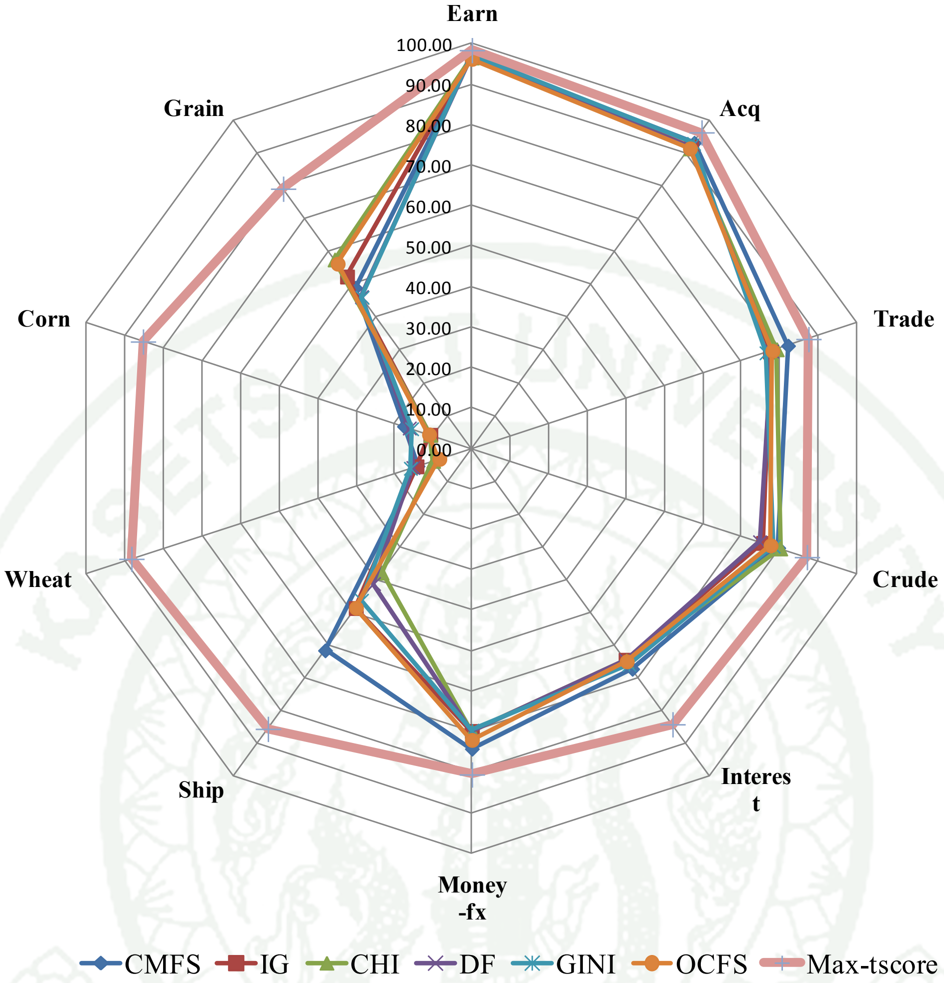

Figure 8 The best f1 of SVM classifier for each class when various feature score are used

Table 7 demonstrates top-10 features selected by various feature scores on Reuter-21578. It can be seen that some features are commonly selected by various scores. This means that the classification performance of common features is the same for those scores; thus, the features that are remained for common features become the key for discriminating the classification performance of various scores (Yang *et al.*, 2012). However, we can see that a feature "wheat" which is highly representative for the minority class (especially wheat, grain, corn) can be ranked in top 10 feature by the Max-tscore.

We believe that such features have more chance to be selected when we increase number of selected features. This is the reason for the superior micro F1 of the Max-tscore in both classifiers. Therefore, we can conclude that the Max-tscore can evaluate level of relevant features of each class and help the ranking method to select them in the optimal feature set that led to decrease the destructive impact of imbalance class distribution.

Table 7 The top10 features selected by various feature scores on Reuter-21578 dataset

Max-tscore	CMFS	IG	CHI	DF	GINI	OCFS	DIA
cts	cts	mln	cts	mln	cts	cts	shr
net	mln	dlrs	net	dlrs	net	net	qtr
shr	net	cts	shr	cts	shr	shr	revs
mln	loss	net	qtr	net	qtr	qtr	mths
oil	shr	pct	revs	loss	revs	oil	shrs
trade	oil	March	note	March	note	trade	div
profit	dlrs	year	loss	year	mln	wheat	avg
dlr	trade	loss	profit	shr	loss	bank	dividend
wheat	profit	billion	mths	pct	March	tonnes	qtly
bank	qtr	shr	trade	company	profit	company	cts

4. K-mean clustering based feature scoring method

In the experiment, we assumed that reducing size of large classes by clustering to smaller subclasses will make features in the small classes statistically more visible to the learning machine than results without clustering. We determined effectiveness of feature scoring measure among 1) the main classes, 2) the ground truth subclasses from the dataset, and 3) resulted K subclasses from the Algorithm 1. We applied Information Gain (IG) to measure feature scores on training data. Feature ranking method is executed with numbers of selected features |F|, ranged from 100 to 2000, stepped by 100 and from 2000 to 4000, stepped by 500. According to Algorithm 1, we generated a vector model for training data by using only top-ranked features instead of all features in order to reduce training time. Features are ranked by Document Frequency (DF). By empirical experiments, 4000 features are sufficient for acceptable clustering result. Then, we used *K*-means clustering onto the vector model. We selected several arbitrary values of |K| in the evaluation. First, we selected |K| = 55 since it is the same number as of ground-truth subclasses and then other lower values (22, 33, and 44) and one higher value (66) are arbitrarily selected in order to investigate effects of subclass sizes on the classification performance. The result, which is a set of clusters, was used as a subclass for measuring feature scores. Next, IG score for each feature was computed and the traditional feature ranking method proceeded.

We adopt Support Vector Machine (SVM) with linear kernel which is a favorite and robust learning algorithm for highly dimensional data as the classifier in the experiments. The experiment was performed using the LibSVM package of WEKA, with the default values of parameters. We investigated the effectiveness of the features for multi-class text categorization. Using traditional feature ranking method, the selected feature subsets were used to train the classifier; then the classification performance was evaluated on training documents.

Figure 9 shows the ranking accuracy of feature ranking method using clustering resulted subclass, ground truth subclass and main class on the RCV1v2 dataset. We observe that 1) the ground truth subclasses outperformed the main class when the number of feature further increases; 2) compared with main class, using ground truth subclasses for feature scoring measure can improve classification accuracy substantially; and 3) with |K| = 22, 33, 44, 55 and 66, our algorithm achieved higher accuracy than the main classes at number of feature from 100 to 4000 and case |K| = 44 performs the best. The accuracy values of both methods are approximately equal when number of features is more than 1800 with all |K| values. Moreover, the slope of ground-truth subclasses line is zero at 1800 features. Basically, the feature ranking method will select 1800 top-ranked features as the optimal feature subset.

