This article can be cited as W. Sharif, N. A. Samsudin, M. M. Deris and M. Aamir, Improved Relative Discriminative Criterion Feature Ranking Technique for Text Classification, International Journal of Artificial Intelligence, vol. 15, no. 2, pp. 61-78, 2017.


# Improved Relative Discriminative Criterion Feature Ranking Technique for Text Classification

**Wareesa Sharif, Noor Azah Samsudin, Mustafa Mat Deris, Muhammad Aamir**

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia (UTHM)
86400, Parit Raja, Johor, Malaysia
Email: wareesa786@gmail.com,azah@uthm.edu.my, mmustafa@uthm.edu.my, amirr.khan1@gmail.com

**ABSTRACT**

*Feature ranking techniques are used to improve the performance of classification in text labeling problems. Most of the feature selection techniques utilize document and term frequencies to rank term. In contrast to document frequency, term frequency support real values of the term. Recent feature ranking techniques use term frequencies with frequently occurring terms, but ignore rarely occurring terms which are as meaningful and important as frequently occurring terms. Moreover, F-measure decreases as features of existing techniques increases. In this paper, Improved Relative Discriminative Criterion (IRDC) technique is proposed to obtain more informative and meaningful rarely occurring terms. IRDC scale up rarely occurring terms that is present in one class and absent in other classes. Additionally, IRDC creates a trade-off between frequently and rarely occurring terms. Experimental results indicate that our proposed technique on reuters21578 and 20newsgroup datasets using well known classifiers like multinomial naïve bayes (MNB), support vector machine (SVM) and decision tree (DT) performed better in terms of F-measure.*

**Keywords:** Text classification, High dimensional data, Feature ranking, Document frequency, Term count, Rare terms, True positive rate, False positive rate.

## 1. INTRODUCTION

With the rapid growth of World Wide Web and electronic documents in digital format, classification becomes vital for organization to manage data (Dwivedi and Arya, 2016; Uysal, 2016). Classification techniques help to classify label from electronic documents such as news, blogs, e-mail and digital libraries (Mohod et al, 2015). Classification techniques have drawn awareness in many applications including image classification, face recognition, text clustering, spam filtering (Delany et al,2005 ; Metsi et al, 2006), email categorization (Kamens, 2005), website classification (Devi, 2008) and text classification (Rehman et al., 2015). Text classification is a challenging task in typical text documents because of ever-increasing amount of electronic documents, web recourses and digital libraries (Paul, 2014). That is why, text classification becomes essential task to label documents into predefined classes (Onan et al, 2016; Parlak and Uysal, 2016).

Text data is high dimensional data (Fragoudis et al, 2005; J. Yang et al, 2016) and this higher dimensionality of feature space impose weighty overhead to build document classifier, because some features can be redundant or irrelevant. These redundant features mislead the classification result (Javed et al, 2012). Therefore, feature ranking techniques are used to select most relevant and informative features, and to reduce the computational time (Xu et al, 2016; Zhan et al, 2016).

Existing feature ranking techniques such as chi-square (Manning et al, 2008; Yang and Pedersen, 1997) and information gain (Forman, 2003) consider document frequency (presence and absence) of term. They ignore the actual value of term in a document (Baccianella et al, 2013). Rehman et al., (2015) and Wang et al. (2015) redesigned the document frequency of term into document frequency for each term count to rank the term. Relative Discriminative Criterion (RDC) (Rehman et al., 2015) and Normalized Relative Discriminative Criterion (NRDC) (Wang et al., 2015) gave high rank to the frequently occurring terms but ignore the rarely occurring terms that are important to improve the accuracy of the classifiers. Moreover, (Sathiaseelan et al., 2015) agreed that (Rehman et al., 2015) concentrated on frequently occurring term, and F-measure of RDC decreases as number of terms increases.

To increase the performance and reduce the computational overhead, we did two modifications in previous algorithms; RDC and NRDC. First, this study considered rarely occurring terms in each class. Second, our concern was to reduce the computational complexity of the NRDC. In this paper, we propose feature ranking technique namely, Improved Relative Discriminative Criterion (IRDC). The proposed IRDC technique gives high rank to rarely occurring term in each class, because rarely occurring terms are meaningful and important for correct classification (Al-Tahrawi, 2013; Al-Tahrawi, 2014). In contrast to rarely occurring terms, frequently occurring terms get high rank in existing feature ranking techniques (e.g., RDC), which decrease their classification performance as number of features increases (Rehman et al., 2015). The proposed technique not only increases the classification performance but also decreases the complexity of existing techniques.

To check the performance of these algorithms for the best feature selection is comparison by using multinomial naïve bayes, decision tree and support vector machine. Experiment on the proposed IRDC technique was conducted using two datasets; Reuters21578 and 20newsgroups.The first key contribution of this paper is to redesign true positive and false positive rate of the term count in positive and negative classes to give high rank to rarely occurring terms in each class. Our proposed IRDC technique considers not only the document frequency (*df*), but also the term count (*tc*) to decide the rank of a term and increase the weight of the rare terms by dividing the summation of term frequency in each class. In IRDC, True Positive Rate (*TPR*) is the normalized document frequency in positive class and False Positive Rate (*FPR)* is the normalized document frequency in negative class. *TPR* and *FPR* are calculated for every term count in each class. The second contribution is to reduce the complexity of the NRDC.

The rest of the paper is organized as follows: Section 2 presents the related work. Subsequently, proposed IRDC technique is illustrated in Section 3. After that, experimental results are presented in Section 4 and results are discussed in section 5. In the end, conclusion is presented in section 6.

## 2. RELATED WORK

In this section, an overview of different feature ranking techniques is presented. Over a decade, feature ranking becomes a significant research area to improve classification accuracy in machine learning due to rapid growth in data collection and storage technologies. A feature selection algorithm can be seen as the combination of a search technique to select best features (Solos et al., 2016). If the dimension increases, the complexity of the dataset also increases because of non-informative and irrelevant features (Vergara and Estévez., 2014). To classify complex and high dimensional datasets is a challenging task for existing feature ranking techniques (García et al., 2016). Most of the techniques are based on document frequency in which presence and absence of term is considered in a document (Azam and Yao, 2012). A document is represented by multi-dimensional feature vectors in which each dimension corresponds to a weighted value such as (i.e., TF-IDF), but TFIDF is not using class information (Manning et al., 2008).