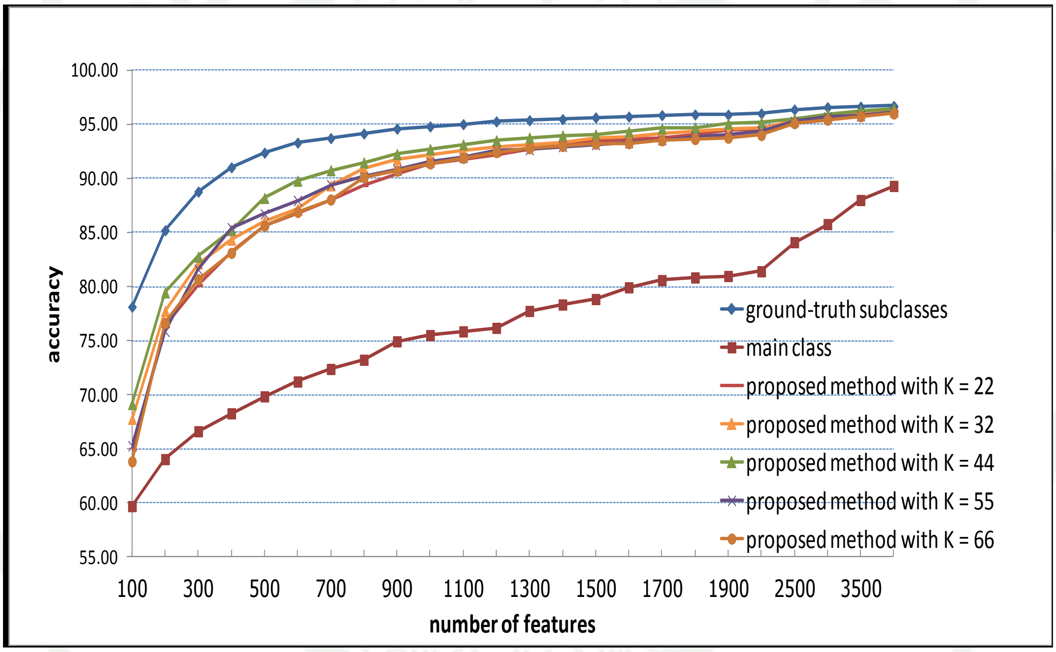

Figure 9 The ranking accuracy of feature ranking method using clustering resulted subclass, ground truth subclass and main class on the RCV1v2 dataset.

Next, classifiers from training process were used to classify testing documents and the classification performance was then evaluated. Figure 10 and Figure 11 illustrate the performance comparison at selected numbers of features (1800 and 4000 features) of the main class, ground truth subclass and resulted class where K = 44. Figure 10 shows that micro-averaged F1 of resulted class is slightly lower than of the ground truth subclasses. Since clustering technique is unsupervised learning method which has no prior knowledge, thus, a number of errors are normally occurred; 2) F1-measures in the smallest class (ECAT) are increased from 0.392 to 0.857 at 1800 features and from 0.547 to 0.881 at 4000 features; and 3) macro-averaged F1 are increased from 0.703 to 0.931 at 1800 features and 0.813 to 0.941 at 4000 features. Figure 11 shows that overall accuracy measures are increased from 79.4 to 93.9 at 1800 features and 87.86 to 95.4 at 4000 features.

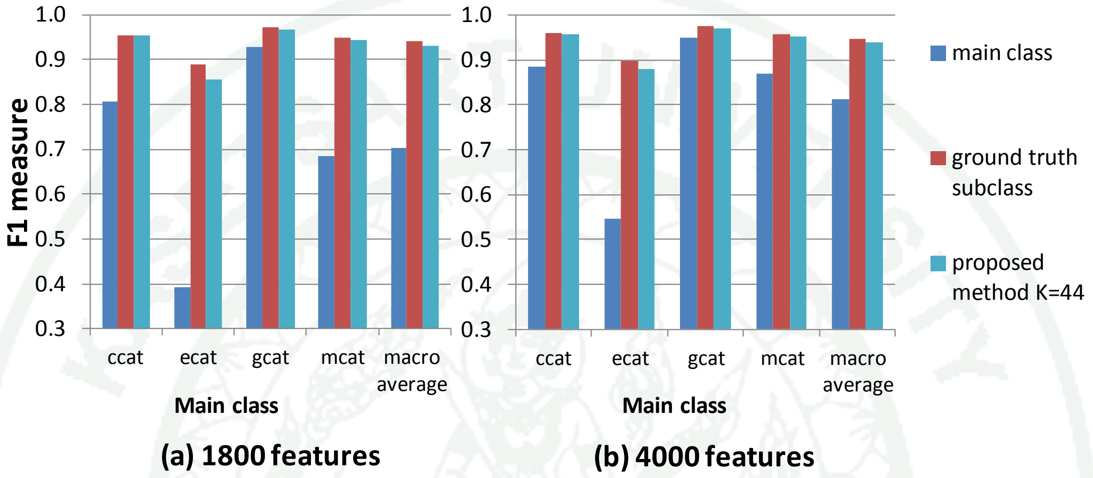

Figure 10 The F1 measures for 1800 features and 4000 features for class variants

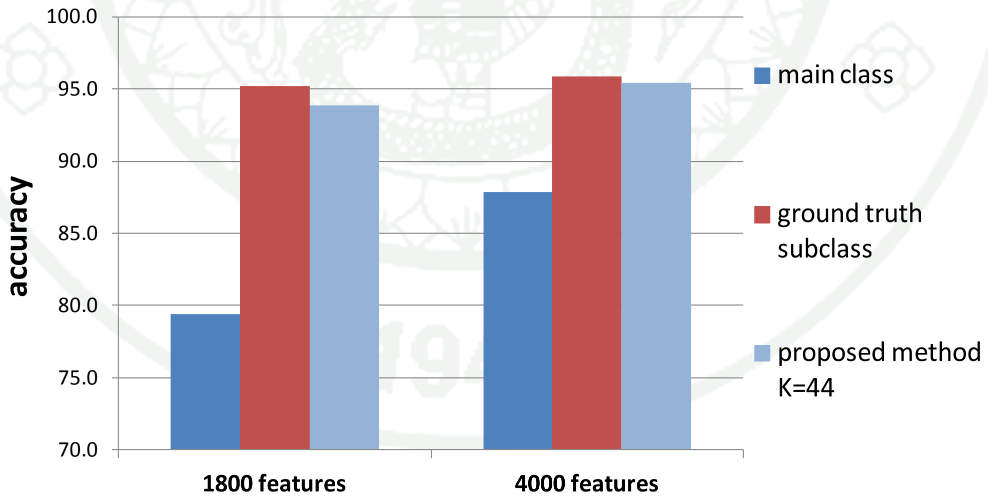

Figure 11 The accuracy for 1800 features and 4000 features for class variants

5. Hierarchical agglomerative clustering based feature scoring method

To evaluate the performance of the proposed method in the multi-class text classification, we employ the Accuracy defined by $Accuracy = \frac{\sum_{j=1}^{|C|} TP_j}{\sum_{j=1}^{|C|}(TP_j+FN_j)} \times 100$ True positive $\mathrm{TP_j}$ is a number of test documents that are classified into class $\mathrm{C_j}$ correctly, and false negative $\mathrm{FN_j}$ is a number of test documents incorrectly classified under other categories.

We also use F1 ($\mathrm{F1_j}$) as the local performance measure, which is a combination of precision P_j and recall R_j. The definition of $F1_j$ is $F1_j = \frac{2 \times P_j \times R_j}{(P_j+R_j)}$, where $P_j = \frac{TP_j}{TP_j+FP_j}$ and $R_j = \frac{TP_j}{TP_j+FN_j}$, which False positive FP_j is numbers of test documents that are classified into class C_j incorrectly. To exhibit the performance of the proposed method on the smaller class, the F1 measures over the multiple classes are summarized using the macro-averaged F1 (Sebastiani, 2002) defined by $MF1 = \frac{\sum_{j=1}^{|C|} F1_j}{|C|}$. The $MF1$ provides equal weights to all classes, regardless of number of documents in each class; therefore, it is not mainly dependent on large classes.