In text dataset, moderate number of text collection produces high dimensionality result in hundred and thousand number of features (Trivedi and Dey., 2016). The most important issue is to deal with high dimensionality feature space in text classification. In this regard, feature ranking in text classification is important to improve the precision, recall and F1-measure.

There are three types of feature selection methods: Filters, Wrapper and Embedded, and described as follows:

- Filter techniques evaluate every term independently according to the chosen weighting technique. It ranks the features after evaluation and takes the subset with the highest weight (Agnihotri et al., 2016; Precup et al., 2007).
- Wrapper algorithms depend on the chosen classifier. In this method, subsets of the initial terms are evaluated and subsequently best performance subset is selected (Gnana et al., 2016). Normally, Heuristic algorithms are used in wrapper for selecting features, but it is time consuming process (Kıran and Fındık, 2015).
- Embedded algorithms are used in classifiers (e.g., artificial neural networks) to select features during classification (Bhatia et al., 2015).

Wrapper method selects the ideal feature subset, while filter method select features on behalf of the score of individual feature. Firstly, filter methods computes the score for features then rank them (Forman, 2003; Yang and Pedersen, 1997). An ideal filter method gives high score to distinctive relevant feature and low score to irrelevant feature. Filter methods are popular than wrapper and embedded methods, because of low computation cost (Uysal and Gunal, 2012). In text classification, there are many filter methods such as information gain (Forman, 2003), chi-square (Manning et al., 2008; Yang and Pedersen, 1997) and odd ratio (Mengle and Goharian, 2009; Mladenic and Grobelnik, 1999) that can work with binary information (presence/absence) of term in training documents. In contrast to document frequency based methods, term frequency methods use the actual value of term (Baccianella et al., 2013; Uysal and Gunal, 2012). (Baccianella et al., 2013) claimed that existing feature selection techniques do not deem the term frequency (term count) to compute the rank of term. By using term frequencies, they logically break the document into "micro-documents" in which every micro-document contains one word (term).

A term appear frequently in one class and absent in other class is assigned high rank (Uysal and Gunal, 2012). As (Rehman et al., 2015) assigned high rank to frequently occurring term and redesign the document frequency of term into document frequency of each term count. To rank the term, (Rehman et al., 2015) consider the document frequency of term with its term count in positive and negative class. In positive class, normalized document frequency is true positive rate (*tpr*) and in negative class, normalized document frequency is false positive rate (*fpr*). Rehman et al., (2015) calculate the *tpr* or *fpr* for every term count. In large documents, the term count can be much bigger than short documents. Therefore, it generates bias result for large documents (Wang et al., 2015). By using the same criteria for feature ranking, (Uysal and Gunal, 2012) introduced Normalized Relative Discriminative Criterion (NRDC) in which they normalized the term count for long and short documents. RDC and NRDC both give high rank to frequently occurring terms and ignore rarely occurring terms. Our proposed feature ranking technique considers rarely occurring term as well as frequently occurring term, and it is explained in the following section.

## 3. PROPOSED TECHNIQUE

Some existing feature ranking methods (e.g., Chi-square) in text classification are based on document frequency (Forman, 2003; Yang and Pedersen, 1997), while others (e.g.,DFS ) rely on term frequency (Uysal and Gunal, 2012; Wang., 2014). Term frequency is the number of times a term appear in document and is more important than documents frequency because it consider the actual value of the term (Azam and Yao, 2012). Feature ranking techniques are used to select the important terms from the dataset. Existing feature ranking techniques (Rehman et al., 2015; Uysal and Gunal, 2012; Wang et al., 2015) used the frequency graph of term count for feature ranking methods to improve the classification performance. This study modified the criteria of (Uysal and Gunal, 2012) in order to propose IRDC for selecting the terms in classes, and the modified criteria is as follows:

- A term present frequently in one class and absent in all other classes are assigned high score.
- A frequently occurring term in all classes should be assigned a low score.
- A term appear rarely in one class and absent in other classes should be assigned relatively high score.
- A term present rarely in some of the classes should be assigned a relatively low score.

The pseudo code of the proposed algorithm is given below.

Step 1. Input: term frequency matrix of dataset
Step 2. Pos_frequency =calculate the no of documents for all term_count against term *t* in positive class
Step 3. Neg_frequency= calculate the no of documents for all term_count against term *t* in negative class
Step 4. Tcmax = maximum term_count for term *t*
Step 5. For *Tc*=1 to Tc_max do
Step 6. $Tp_{tc}$= term *t* appear in positive documents having term_count *tc*
Step 7. $Fp_{tc}$= term t appear in negative documents having term_count *tc*
Step 8. $TPR_{tc}$ =Tptc/pos_frequency
Step 9. $FPR_{tc}$=Fptc/neg_frequency
Step 10. IRDC= $[(TPR_{tc}-FPR_{tc})/ \min(TPR_{tc}, FPR_{tc})] * tc$
Step 11. End
Step 12. AUCt = 0
Step 13. For Tc=1 to tc_max do
Step 14. AUCt= AUCt + $(IRDC_{tc}+IRDC_{tc+1}) / 2$
Step 15. End
Output: final list of 1500 top selected features