In the experiment, we assumed that reducing size of large classes by clustering to smaller subclasses will make features in the small classes statistically more visible to the learning machine than results without clustering.

We determined effectiveness of feature scoring measures among four feature sets from: 1) the main classes, 2) the ground truth subclasses from the dataset, and 3) resulted K subclasses from the HAC algorithm with baseline cut-off value, and 4) resulted K subclasses from the Algorithm 1 with class-sensitive cut-off value.

We generated a vector model for training data by using 3,000 top-ranked features instead of all features in order to reduce training time and preliminary empirical experiments showed that 3,000 features are sufficient for acceptable clustering results. These features are ranked by Document Frequency (DF) in this process.

We applied Information Gain (IG) to measure feature scores on training data which is labeled with four different class labels. First, main class is used as the baseline method. Second, ground truth subclasses are extracted from metadata of RCV1v2. Third, we used HAC algorithm onto the vector model. We manually analyze several arbitrary values of cut-off threshold for each main class and evaluate the results of HAC algorithm and empirically select the best resulted value as the baseline cut-off. The baseline cut-off can cluster the main classes into 161 result subclasses which consist of 15, 29, 48 and 69 subclasses of main classes CCAT, ECAT, GCAT and MCAT respectively. The result, which is a set of clusters, was used as a subclass for measuring feature score.

After finding the baseline cut-off, the Algorithm 1 has been applied on the vector space of training data. By setting the value of a parameter $\alpha = 0.01$, Algorithm 1 obtains the resulted cluster which is consisted of 284 subclasses (134, 37, 35 and 81 subclasses from main classes CCAT, ECAT, GCAT and MCAT, respectively.) The resulted clusters were used as subclasses for measuring score as well.

Then, the traditional feature ranking method proceeded. Feature ranking method is executed with numbers of selected features |F|, ranged from 100 to 2000, stepped by 100 and from 2000 to 3000, stepped by 500. We adopt SVM with linear kernel and Naïve Bayes as the classifiers in the experiments. Using traditional feature ranking method, the selected feature subsets were used to train the classifier; then the classification performance was evaluated on training documents. The performance of the classifier is estimated by four-fold cross-validation. To eliminate random variation, the performances were averaged over the three runs. We use Student’s paired two-tailed t-test in order to evaluate the statistical significance (at the significant level 0.05) of the difference between the various results.

5.1 Results on Naïve Bayes

Figure 12 shows the ranking accuracy of Naïve Bayes classifier with feature ranking method using clustering resulted subclass, ground truth subclass and main class on the RCV1v2 training dataset. We observe that 1) the ground truth subclasses outperformed the main class when the number of feature further increases, thus, using subclasses for feature scoring measure can improve accuracy substantially of low-effective classifier such as Naïve Bayes; 2) using baseline cut-off value, resulted sub-classes achieved accuracy which is similar trend to the ground truth subclass (with correlation coefficient > 0.9); and 3) our algorithm with class-sensitive cut-off achieved higher accuracy that is superior to the main classes at lower number of feature from 100 to 1100 but inferior to the ground truth subclasses, and main class at higher number of feature from 1100 to 3000. However, this work has focused on selection small number of feature for classification. The accuracy of resulted subclasses by the proposed method is maximal at 400 features which is higher than the maximum accuracy of the main class. By the ranking wrapper-based paradigm, thus the 400 top-ranked features are selected as the optimal feature subset for Naïve Bayes classification for class variants.

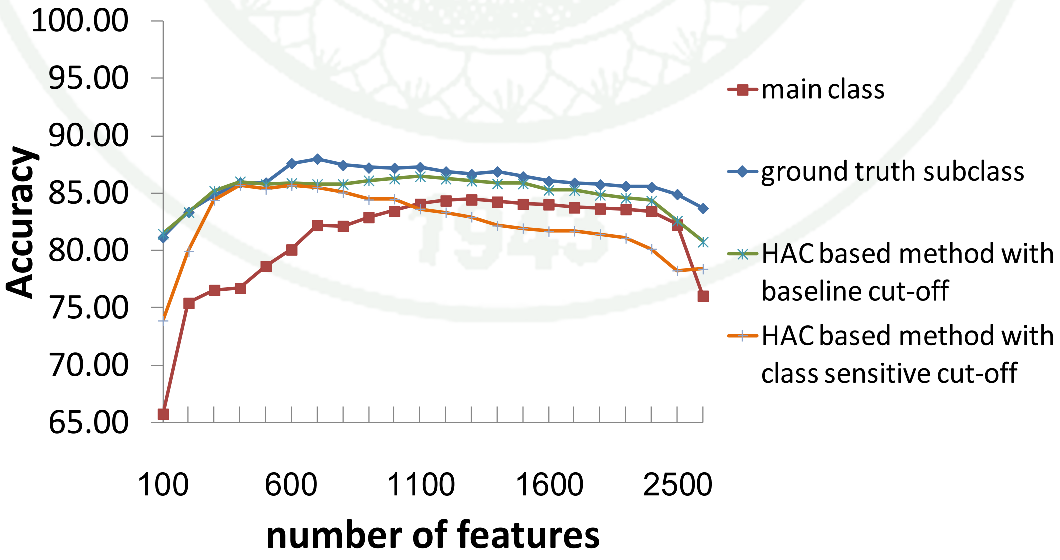

Figure 12 Accuracy of NB on the RCV1v2 dataset.

Next, Naïve Bayes classifiers from training process were used to classify testing documents and the classification performance was then evaluated. Figure 13 and Figure 14 illustrates the performance comparison at selected numbers of features (400 features) of the main class, ground truth subclass and resulted class. Figure 13 shows that macro-averaged F1 of Naïve Bayes of resulted class is slightly lower than of the ground truth subclasses in the same result as the SVM classifier because of normally occurred errors from the clustering method. However, the replacement main classes by resulted subclasses makes the classification obtain the F1-measures in the smallest class (ECAT) increased from 0.641 to 0.666. The macro-averaged F1 are increased from 0.743 to 0.811. Especially, Figure 14 shows that recall of ECAT is increased from 0.849 to 0.873 and the averaged recall is increased from 0.805 to 0.850.

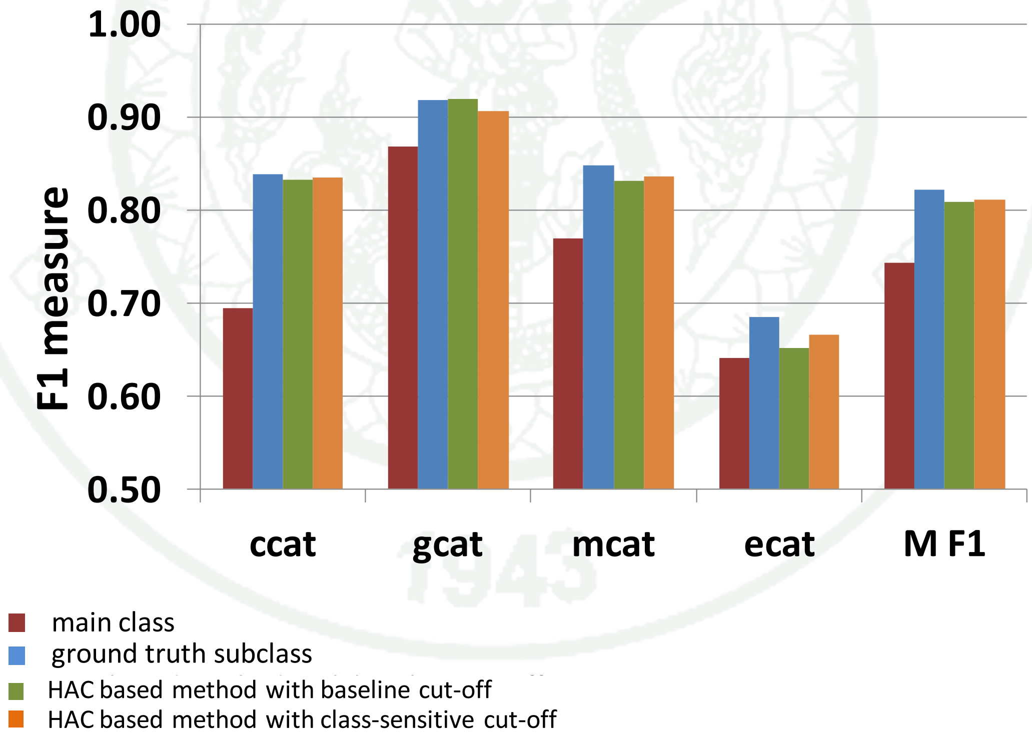

Figure 13 The F1 measures of NB with 400 features for class variants

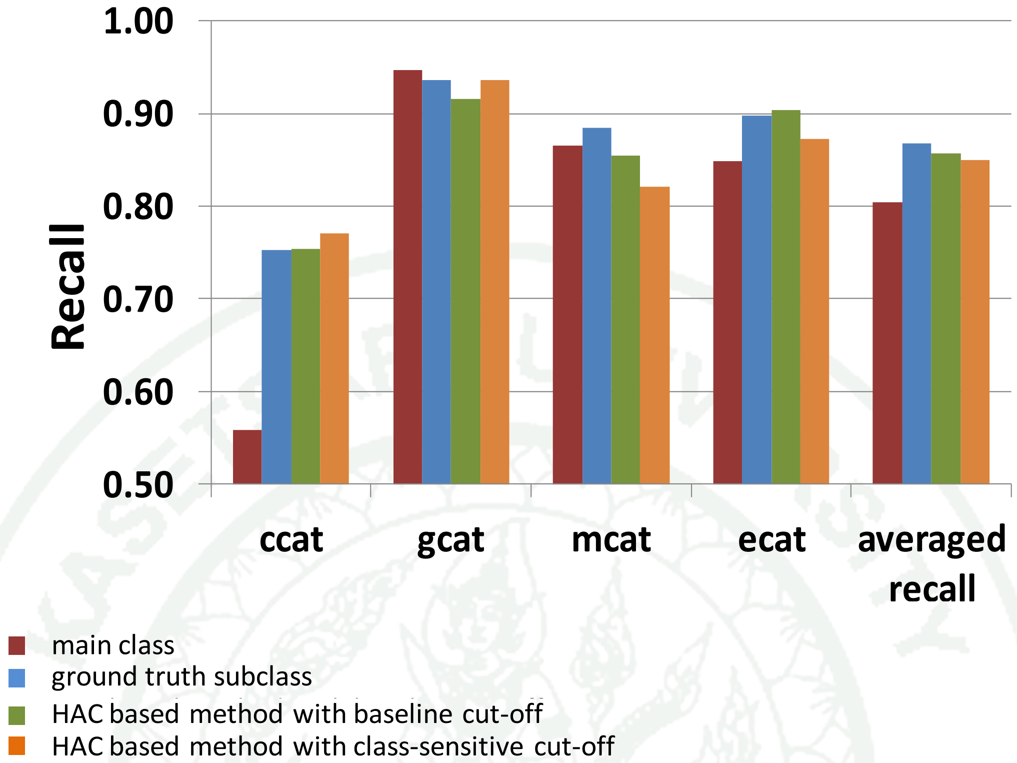

Figure 14 The Recall of NB with 400 features for class variants

5.3 Results on SVM

Figure 15 shows the ranking accuracy of SVM classifier with feature ranking method using clustering resulted subclass, ground truth subclass and main class on the RCV1v2 training dataset. We observe that 1) the ground truth subclasses also outperformed the main class when the number of feature further increases; 2) compared with main class, using ground truth subclasses for feature scoring measure can improve classification accuracy substantially; 3) using baseline cut-off value, resulted subclasses achieved equal accuracy to the ground truth subclasses; 4) our algorithm achieved higher accuracy than the main classes at number of feature from 100 to 3000 and approximately equal to the ground truth subclasses; and 5) all lines in the Figure 3 are similar trend (with correlation coefficient > 0.9). The slope of resulted subclasses line by the proposed method is zero at 1300 features. Basically, the feature ranking method will select 1300 top-ranked features as the optimal feature subset for SVM classifier for class variants.

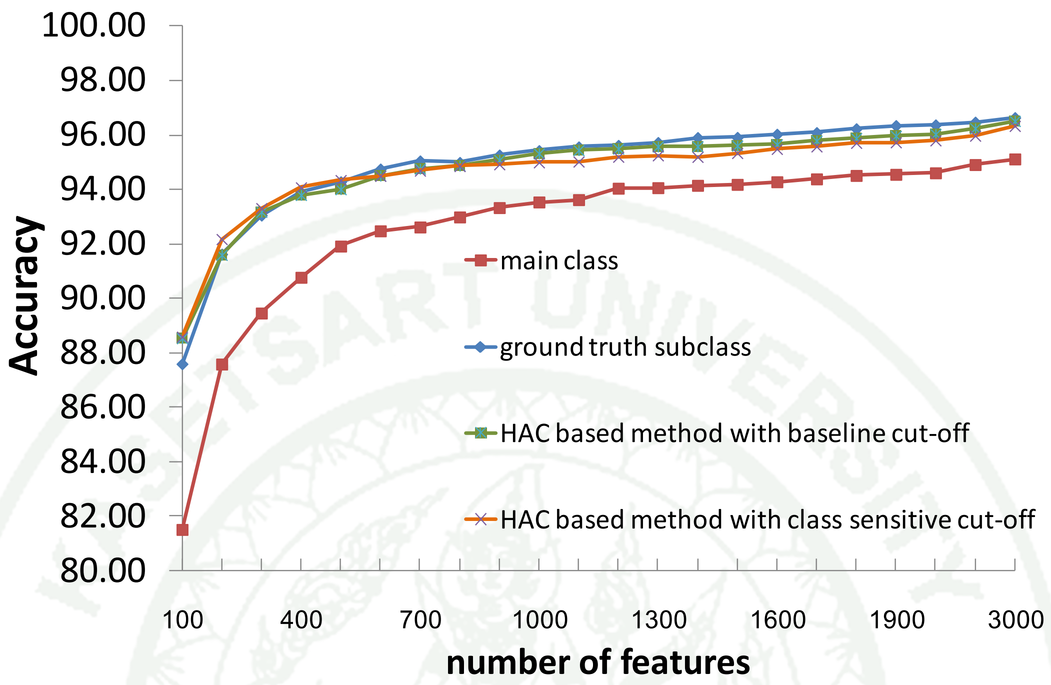

Figure 15 Accuracy of SVM on the RCV1v2 dataset.

Next, SVM classifiers from training process were used to classify testing documents and the classification performance was then evaluated. Figure 16 and Figure 17 illustrates performance comparison at selected numbers of features (1300 features) of the main class, ground truth subclass and resulted class. Figure 16 shows that macro-averaged F1 of resulted class is slightly lower than of the ground truth subclasses. Since clustering technique is unsupervised learning method which has no prior knowledge, thus, a number of errors are normally occurred. F1-measures in the smallest class (ECAT) are also increased from 0.862 to 0.872. The macro-averaged F1 are increased from 0.918 to 0.931. Figure 17 shows that recall of ECAT is increased as well, from 0.809 to 0.829 and the averaged recall is increased from 0.904to 0.923.