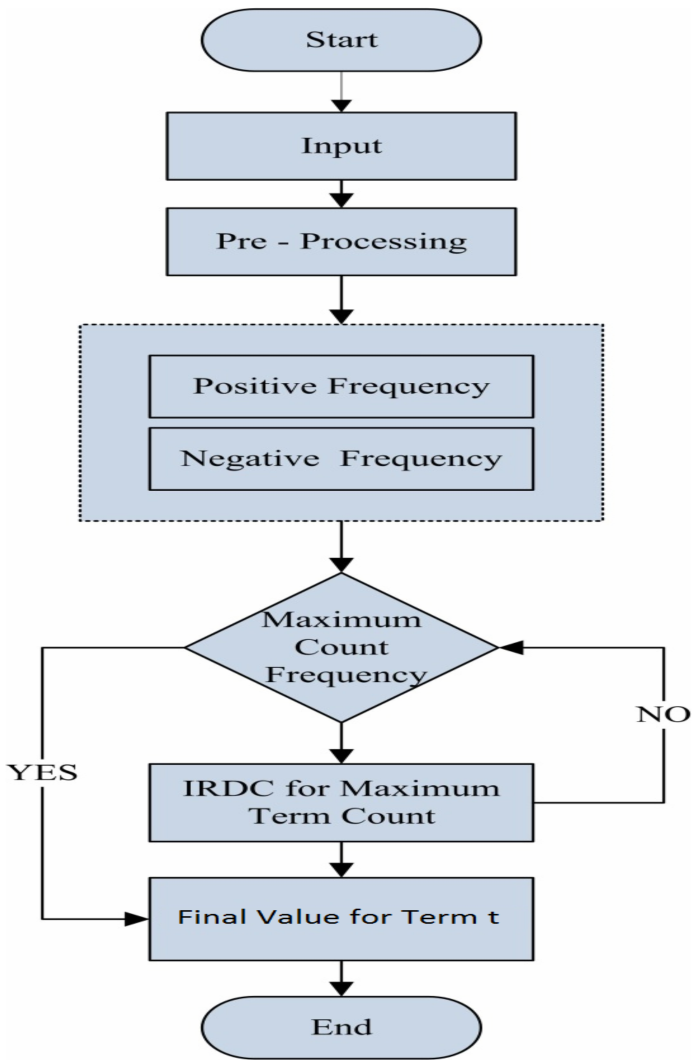


**Figure 1.** Flowchart for IRDC

Contrary to the studies conducted by (Rehman et al., 2015) and (Wang et al., 2015), this study considers rarely occurring terms in feature ranking, and assign high rank to rarely occurring terms which are important and meaningful in each class for correct classification. Subsequently, it minimizes the complexity of existing algorithm; NRDC. Whereas, (Rehman et al., 2015) and (Wang et al., 2015) redesign the document frequency as number of documents have term *t* in which its term count is *tc*. In positive class, normalized document frequency is presented by true positive rate (*tpr*) and in negative class shows as false positive rate (*fpr*) in RDC and NRDC, and it is shown as follows:

$$tpr = \frac{tp_{tc}}{doc\_in\_pos\_class} \tag{1}$$

where $tp_{tc}$ is term count in positive class. This *tpr* gave value to only frequently occurring terms. But if term occurs rarely, it gives low rank to rare terms by the dividing number of documents in positive class. By doing this, RDC and NRDC ignore the rare terms. These rare terms are important and meaningful to classify the documents into correct class that affect the performance of the classifier. Our proposed feature ranking technique creates trade-off between frequently and rarely occurring terms. In this way, the proposed technique does not ignore frequent terms, but relatively low down frequently occurring terms and scale up rarely occurring terms. Instead of divide the frequency of term count with the number of positive documents, we divide the document frequency of term count with the summation of documents frequency of term count in positive class to assign high rank to $TPR_{tc}$ for rarely occurring terms. However, equation (1) is replaced by:

$$TPR_{tc} = \frac{tp_{tc}}{\sum_{i=0}^{n} tc} \tag{2}$$

According to

$$fpr = \frac{fp_{tc}}{doc\_in\_neg\_class} \tag{3}$$

$fp_{tc}$ is term count in negative class. This *fpr* assign value to only frequently occurring terms in negative class. But for rare terms, it gives low score by dividing the number of documents in negative class. Consequently, it ignored the rarely occurring terms which are important to improve classification performance.

Since equation (3) ignores the rare terms in existing feature ranking techniques, this study scale up rare terms and relatively low down frequent terms. Instead of dividing the frequency of term count by the number of negative documents, we divide the document frequency of term count by the summation of documents frequency of term count in negative class as done in positive class for assigning high rank to rare terms. Consequently, equation (3) is replaced by:

$$FPR_{tc} = \frac{fp_{tc}}{\sum_{i=0}^{n} tc} \tag{4}$$

Existing techniques (e.g., RDC) ignore the rare terms due to dividing the difference of between $tpr_{tc}$ and $fpr_{tc}$ by the product of min($tpr_{tc}$, $fpr_{tc}$) and *tc*, as shown in the expression of RDC:

$$RDC = \frac{\left( \left| tpr_{tc} - fpr_{tc} \right| \right)}{\min\left( tpr_{tc}\ fpr_{tc} \right) * tc} \tag{5}$$

By doing this, these feature ranking techniques assign high rank to frequently terms and low rank to rarely occurring terms.

Contrary to RDC which low down the rank of rare terms and scale up the rank of frequent terms, modified RDC see equation (6) scale up rare terms and relatively scale down the rank of frequent terms. In doing so, IRDC selects all relevant features which are important to improve classification performance in terms of F-measure. For this purpose, IRDC multiply the term count (*tc*) with the division of difference between $TPR_{tc}$ and $FPR_{tc}$, and minimum of $TPR_{tc}$ and $FPR_{tc}$, which increases the rank of rarely occurring terms in a class. Consequently, equation (5) is replaced by:

$$IRDC = \frac{\left( \left| TPR_{tc} - FPR_{tc} \right| \right)}{\min\left( TPR_{tc}, FPR_{tc} \right)} * tc \tag{6}$$

Our proposed IRDC technique involves 4 steps in feature ranking, as follows:

(1) To compute document frequencies of the terms with term counts in in positive and negative classes.
(2) To calculate $TPR_{tc}$ and $FPR_{tc}$ in positive and negative classes.
(3) To calculate IRDC value for each term.
(4) To compute area under the curve (AUC) for each term.

The above four steps are illustrated with an example dataset. The example dataset with six documents and five unique terms; charger, keyboard, processor, LCD, and motherboard, is shown in Table 1. It is a balanced dataset where each class consists of three documents. Document frequencies for each term for different term counts in both classes are shown in Table 2. Depending on the document lengths, term counts for a term in different documents of a class range from one to a maximum value. Normally, lengthy documents have greater terms counts than smaller documents.

**Table1**. Example dataset with six documents and five unique terms

| Document | Class | Document content |
|---|---|---|
| Doc 1 | Positive | charger, keyboard, processor, processor |
| Doc 2 | Positive | processor, LCD, motherboard, LCD |
| Doc 3 | Positive | LCD, motherboard, charger |
| Doc 4 | Negative | charger, motherboard |
| Doc 5 | Negative | processor, charger, processor, motherboard |
| Doc 6 | Negative | processor, charger, charger, LCD, processor |

**Table 2.** Document frequency of the terms with term_count.

| Term | Charger | | LCD | | Motherboard | | Processor | | Keyboard | |
|---|---|---|---|---|---|---|---|---|---|---|
| Term count | Positi ve | Negati ve | Positi ve | Negati ve | Positi ve | Negati ve | Positi ve | Negati ve | Positi ve | Negati ve |
| 1 | 2 | 2 | 1 | 0 | 2 | 2 | 1 | 0 | 1 | 0 |
| 2 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

We calculated $TPR_{tc}$ and $FPR_{tc}$ for each term count in positive class and negative class, respectively. Whereas, $TPR_{tc}$ is calculated by dividing the document frequency of each term count with the summation of document frequencies for all term counts of term in positive class. Similarly, $FPR_{tc}$ is calculated by dividing the document frequency of each term count with the summation of document frequencies for all term counts of term in negative class. Table 3 presents calculation of $TPR_{tc}$ and $FPR_{tc}$ for each term count of term in positive and negative classes.

**Table 3.** Calculation of $TPR_{tc}$ and $FPR_{tc}$ in positive and negative classes.

| Term | Term Count | $TPR_{tc}$ in Positive Class | $FPR_{tc}$ in Negative Class |
|---|---|---|---|
| 1 | 1 | Charger_P= 2/(2+0+0)=1 | Charger_N=2/(2+1+0)=0.6667<br>Charger_N=1/(2+1+0)=0.3333 |
| 2 | 1 | LCD_P=1/(1+1+0)=0.5 | LCD_N= 0/(0+1+0)=0 |
| | 2 | LCD_P=1/(1+1+0)=0.5 | LCD_N= 1/(0+1+0)=1 |
| 3 | 1 | Mother board_P=2/(2+0+0)=1 | Mother board_N= 2/(2+0+0)=1 |
| 4 | 1 | Processor_P=1/(1+1+0)=0.5 | Procoessor_N= 0/(0+2+0)=0 |
| | 2 | Processor_P=1/(1+1+0)=0.5 | Procoessor_N= 2/(0+2+0)=1 |
| 5 | 1 | Keyboard_P= 1/(1+0+0)=1 | Keyboard_N=0/(0+0+0)=0 |

Table 4 shows IRDC values for different term counts of the term. If term appears rarely in one class, existing techniques gave low rank to that term but IRDC gave comparatively high rank to rare term. (Uysal & Gunal, 2012) describe that if a term present in one class, minimum document frequency of that term is zero and dividing the difference by zero leads to undefined number. To avoid division by zero, we divide the difference of between $TPR_{tc}$ and $FPR_{tc}$ by a small number ($\epsilon$). The value of small number ($\epsilon$) is 0.1 (Rehman et al., 2015; Wang et al., 2015). Moreover, term count is another important factor for determining term rank. Normally as term count increases, document frequency of the term count decreases and eventually fall to zero. By dividing a factor ($\epsilon$), difference of higher term counts will have more advantage than lower term counts. In order to give higher weight to difference of between $TPR_{tc}$ and $FPR_{tc}$, division of the difference by minimum is further multiplied by term count (*tc*). Loop will continue until max_term count find. After that, final value for term *t* is find through AUCt and algorithm stop which is shown in Figure1. In this way, the bias for rarely occurring term will be reduced.

**Table 4.** IRDC Calculations for Terms

| Term and Term Count | Positive ($Tpr_{tc}$) | Negative ($Fpr_{tc}$) | Difference (D) | Minimum (γ) | E | IRDC = (D/γ)*tc |
|---|---|---|---|---|---|---|
| Charger | | | | | | |
| *tc* 1 | 1 | 0.6667 | 0.3333 | 0.6667 | | (0.3333/ 0.6667)*1 = 0.4999 |
| *tc* 2 | 0 | 0.3333 | 0.3333 | 0 | 0.1 | (0.3333/0.1)*2 = 6.6665 |
| LCD | | | | | | |
| *tc* 1 | 0.5 | 0 | 0.5 | 0 | 0.1 | (0.5/0.1)*1 = 5 |
| *tc* 2 | 0.5 | 1 | 0.5 | 0.5 | | (0.5/0.5)*2 = 2 |
| Motherboard | | | | | | |
| *tc* 1 | 1 | 1 | 0 | 0 | 0.1 | (0/0.1)*1 = 0 |
| Processor | | | | | | |
| *tc* 1 | 0.5 | 0 | 0.5 | 0 | 0.1 | (0.5/0.1)*1 = 5 |
| *tc* 2 | 0.5 | 1 | 0.5 | 0.5 | | (0.5/0.5)*2 = 2 |
| Keyboard | | | | | | |
| *tc* 1 | 1 | 0 | 1 | 0 | 0.1 | (1/0.1)*1 = 10 |

In align with the prior studies conducted by (Rehman et al., 2015) and (Wang et al., 2015), we also consider area under the curve (AUC) for term rank. The term keyboard which is rarely occurring in one class gets the highest area under the curve (AUC). However, Figure 1 show steps involved in IRDC algorithm and to calculate AUC for term *t*.