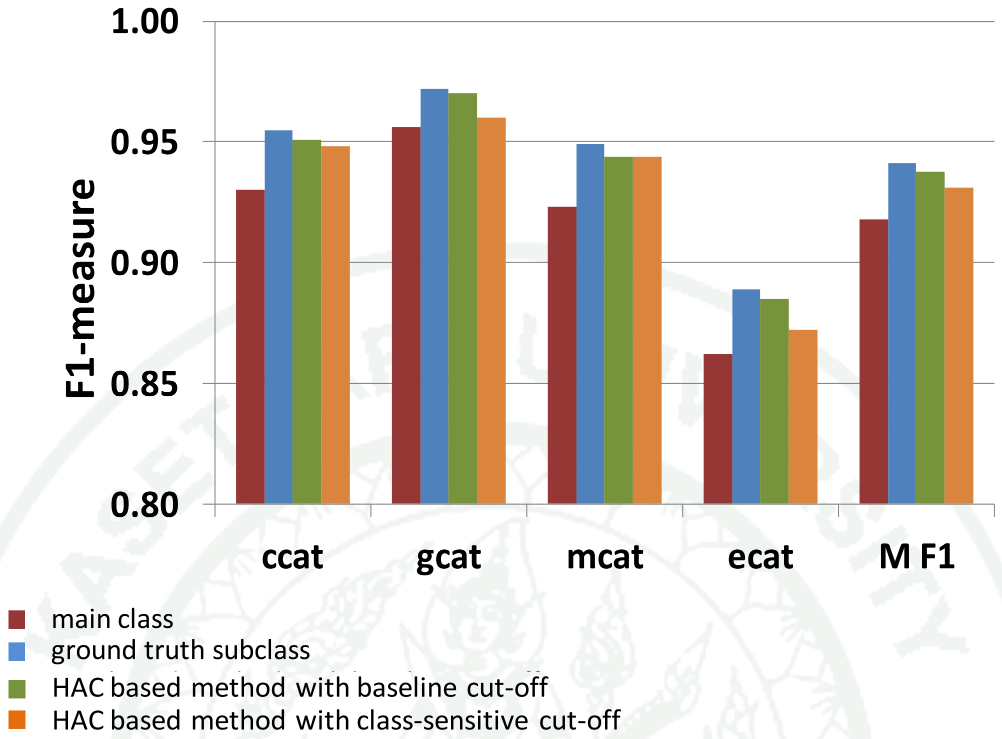

Figure 16 The F1 measures of SVM with 1300 selected features for class variants

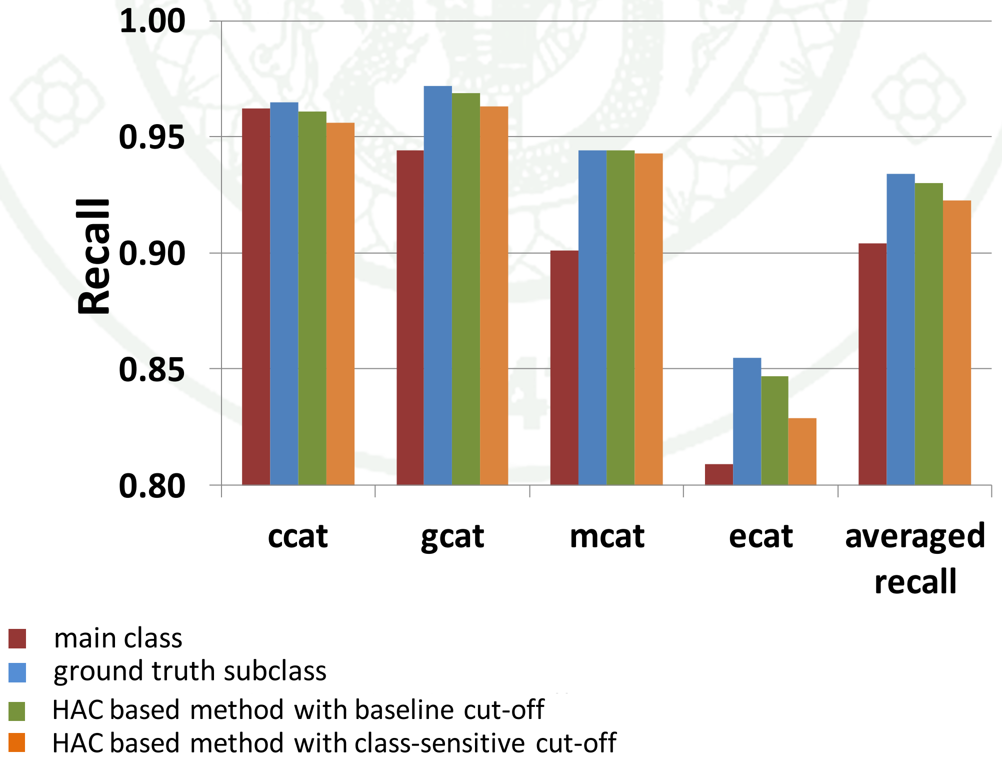

Figure 17 The Recall of SVM with 1300 selected features for class variants

6. Unsupervised Fast correlation-based feature selection for clustering

In the experiment, we use the RCV1v2 dataset and choose the data samples with the highest four topic codes (CCAT, ECAT, GCAT, MCAT) in the “Topic Codes” hierarchy which contains 19,806 training documents and 16,942 testing documents. Furthermore, we also use Isolet dataset. There are 617 real features with 7797 instances and 26 classes. We generate a vector model for training data without using class label based on our selected features. Then, we use K-Means clustering onto the vector model. The result, which is a set of clusters, will be used as a new model for clustering onto the testing data (also without using class label.) The labels assigned by clustering testing data are used to compare with class labels given from corpora.

In order to assess clustering performance under different feature selection method, three qualitative measures are selected. For Average Accuracy (AA) (Luying et al., 2005, Shamsinejadbabki and Saraee, 2011) and RS Rand Statistics (RS) (Achtert *et al.*, 2012;Luying *et al.*, 2005a), we count number of documents, which have the same topics, in the same cluster and number of documents, which have different topics, in different clusters. In our clusters and in the corpus, both documents are placed in the same clusters: *ss*. In our clusters both documents are placed in the same clusters but in corpus they are in different clusters: *sd*. In our clusters documents are placed in different clusters but in the corpus they are in the same clusters: *ds*. In our clusters and in the corpus both documents are placed in different clusters: *dd*. Then, AA and RS are defined as follows:

$$AA = \frac{1}{2} \times \left(\frac{ss}{ss+ds} + \frac{dd}{sd+dd}\right)$$

$$RS = \frac{(ss+dd)}{ss+sd+ds+dd}$$

Another measure for evaluating clustering is the macro F1-measure (F1) (Mitra *et al.*, 2012;Shamsinejadbabki and Saraee, 2012) that is evaluated as

$$F1 = \sum_{i} \frac{n_i}{N} \left[max_{j \in C} \left\{ \frac{2 \times Recall(i,j) \times Precision(i,j)}{Recall(i,j) + Precision(i,j)} \right\} \right]$$

which $Recall(i,j) = \frac{n_{ij}}{n_i}, Precision(i,j) = \frac{n_{ij}}{n_j}$, where n_{ij} is the number of instances belonging to class *i* in corpus that falls in cluster *j*, and n_i, n_j are the cardinalities of class *i* cluster *j* respectively.