AUC for charger = [(0.4999+6.6665)/2] + [(6.6665+0)/2] = 3.5829+3.3332 = 6.9161

AUC for LCD = [(5+2)/2] + [(2+0)/2] = 3.5+ 1= 4.5

AUC for processor = [(5+2)/2] + [(2+0)/2] = 3.5+1= 4.5

AUC for keyboard = [(0+10)/2] + [(10+0)/2] = 5 + 5= 10

## 4. EXPERIMENTAL SETUP

We conducted experiments using two benchmark datasets, namely reuter21578 and 20newgroup, which has been used in several past experimental studies (Albishre et al., 2015; Zong et al., 2015). These datasets are taken from UCI data repository in raw form. From reuters21578 dataset, 15 classes are used that are skewed in size. Another dataset 20newsgroup is a balanced dataset and has 20 large classes. Both datasets are single label datasets. Word-stemming is applied and also removes the stop words by using stop word list. For stemming procedure, porter stemmer (Karaa and Gribâa, 2013) is used to remove the too rare and too frequent terms. Feature ranking algorithm is written in Java platform and experiments are performed with three classifiers namely: Multinomial Naïve Bayes, Decision Tree and Support Vector Machine. Experiments are run on machine learning toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.7.11. It is an open-source platform that contains many machine learning algorithms implemented in JAVA. In WEKA toolkit, the default number of iterations to get statistically meaningful results is 10. In 10 cross validation, datasets are divided randomly into 10 mutually exclusive folds. Training process is used for 10 times and testing process also used for 10 times. The results of micro-averaged and macro-averaged are presented in Table 6 and Table 7.

### 4.1. Measuring Criteria

In text classification (Tang and Liu, 2005), realize that accuracy is not only criteria for measuring the performance of an algorithm. Whereas, precision, recall and micro F1- measure can also be used (Tang and Liu, 2005). Precision is computed in terms of

$$Precision = \frac{tp}{tp + fp} \tag{7}$$

whereas, *tp* denote the true positive rate and *fp* show the false positive rate in precision

Recall is computed in terms of

$$Recall = \frac{tp}{tp + fn} \tag{8}$$

whereas, *tp* describe the true positive rate and *fn* denote the false negative rate in recall.

F1-measure is harmonic mean of precision and recall (Forman, 2003). Micro-averaged of classes are computed by using precision and recall, but macro-averaged is computed as (Uysal and Gunal, 2012):

$$Macro\ average\ F1 = \frac{\sum_{j=1}^{c} \frac{2 * p_j * r_j}{p_j + r_j}}{C} \tag{9}$$

whereas, $p_j$ denote the precision and $r_j$ denote the recall of the $j^{th}$ class, and decision of all classes is divided by number of classes in macro-averaged

$$Micro\ average\ F1 = \frac{2 * p * r}{p + r} \tag{10}$$

whereas, *p* denote the precision and *r* denote the recall of the jth class. Decision of all classes is calculated in micro-averaged.

### 4.2. Rarely Occurring Terms

In text classification, rare terms are also important and meaningful which effect accuracy.

**Proposition:** IRDC rank features better than that of RDC and NRDC.

**Proof:** Precision and recall is directly proportional with *tp*. So, we can say that more *tp*, more F-measure. In contrast to RDC and NRDC which calculate *tpr* only for frequently occurring terms, IRDC calculate *TPR* not only for rarely occurring terms but also for frequently occurring terms. Whereas, *tpr* is denoted by true positive value of IRDC, NRDC and RDC. As per our experimental results shown in Figures 2. and 3, indicating that IRDC performs significantly better than NRDC and RDC in terms of F-measure. It is because that as compare to RDC and NRDC, IRDC generates more *tp*.

Hence, *tp*_IRDC > *tp*_NRDC and *tp*_IRDC > *tp*_RDC.

Therefore, F_IRDC > F_NRDC and F_IRDC > F_RDC

### 4.3. Computational Complexity

In text classification, computational complexity of NRDC and IRDC is checked on the number of iteration.

**Proposition:** Computational complexity of IRDC is lower than NRDC.

**Proof:** The computational complexity of NRDC is O(N2) to select 1500 top features from two text datasets. In the case of our proposed IRDC algorithm, complexity to select 1500 top features is O(2N) where N is the number of iterations, and O(2N) < O(N2).

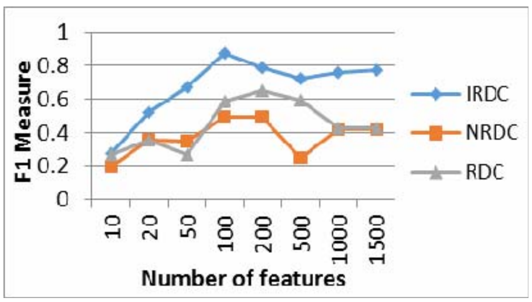
Therefore, computational complexity of IRDC < computational complexity of NRDC.

## 5. RESULT AND DISCUSSION

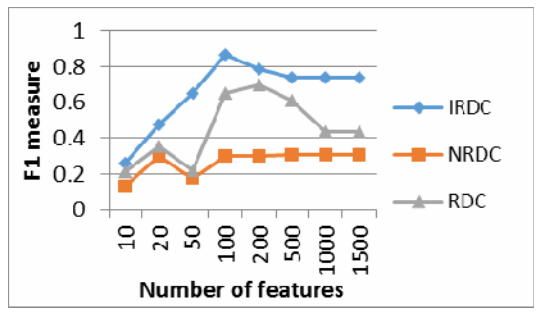
After performing the experiments, the results are compared with Relative Discriminative Criterion (RDC) and Normalized Relative Discriminative Criterion (NRDC). Performances of these feature ranking algorithms are examined on different number of features using two different datasets: Reuters21578 and 20Newsgroup. Series of experiments are conducted on top 10, 20, 50, 100, 200, 500, 1000, 1500 features selected from Reuter21578, and results are shown in Table 6 and Figure 2.