Our proposed selection algorithm is evaluated by comparing the effectiveness of our optimal feature subset with other subsets selected by a baseline algorithm. We use ranking wrapper-based feature selection, which is the most preferable filter-based method to determine the best number of highly relevance feature as the baseline. We applied four unsupervised feature scoring schemes, DF, TC, TV and MM, to determine feature subset on training documents of RCV1v2 dataset with different cut-off number of features ranging from 500 to 4000. At each cut-off number, the performance of the K-Means clustering for the selected feature subset is estimated by 10-fold 10-time cross-validation. The results show that the optimal number of features for DF, TC, TV and MM are 1300, 500, 500 and 500, respectively.

We then used the proposed outlier-based feature selection in Algorithm 1 to select a feature subset with these four feature scores. We set merely a parameter α, to identify the optimal threshold. From preliminary parameter setting, numbers of features selected by the proposed algorithm for DF, TC, TV, and MM are 486, 483, 537, and 470, respectively. The result shows that the proposed algorithm almost selects a smaller feature subset when compared with the feature subset selected using the baseline algorithm. The selected feature subsets are used to train the clustering; then the clustering performance is evaluated on testing documents.

Table 8 shows the performance of features selected by each algorithm for DF, IC, IV, and MM. The performance values in each table are 10-run average values. We compute Student’s independent two-tailed t-test in order to evaluate the statistical significance of the difference between the two averaged values: the one from the proposed method and the one from baseline method. The *p*-Val is the probability associated with an independent two-tailed t-Test. The “compare” means that the proposed method is statistically significant (at the 0.05 level) win or loss over the baseline methods and equal means no statistically significant difference. The experimental result shows that the proposed method with four feature scores get comparably equal clustering performance to the baseline method.

Moreover, we also evaluate performance of the proposed method on the Isolet with of baseline method presented. Table 9 shows comparison of validation. The number of selected features from the proposed method is lower than number from the baseline method. Table 2 shows that proposed algorithm achieve higher F1 than the baseline and RS value of both methods are equal.

Table 8 Comparing three performance measures between feature subsets selected by the UFCBF method and the baseline method for DF, TC, TV and MM on RCV1v2 dataset.

Feature Score	Method	#feature	F1	AA	RS
DF	baseline	1300	0.688	0.553	0.626
	UFCBF	486	0.693	0.554	0.632
	p-Value		0.857	0.969	0.722
	compare		equal	equal	equal
TC	baseline	500	0.667	0.525	0.600
	UFCBF	483	0.684	0.557	0.633
	p-Value		0.588	0.278	0.234
	compare		equal	equal	equal
TV	baseline	500	0.683	0.552	0.626
	UFCBF	537	0.645	0.555	0.629
	p-Value		0.204	0.837	0.841
	compare		equal	equal	equal
MM	baseline	500	0.665	0.539	0.613
	UFCBF	470	0.702	0.563	0.638
	p-Value		0.149	0.143	0.127
	compare		equal	equal	equal

Table 9 A comparison of the performance on Isolet dataset

Method	#feature	F1	Rand	AA
baseline	617	0.365	-	-
	274	0.336	0.94	-
	275	0.344	0.94	-
Purposed	263	0.530	0.94	0.62
Compare		Win	equal	-

Discussion

We conclude that the outlier based feature selection algorithm (in Algorithm 2) can select a smaller set of highly relevant features to the categories of interest using only one parameter (significant level) without prior knowledge. For task of text categorization with imbalanced category size, the experimental results show that the outlier-based feature selection efficiently obtains a high degree of dimensionality reduction and it is practical for highly-dimensional data. By comparing F1, micro-F1, and macro-F1 measures, results showed that classification accuracy of the proposed algorithm significantly exceeds of the ranking wrapper-based algorithms. Its non-iterative process has low computational cost and in experiments it was never trapped at the local optima. The results were the same using different feature scores. Therefore, the proposed outlier-based feature selection method is likely to be invariant to feature score.

The proposed DFD score obtained the largest micro-averaged F1 among these four feature scores in the classifier and its macro-averaged F1 is higher than IG as well. As we can see in the results obtained from the DFD are statistically better than from the IG. For micro-averaged F1, DFD is better than CHI and DF. However, for macro-averaged F1, DFD is statistically similar to DF and slightly smaller than CHI. These shows that DFD works well and can be used instead of DF, CHI and IG when the distribution of the category is imbalanced.

In the experiment section 3, the results showed that one of the proposed scores, the Max-tscore based on maximizing significance of feature relevancy for all classes, works very well on the Reuter21578 dataset, especially the F1 of the minority classes are improved without sacrificing the F1 of majority classes. Since the highly relevant features, which are the key for discriminating the classification performance of various scores, can be ranked in the top feature by the Max-tscore. Such features have more chance to be selected when number of selected features is increased. This is the reason why the superior micro F1 of the Max-tscore in both classifiers. Therefore, we conclude that the statistic based feature score can evaluate level of relevant features of each class and helps the ranking based feature selection method to select them in the optimal feature set that led to decrease the destructive impact of imbalance class distribution. Therefore, the t-test score is insensitive to the problem of imbalanced class distribution. This means that the proposed score can be used instead of the other feature scores when the distribution of the class is highly imbalanced.

By the experimental results subsection 4, we can summarize that using only 1800 features performances (F1 and accuracy) of our proposed with K=44 are slightly lower than ground truth; however, it is much better than using main class only. More importantly, performances on the smaller classes are much improved. Therefore, K-mean cluster based method can obtain resulted cluster which can be used instead of main class for feature scoring measure on multi-class dataset with highly imbalance category sizes. Particularly, if the ground truth subclasses are unavailable, the proposed algorithm based on clustering technique can still process and yields the sufficient results. The suitable parameter |K| may be tuned by empirical process but we argue that parameter |K| should be defined as equal as or lower than number of ground truth subclasses.

We can summarize the experimental results in subsection5 that HAC based method can reduce effect of class-imbalanced distribution in the dataset as well. Performances on the proposed smaller classes are much improved from using original main classes. HAC algorithm can be used to re-cluster efficient subclasses for selecting features. Selected features are effective when they are evaluated on both SVM and Naïve Bayes. Including a method for determining appropriate cut-off threshold based on a statistical parameter and class-sensitive weighting can automatically, efficiently and effectively separate the imbalance dataset. Particularly in the real application, if the ground truth subclasses do not exist, the proposed algorithm is still available and it yields the sufficient results.

Finally, the unsupervised fast correlation-based feature selection is the effective and computationally efficient algorithm that dramatically reduces size of feature set in high dimensional datasets for data clustering. The algorithm eliminates a large number of irrelevant and redundant features and selects a subset of informative features that provide more discriminating power for unsupervised learning model. When we tested it on the RCV1v2 and Isolet data corpus, the results confirm that the proposed algorithm can greatly reduce the size of feature sets while maintaining the clustering performance of learning algorithms. The algorithm uses a simple statistic based threshold determination to develop a novel filter-based feature selection technique. Our approach does not require iterative empirical processing or prior knowledge. Compared to traditional hybrid feature selection the optimal subset from our proposed algorithm is significantly comparable or even better than the baseline algorithm. Experiments showed that proposed method works well and can be used with the unsupervised learning algorithm which class information is unavailable and the dimension of data is extremely high. Our proposed method can not only reduce the computation cost of text document analysis but can also be applied to other textual analysis applications.

CONCLUSION AND RECOMMENDATION

Conclusion

In this research, we propose an effective and efficient algorithm that dramatically reduces the dimensionality for multi-class Text Categorization and Text Clustering. The propose method drastically eliminates a large number of irrelevant feature, selects a set of relevant feature, extracts all significant relevant signature features and establishes the optimal feature set of mostly informative features that provide more discriminating power for text document learning model.

We use several different text document datasets as training and testing data; and apply the proposed algorithm to explore a subset of terms from a highly-dimensional original term set. Our experimental results show that the use of proposed algorithm can radically reduce the dimensionality of data sets while maintaining or improving the classification performance of learning algorithms. The algorithm uses simple statistical techniques (standard score, outlier detection, hypothesis t-Testing), data-adaptive threshold technique and correlation function in collaboration to develop a novel feature selection technique for text analysis. Our approach is non-iteration empirical processing and prior knowledge is not provided. Compared to traditional hybrid feature selection (scoring method, wrapper algorithms with sequential forward search), the proposed method explored optimal subset that significantly outperforms the existing algorithm. However, the proposed method has been proven that it is easier and it requires less computing time than using traditional wrapper algorithms. Our method can not only reduce the computation cost of text document analysis but can also be applied to other textual analysis applications. In addition to ease of use, this approach effectively addresses the feature redundancy problem.

Recommendation

The limitation of this study is that data used in this analysis are only text document data. Therefore, the results may not be the same even though we believe that the same process can be applied. Factor analysis may reveal semantic explanation. We believe that factor analysis could be a better support text document data analysis for further in-depth study.

LITERATURE CITED

Achtert, E., S. Goldhofer, H.-P. Kriegel, E. Schubert and A. Zimek. 2012. Evaluation of Clusterings - Metrics and Visual Support, pp. 1285-1288. *In* **ICDE'12.**

Aggarwal, C.C. and P.S. Yu. 2005. An Effective and Efficient Algorithm for High-Dimensional Outlier Detection. **The VLDB Journal** 14 (2): 211-221.

Aghdam, M.H., N. Ghasem-Aghaee and M.E. Basiri. 2009. Text Feature Selection Using Ant Colony Optimization. **Expert Systems with Applications** 36 (3, Part 2): 6843-6853.

Alelyani, S., J. Tang and H. Liu. 2013. Feature Selection for Clustering: A Review, pp. 29-60. *In* C. Aggarwal and C. Reddy, eds. **Data Clustering: Algorithms and Applications.** CRC Press.

Almeida, L.P., A.R. Vasconcelos and M.G. Maia. 2009. A Simple and Fast Term Selection Procedure for Text Clustering, pp. 47-64. *In* N. Nedjah, L. Macedo Mourelle, J. Kacprzyk, F. G. França and A. De Souza, eds. **Intelligent Text Categorization and Clustering.** Springer Berlin Heidelberg.

Arauzo-Azofra, A., J. Benitez and J. Castro. 2008. Consistency Measures for Feature Selection. **Journal of Intelligent Information Systems** 30 (3): 273-292.

Bache, K. and M. Lichman. (2013) Uci Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, Irvine, CA.

Baeza-Yates, R. and B. Ribeiro-Neto. 1999. **Modern Information Retrieval.** Addison Wesley Longman, New York.

Ben-Gal, I. 2005. Outlier Detection, p. *In* M. O. and R. L., eds. **Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers.** Kluwer Academic Publishers.

Blachnik, M., W. Duch, A. Kachel and J. Biesiada. 2009. Feature Selection for Supervised Classificaiton: A Kolmogorov-Smirnov Class Correlation-Based Filter, p. *In* **AIMeth, Symposium on Methods of Artificial Intelligence.** Gliwice, Poland.

Carter, N.J., N.C. Schwertman and T.L. Kiser. 2009. A Comparison of Two Boxplot Methods for Detecting Univariate Outliers Which Adjust for Sample Size and Asymmetry. **Statistical Methodology** 6 (6): 604-621.

Combarro, E.F., E. Montanes, I. Diaz, J. Ranilla and R. Mones. 2005. Introducing a Family of Linear Measures for Feature Selection in Text Categorization. **IEEE Transactions on Knowledge and Data Engineering** 17 (9): 1223-1232.

Dai, J., Z. He and F. Hu. 2009. A High Performance Algorithm for Text Feature Automatic Selection, pp. 414-417. *In* **Proceedings of the 2009 International Symposium on Information Processing (ISIP'09).** Academy Publisher, Huangshan, P.R. China.

Duangsoithong, R. and T. Windeatt. 2009. Relevant and Redundant Feature Analysis with Ensemble Classification, pp. 247-250. *In* **Advances in Pattern Recognition, 2009. ICAPR '09. Seventh International Conference on.**

Ferreira, A. and M. Figueiredo. 2011. Efficient Unsupervised Feature Selection for Sparse Data, pp. 1-4. *In* **EUROCON - International Conference on Computer as a Tool (EUROCON), 2011 IEEE.**

Ferreira, A.J. and M.A.T. Figueiredo. 2012. Efficient Feature Selection Filters for High-Dimensional Data. **Pattern Recognition Letters** 33 (13): 1794-1804.

Foithong, S., O. Pinngern and B. Attachoo. 2012. Feature Subset Selection Wrapper Based on Mutual Information and Rough Sets. **Expert Systems with Applications** 39 (1): 574-584.

Forman, G. 2003. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. **J. Mach. Learn. Res.** 3 1289-1305.

Guyon, I., Andr, #233 and Elisseeff. 2003. An Introduction to Variable and Feature Selection. **J. Mach. Learn. Res.** 3 1157-1182.

Hastie, T., R. Tibshirani and J. Friedman. 2008. **The Elements of Statistical Learning: Data Mining, Inference and Prediction.** Springer.

He, H. and E.A. Garcia. 2009. Learning from Imbalanced Data. **IEEE Trans. on Knowl. and Data Eng.** 21 (9): 1263-1284.

Hodge, V. and J. Austin. 2004. A Survey of Outlier Detection Methodologies. **Artificial Intelligence Review** 22 (2): 85-126.

Hsiao, W.-F. and T.-M. Chang. 2008. An Incremental Cluster-Based Approach to Spam Filtering. **Expert Systems with Applications** 34 (3): 1599-1608.

Jo, T. and N. Japkowicz. 2004. Class Imbalances Versus Small Disjuncts. **SIGKDD Explor. Newsl.** 6 (1): 40-49.

Johnson, R.A. and D.W. Wichern. 2002. **Applied Multivariate Statistical Analysis.** 5 th. Prentice Hall, Englewood Cliffs, New Jersey.

Kihoon, Y. and S. Kwek. 2005. An Unsupervised Learning Approach to Resolving the Data Imbalanced Issue in Supervised Learning Problems in Functional Genomics, pp. 6-11. *In* **Hybrid Intelligent Systems, 2005. HIS '05. Fifth International Conference on.**

Lee, L.-W. and S.-M. Chen. 2006. New Methods for Text Categorization Based on a New Feature Selection Method and a New Similarity Measure between Documents, pp. 1280-1289. *In* M. Ali and R. Dapoigny, eds. **Advances in Applied Artificial Intelligence.** Springer Berlin / Heidelberg.

Lee, S., J. Song and Y. Kim. 2010. An Empirical Comparison of Four Text Mining Methods. **Journal of Computer Information Systems** 51 (1): 1-10.

Lewis, D.D., Y. Yang, T. Rose and F. Li. 2004. Rcv1: A New Benchmark Collection for Text Categorization Research. **Journal of Machine Learning Research** 5 361-397.

Liu, H. and L. Yu. 2005. Toward Integrating Feature Selection Algorithms for Classification and Clustering. **IEEE Transactions on Knowledge and Data Engineering** 17 (4): 491-502.

Liu, T., S. Liu and Z. Chen. 2003. An Evaluation on Feature Selection for Text Clustering, pp. 488-495. *In* **In ICML.**

López, V., A. Fernández, S. García, V. Palade and F. Herrera. 2013. An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. **Information Sciences** 250 (0): 113-141.