**Table 6.** Result of Reuter21578 dataset

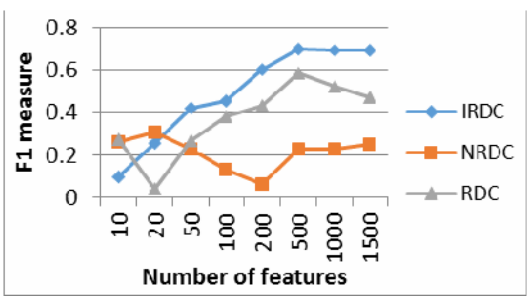
| Classifier | F-measure | Number of features | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 50 | 100 | 200 | 500 | 1000 | 1500 |
| SVM | Macro averaged | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| | Micro averaged | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| Multinomial naïve bayes | Macro averaged | RDC | NRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| | Micro averaged | RDC, NRDC | IRDC, NRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| Decision tree | Macro averaged | NRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| | Micro averaged | NRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |



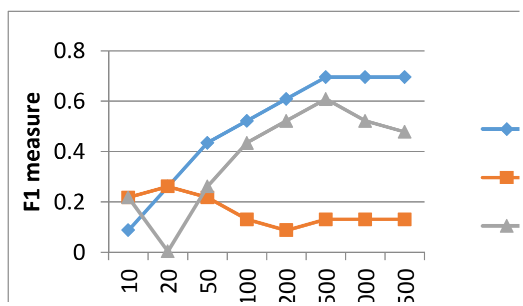

(a) Macro-averaged**(**Support Vector Machine**)**



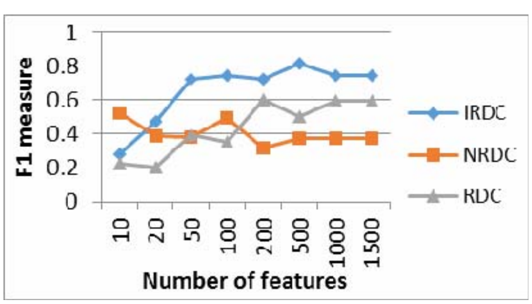

(b) Micro-averaged **(**Support Vector Machine**)**



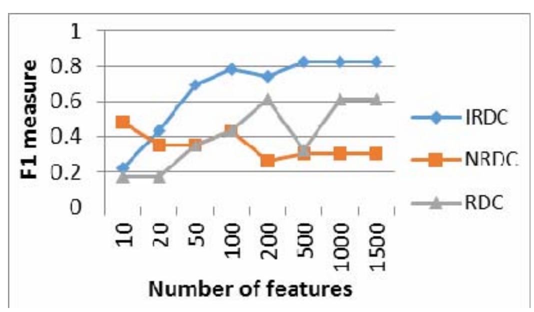

(c) Macro-averaged(Multinomial Naïve Bays)




(d) Micro-averaged(Multinomial Naïve Bays)




(e) Macro-averaged (Decision Tree)




(f) Micro-averaged (Decision Tree)

**Figure 2.** Result for dataset Reuter21578

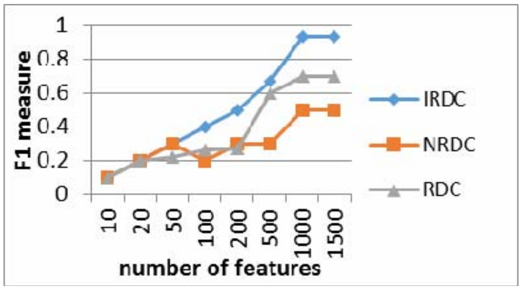
This experiment shows that as number of features increases, F-measure of IRDC also increases. For reuters21578 dataset, proposed IRDC gave 69.56% micro-averaged and 69.54% macro-averaged with multinomial naïve bayes, while RDC produced 48% micro-averaged and 47.30% macro-averaged, and NRDC generated 42% macro-averaged and 31% micro-averaged on reuter21578 dataset with top 1500 features, as shows in Figure 2 (c) and (d). Table 6 presents the number of time IRDC produced good result in case of reuters21578 dataset for micro-averaged and macro-averaged.

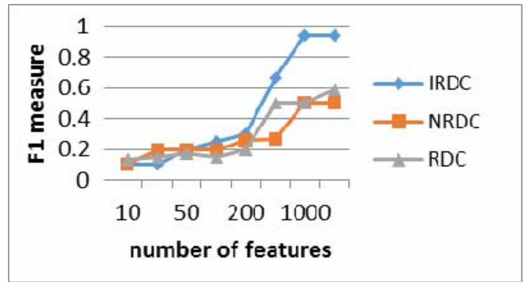
For multinomial naïve bayes, IRDC shows good result than RDC and NRDC for top 50,100, 200, 500, 1000, 1500 features. For 10 features only, result of micro-averaged for NRDC and RDC are same. In case of macro-averaged for 10 features, RDC shows good result. For micro-averaged for top 20 features, result of NRDC and IRDC are equal but for macro-averaged result of NRDC is better. Overall result of our proposed "Improved Relative Discriminative Criterion (IRDC)" is better than that of NRDC and RDC. In case of support vector machine classifier, IRDC produced 73.91% micro-averaged and 77% macro-averaged while RDC produced 43.47% micro-averaged and 42.92% macro-averaged with Reuter dataset. NRDC produced 31% micro-averaged and 42% macro-averaged. For top term 10, 20, 50, 100, 200, 500, 1000, 1500, IRDC gave better result than that NRDC and RDC present in Figure 2 (a) and (b).

IRDC with decision tree shows 82.60% micro-averaged and 74.39% macro-averaged, while RDC produced 60.86% micro-averaged and 59.66% macro-averaged, and NRDC shows 30.43% micro-averaged and 37.85% macro-averaged with top 20, 50, 100, 200, 500, 1000, 1500 features. F-measure shows in Figure 2 (e) and (f). For top 10 features only, NRDC result is better. Overall result of IRDC is better than NRDC and RDC. In case of NRDC and RDC, when the number of features is increasing, the performance of the classifier is decreasing. For IRDC, numbers of features are increasing; the performance of classifier is also increasing. However, IRDC is powerful technique to choose important features form huge data. Second experiment is done on 20newsgroup dataset which can be seen in Table 7 and Figure 3.