Luying, L., K. Jianchu, Y. Jing and W. Zhongliang. 2005a. A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering, pp. 597-601. *In* **Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference on.**

Luying, L., K. Jianchu, Y. Jing and W. Zhongliang. 2005b. A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering, 597-601. *In*.

MacQueen, J.B. 1967. Some Methods for Classification and Analysis of Multivariate Observations, pp. 281-297. *In* L. M. L. Cam and J. Neyman. **Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability.** University of California Press.

Makrehchi, M. and M.S. Kamel. 2005. Text Classification Using Small Number of Features, pp. 636-636. *In* P. Perner and A. Imiya, eds. **Machine Learning and Data Mining in Pattern Recognition.** Springer Berlin / Heidelberg.

Makrehchi, M. and M.S. Kamel. 2011. Impact of Term Dependency and Class Imbalance on the Performance of Feature Ranking Methods. **International Journal of Pattern Recognition and Artificial Intelligence** 25 (07): 953-983.

Manning, C.D., P. Raghavan and H. Sch\"utze. 2008. **Introduction to Information Retrieval.** Cambridge University Press, New York, NY, USA.

Mitra, S., P.P. Kundu and W. Pedrycz. 2012. Feature Selection Using Structural Similarity. **Information Sciences** 198 (0): 48-61.

Nuntiyagul, A., K. Naruedomkul, N. Cercone and D. Wongsawang. 2007. Keyword Extraction Strategy for Item Banks Text Categorization. **Computational Intelligence** 23 28-44.

Pramokchon, P. and P. Piamsa-nga. 2014a. An Unsupervised, Fast Correlation-Based Filter for Feature Selection for Data Clustering, pp. 87-94. *In* T. Herawan, M. M. Deris and J. Abawajy, eds. **Proceedings of the First International Conference on Advanced Data and Information Engineering (Daeng-2013).** Springer Singapore.

Pramokchon, P. and P. Piamsa-nga. 2014b. A Feature Score for Classifying Class-Imbalanced Data, pp. 433-428. *In* **The 18th International Computer Science and Engineering Conference (ICSEC) 2014**. Khon Kaen, Thailand

Pramokchon, P. and P. Piamsa-nga. 2014c. Reducing Effects of Class Imbalance Distribution in Multi-Class Text Categorization, pp. 263-272. *In* S. Boonkrong, H. Unger and P. Meesad, eds. **Recent Advances in Information and Communication Technology.** Springer International Publishing.

Rangarajan, L. and Veerabhadrappa. 2010. Bi-Level Dimensionality Reduction Methods Using Feature Selection and Feature Extraction. **International Journal of Computer Applications** 4 (2): 33-38.

Ruiz, R., J. Riquelme and J. Aguilar-Ruiz. 2005. Heuristic Search over a Ranking for Feature Selection, pp. 498-503. *In* J. Cabestany, A. Prieto and F. Sandoval, eds. **Computational Intelligence and Bioinspired Systems.** Springer Berlin / Heidelberg.

Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. **ACM Comput. Surv.** 34 (1): 1-47.

Seo, S. 2006. **A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets.** Master of Science, University of Pittsburgh.

Shamsinejadbabki, P. and M. Saraee. 2012. A New Unsupervised Feature Selection Method for Text Clustering Based on Genetic Algorithms. **Journal of Intelligent Information Systems** 38 (3): 669-684.

Shang, W., H. Huang, H. Zhu, Y. Lin, Y. Qu and Z. Wang. 2007. A Novel Feature Selection Algorithm for Text Categorization. **Expert Syst. Appl.** 33 (1): 1-5.

Somol, P., J. Novovicova and P. Pudil. 2010. Efficient Feature Subset Selection and Subset Size Optimization. **Pattern Recognition Recent Advances** 75-97.

Soucy, P. and G. Mineau. 2003. Feature Selection Strategies for Text Categorization, pp. 993-993. *In* Y. Xiang and B. Chaib-draa, eds. **Advances in Artificial Intelligence.** Springer Berlin / Heidelberg.

Tamhane, A.C. and D.D. Dunlop. 2000. **Statistics and Data Analysis.** Prentice Hall.

Uchyigit, G. and K. Clark. 2007. A New Feature Selection Method for Text Classification. **International Journal of Pattern Recognition and Artificial Intelligence** 21 (2): 423-438.

Yang, J., Y. Liu, X. Zhu, Z. Liu and X. Zhang. 2012. A New Feature Selection Based on Comprehensive Measurement Both in Inter-Category and Intra-Category for Text Categorization. **Information Processing & Management** 48 (4): 741-754.

Yang, Y. and J.O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization, pp. 412-420. *In* **14th International Conference on Machine Learning.** Morgan Kaufmann Publishers Inc.

Yanjun, L., L. Congnan and S.M. Chung. 2008. Text Clustering with Feature Selection by Using Statistical Data. **IEEE Transactions on Knowledge and Data Engineering** 20 (5): 641-652.

Yonghong, L. and A.K. Jain. 1998. Classification of Text Documents, pp. 1295-1297 vol.1292. *In* **Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on.**

Yu, L. and H. Liu. 2004. Efficient Feature Selection Via Analysis of Relevance and Redundancy. **J. Mach. Learn. Res.** 5 1205-1224.

Zheng, Z., X. Wu and R. Srihari. 2004. Feature Selection for Text Categorization on Imbalanced Data. **SIGKDD Explor. Newsl.** 6 (1): 80-89.

Zonghu, W., L. Zhijing, C. Donghui and T. Kai. 2011. A New Partitioning Based Algorithm for Document Clustering, pp. 1741-1745. *In* **Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on.**

CURRICULM VITAE

NAME **:** Mr. Part Pramokchon

BIRTH DATE **:** December 22, 1978

BIRTH PLACE **:** Lampang, Thailand

EDUCATION **:**

YEAR	**INSTITUTE**	**DEGREE/DIPLOMA**
1999	Chiang-Mai Univ.	B.Sc.(Computer Science)
2003	Chiang-Mai Univ.	M.S.(Computer Science)

POSITION/TITLE **:** Lecturer

WORK PLACE **:** Faculty of Science, Maejo University

SCHOLARSHIP/AWARDS **:** UDC scholarship

Publications

Pramokchon, P. and P. Piamsa-nga. 2014a. An Unsupervised, Fast Correlation-Based Filter for Feature Selection for Data Clustering, pp. 87-94. *In* T. Herawan, M. M. Deris and J. Abawajy, eds. **Proceedings of the First International Conference on Advanced Data and Information Engineering (Daeng-2013).** Springer Singapore.

Pramokchon, P. and P. Piamsa-nga. 2014b. Reducing Effects of Class Imbalance Distribution in Multi-Class Text Categorization, pp. 263-272. *In* S. Boonkrong, H. Unger and P. Meesad, eds. **Recent Advances in Information and Communication Technology.** Springer International Publishing.

Pramokchon, P. and P. Piamsa-nga. 2014c. A Feature Score for Classifying Class-Imbalanced Data, pp. 433-428. *In* **The 18th International Computer Science and Engineering Conference (ICSEC) 2014.** Khon Kaen, Thailand

Pramokchon, P. and P. Piamsa-nga. 2014d. Fast Feature Selection Method for High-Dimensional Datasets by Effective Threshold Estimation.
(To be submitted)

Pramokchon, P. and P. Piamsa-nga. 2014e. Content-Adaptive Feature Selection for Classifying Class-Imbalanced Data.
(To be submitted)