**Table 7:** Result of 20Newsgroup dataset

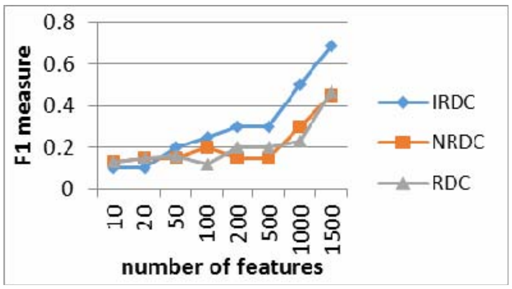
| Classifier | F1 Measure | Number of Features | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 50 | 100 | 200 | 500 | 1000 | 1500 |
| SVM | Macro averaged | NRDC | NRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| | Micro averaged | NRDC | NRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| Multinomial naïve bayes | Macro averaged | RDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| | Micro averaged | RDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| Decision tree | Macro averaged | RDC | RDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |
| | Micro averaged | RDC | RDC | IRDC | IRDC | IRDC | IRDC | IRDC | IRDC |

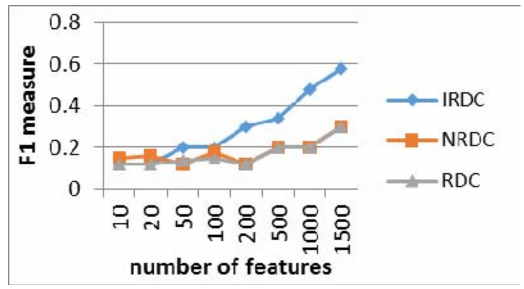



(a) Macro-averaged (Multinomial Naïve Bayes)

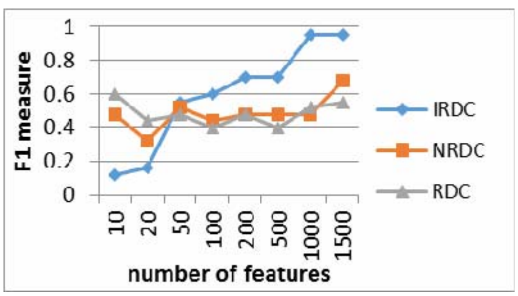
(b) Micro-averaged (Multinomial Naïve Bayes)

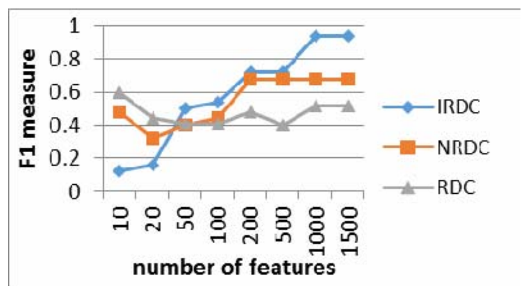



(c) Macro-averaged(Support Vector Machine)

(d) Micro-averaged (Support Vector Machine)




(e) Macro-averaged (Decision Tree)

(f) Micro-averaged (Decision Tree)

**Figure 3.** Result of 20 newsgroup dataset

For 20newsgroup dataset, top 1500 features are selected in which multinomial naïve bayes present high F1-measure result for 20, 50, 100, 500, 1000, 1500 features. For 10 features only, RDC perform better. Result of IRDC using multinomial naïve bayes is micro-averaged 92% and macro-averaged 93.60%. Result of NRDC as micro-averaged 50% and macro-averaged 50%, while RDC produced 59% micro-averaged and 70% as macro-averaged. IRDC generated high micro-averaged and macro-averaged than that of NRDC and RDC, as shown in Figure 3 (a) and (b).

By using support vector machine, IRDC produced micro-averaged 58% and macro-averaged 69% while RDC produced micro-averaged 30% and macro-averaged 47%. NRDC produced micro-averaged 30% and macro-averaged 45%. Additionally, performance for NRDC is better only for 10 and 20 features, when support vector machine is used for classification. For 50, 100, 200, 500, 1000, 1500 features, IRDC produced better micro-averaged and macro-averaged than that of RDC and NRDC, as shown in Figure 3 (c) and (d).

By using decision tree, top 1500 features are used for classification. IRDC generated micro-averaged 94% and macro-averaged 95%, whereas RDC produced micro-averaged 52% and macro-averaged 55% for 20newsgroup dataset. NRDC produced micro-averaged 68% and macro-averaged 68.8%. RDC result is better only for first 10 and 20 features, but for 50,100,200,500,1000,1500, IRDC produced best result. We also observed general behaviour of IRDC for top 1500 features for 20newsgroup dataset, which is higher than that of NRDC and RDC, as illustrated in Figure 3 (e) and (f).

## 6. CONCLUSION

In machine learning algorithm, high dimensionality in text data is a challenging problem. Feature ranking is a technique that is used to reduce the features that are not important for classification. Previous feature ranking techniques select frequently occurring terms and ignore rarely occurring terms that can be meaningful and significant for correct classification. Since, our experimental results showed that IRDC gives performance better than RDC and NRDC, in terms of micro-averaged and macro-averaged F-measures. Also IRDC is more convergence than the existing techniques, as each iteration minimizes the solution set and getting near to the final result. However, it is indicated that rarely occurring terms are as important as frequently occurring terms in each class to improve the performance of a classifier in text classification. The results also showed that our proposed technique minimizes the complexity of normalize relative discriminative criterion (NRDC) algorithm.

As a future work, we will evaluate efficiency of IRDC on different other datasets (e.g., medical dataset). The IRDC opens a new direction for incorporating rarely occurring term counts for the calculation of term rank. In future, we will further investigate how to group the term counts more effectively to determining the term rank. We will also work on modifications needed for the application of IRDC on non-text datasets, and compare performance of IRDC with other feature ranking technique on non-text datasets.

## ACKNOWLEDGEMENT

The support of Malaysia Ministry of Education Fundamental Research under Grant Scheme Vot 1609 is acknowledged.

## REFERENCES

Agnihotri, D., Verma, K., & Tripathi, P., 2016. *Computing symmetrical strength of N-grams: a two pass filtering approach in automatic classification of text documents.* SpringerPlus, **5**, 942.

Al-Tahrawi, M. M., 2013. *The role of rare terms in enhancing the performance of polynomial networks based text categorization*. Journal of Intelligent Learning Systems and Applications. **5**, 84-89.

Al-Tahrawi, M. M., 2014. *The significance of low frequent terms in text classification.* International Journal of Intelligent Systems, **29**, 389-406.

Albishre, K., Albathan, M., & Li, Y., 2015. *Effective 20 newsgroups dataset cleaning.* International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT).

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

Baccianella, S., Esuli, A., & Sebastiani, F., 2013. *Using micro-documents for feature selection: The case of ordinal text classification.* Expert Systems with Applications, **40**, 4687-4696.

Bhatia, K., Jain, H., Kar, P., Varma, M., & Jain, P., 2015. *Sparse local embeddings for extreme multi-label classification*. Advances in Neural Information Processing Systems, 730-738.

Delany, S. J., Cunningham, P., & Coyle, L., 2005. *An assessment of case-based reasoning for spam filtering.* Artificial Intelligence Review, **24**, 359-378.

Devi, M. I., Rajaram, R., & Selvakuberan, K., 2008. *Generating best features for web page classification*. Webology, **5**.

Dwivedi, S. K., & Arya, C. 2016. *Automatic text classification in information retrieval: A survey.* In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, ACM, 131.

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

Fragoudis, D., Meretakis, D., & Likothanassis, S., 2005. *Best terms: an efficient feature-selection algorithm for text categorization.* Knowledge and Information Systems, **8**, 16-33.

García-Torres, M., Gómez-Vela, F., Melian-Batista, B., & Moreno-Vega, J. M., 2016. *High-dimensional feature selection via feature grouping: A variable neighborhood search approach.* Information Sciences, **326**, 102-118.

Gnana, D. A. A., Appavu, S., & Leavline, E. J., 2016. *Literature Review on Feature Selection Methods for High-Dimensional Data.* International Journal of Computer Applications, **136**, 9-17.

Javed, K., Babri, H. A., & Saeed, M., 2012. *Feature selection based on class-dependent densities for high-dimensional binary data*. IEEE Transactions on Knowledge and Data Engineering, **24**, 465-477.

Jiang, S., Pang, G., Wu, M., & Kuang, L., 2012. *An improved K-nearest-neighbor algorithm for text categorization.* Expert Systems with Applications, **39**, 1503-1509.

Kamens, B., 2005. *Bayesian filtering: Beyond binary classification.* Fog Creek Software, Inc.

Karaa, W. B. A., & Gribâa, N., 2013. *Information retrieval with porter stemmer: a new version for English.* Advances in Computational Science, Engineering and Information Technology, Springer, Heidelberg, **225**, 243-254.

Kıran, M. S., & Fındık, O., 2015. *A directed artificial bee colony algorithm.* Applied Soft Computing, **26**, 454-462.

Lichman, M., 2013. *UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups]*. Irvine, CA: University of California, School of Information and Computer Science.

Lichman, M., 2013. *UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection].* Irvine, CA: University of California, School of Information and Computer Science.

Manning, C. D., Raghavan, P., & Schütze, H., 2008. *Introduction to information retrieval.* Cambridge university press Cambridge.

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

Metsis, V., Androutsopoulos, I., & Paliouras, G., 2006. *Spam filtering with naive bayes-which naive bayes?*. In CEAS, **17**, 28-69.

Mladenic, D., & Grobelnik, M., 1999. *Feature selection for unbalanced class distribution and naive bayes.* In ICML.

Mohod, S. W., Dhote, C. A., & Thakare, V. M., 2015. *Modified Approach of Multinomial Naïve Bayes for Text Document Classification*. International Journal of Computer Science & Communication, 196-200.

Onan, A., Korukoglu, S., & Bulut, H., 2016. *Ensemble of keyword extraction methods and classifiers in text classification*. Expert Systems with Applications, **57**, 232-247.

Parlak, B., & Uysal, A. K., 2016*. The impact of feature selection on medical document classification.* In Proceedings of 11th Iberian Conference on Information Systems and Technologies (CISTI), 2016. 1-5.

Paul, A., 2014. Effect of imbalanced data on document classification algorithms. *Auckland University of Technology*.

Rehman, A., Javed, K., Babri, H. A., & Saeed, M., 2015. *Relative discrimination criterion–A novel feature ranking method for text data.* Expert Systems with Applications, **42**, 3670-3681.

Precup, R. E., Preitl, S., & Korondi, P. (2007). *Fuzzy controllers with maximum sensitivity for servosystems*. IEEE Transactions on Industrial Electronics, **54**, 1298-1310.

Sathiaseelan, J., 2015. *A technical study on Information Retrieval using web mining techniques.* In Proceedings of 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 1-5.

Solos, I. P., Tassopoulos, I. X., & Beligiannis, G. N. (2016). *Optimizing shift scheduling for tank trucks using an effective stochastic variable neighbourhood approach.* International Journal of Artificial Intelligence, **14**(1), 1-26.

Tang, L., & Liu, H., 2005. *Bias analysis in text classification for highly skewed data.* In Proceedings of Fifth IEEE International Conference on Data Mining.

Trivedi, S. K., & Dey, S., 2016. *A Comparative Study of Various Supervised Feature Selection Methods for Spam Classification.* In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies.

Uysal, A. K., 2016. *An improved global feature selection scheme for text classification.* Expert Systems with Applications, **43**, 82-92.

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

Vergara, J. R., & Estévez, P. A., 2014. *A review of feature selection methods based on mutual information.* Neural Computing and Applications, **24**, 175-186.

Wang, D., Zhang, H., Liu, R., Lv, W., & Wang, D., 2014. *T-test feature selection approach based on term frequency for text categorization.* Pattern Recognition Letters, **45**, 1-10.

Wang, F., Zhang, Y., Xiao, H., Kuang, L., & Lai, Y., 2015. *Enhancing stock price prediction with a hybrid approach based extreme learning machine.* In Proceedings of IEEE International Conference on Data Mining Workshop.

Xu, J., Tang, B., He, H., & Man, H., 2016. *Semisupervised feature selection based on relevance and redundancy criteria.* IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2016.2562670.

Yang, J., Liu, Z., & Qu, Z., 2016. *A Novel Feature Selection Based Gravitation for Text Categorization.* International Journal of Database Theory and Application, **9**, 211-228.

Yang, Y., & Pedersen, J. O., 1997. *A comparative study on feature selection in text categorization.* In ICML.

Zhang, L., Jiang, L., & Li, C., 2016. *A new feature selection approach to naive Bayes text classifiers.* International Journal of Pattern Recognition and Artificial Intelligence, **30**, 1650003.

Zong, W., Wu, F., Chu, L.-K., & Sculli, D., 2015. *A discriminative and semantic feature selection method for text categorization.* International Journal of Production Economics, **165**, 215-222.