N-grams based feature selection and text representation for Chinese Text Classification

Article *in* International Journal of Computational Intelligence Systems · December 2009

DOI: 10.2991/ijcis.2009.2.4.5

5 authors, including:

Zhihua Wei
Tongji University
32 PUBLICATIONS **183** CITATIONS

SEE PROFILE

Jean-Hugues Chauchat
University of Lyon
47 PUBLICATIONS **381** CITATIONS

SEE PROFILE

International Journal of Computational Intelligence Systems, Vol.2, No. 4 (December, 2009), 365-374

N-grams based feature selection and text representation for Chinese Text Classification

Zhihua WEI [1,2,3], Duoqian MIAO [1,2],
Jean-Hugues CHAUCHAT [3], Rui ZHAO [1], Wen LI [1,2]

[1] Department of Computer Science and Engineering, Tongji University,
Cao'an Road, 4800
Shanghai,201804, China

E-mail: zhihua.wei@hotmail.com,miaoduoqian@163.com,zhaorui1@126.com

[2] Key laboratory "Embedded System and Service Computing" Ministry of Education, Tongji University ,
Cao'an Road,4800
Shanghai,201804, China

[3]Université de Lyon, Laboratoire ERIC-Lyon2 ,
avenue Pierre Mendès-France, 5
Bron Cedex,69676,France

E-mail: jean-hugues.chauchat@univ-lyon2.fr

Received: 30/12/08
Accepted: 28/05/09

Abstract

In this paper, text representation and feature selection strategies for Chinese text classification based on n-grams are discussed. Two steps feature selection strategy is proposed which combines the preprocess within classes with the feature selection among classes. Four different feature selection methods and three text representation weights are compared by exhaustive experiments. Both C-SVC classifier and Naive bayes classifier are adopted to assess the results. All experiments are performed on Chinese corpus TanCorpV1.0 which includes more than 14,000 texts divided in 12 classes. Our experiments concern: (1) the performance comparison among different feature selection strategies: absolute text frequency, relative text frequency, absolute n-gram frequency and relative n-gram frequency; (2) the comparison of the sparseness and feature correlation in the "text by feature" matrices produced by four feature selection methods; (3) the performance comparison among three term weights: 0/1 logical value, n-gram frequency numeric value (TF) and Tf*idf value.

Keywords: Chinese text classification, n-gram, feature selection, text representation weight.

1. Introduction

With the rapidly increasing quantity of web sources and electronic texts in Chinese, much attention has been paid to the Chinese text classification (TC). In addition to some difficulties in text classification in English, Chinese TC exhibits the following difficulties: (1) there is no space between words in Chinese text. (2) There is no punctuation mark (word endings). (3) There are 20,000 to 50,000 characters fre-

quently used in Chinese, which are much more than the number of characters used in English. The problem of Chinese TC is difficult and challenging.

The great difference between Chinese TC and Latin languages TC lies in the text representation. In a TC task, the term can be a word, a character or a n-gram. These features play the same role in Chinese TC. However, unlike most of western languages, Chinese words do not have a remarkable boundary. This means that the word segmentation is necessary before any other preprocessing. The use of a dictionary is necessary. Word sense disambiguation issue and unknown word recognition problem limit the precision of word segmentation.

For example, the sentence " *物理学起来很难。(Physics is difficult.)"* can be segmented as two kinds of forms:

物理 / 学 / 起来 / 很 / 难 /。(right)
Physics/ study / up / very / difficult.
物理学 / 起来 / 很 / 难。(error)
Physics / up / very / difficult.

Word sense disambiguation (WSD) is one of the most important and complex processes in NLP. The fact that a word can have multiple meanings, as well as the presence of unknown words in a text, make the segmentation a difficult task. In addition, many unknown words are closely related to the document theme. For example, in sentence *"流感到冬天很普遍。(The Flu is common in winter.)"*, "流感" is a abbreviation of a disease which is one of unknown word, whereas, *"感到"* is a word in dictionary. This sentence may have two kinds of segmentations. But only in the first form, the word "flu" (which indicates that the document belongs to medical field) can be recognized.

流感 / 到 / 冬天 / 很 / 普遍 /。(right)
Flu / arrive /winter / very / common.
流 / 感到 / 冬天 / 很 / 普遍 / 。(error)
Flow/ feel /winter / very / common.

Even if we could get a correct segmentation, the same word may have multiple meanings in different contexts. For example, *"板块"* has the different meanings in the following two sentences.

太平洋 / 板块 / 很 / 活跃 /。
(Pacific plate is very active.)
只有 / 两 / 个 / 板块 / 的 / 股票 / 上涨 /。
(Shares of only 2 blocs rose.)

In first sentence, *"板块"(plate)* may be a feature of texts in geographic class, while *"板块"(blocs)* means a symbol of texts in economy class.

In fact, there are few studies concerning the relation between the improvement in precision of Chinese word segmentation and corresponding results of Chinese text classification.

Usually, there are two steps in the construction of an automated text classification system. The first step is that the texts are coded into a representation more suitable for the learning algorithm. There are various ways of representing a text such as by using word fragments, words, phrases, meanings, and concepts [1]. Different text representations have different dependence on the language of the text. The second step is concerned by choosing the learning algorithm. In this paper, we focus on the first step. We represent texts with character n-grams which is a method independent of languages. That is, it avoids complicated word segmentation process in Chinese TC. We will discuss different text representations and feature selection strategies in Chinese TC based on n-grams in following parts.

The organization of this paper is as follow. In section 2, related work in Chinese TC based on n-grams is reviewed. In section 3, the text representation methods in our work are introduced. In section 4, term preprocessing method within classes and the feature selection method among classes are presented. In section 5, various experiments are shown for comparing different feature selection methods, different text representation weights and so on. Conclusions are given in section 6.

2. Related work in Chinese TC based on n-grams

A character n-gram is a sequence of *n* consecutive characters. The set of n-grams (usually, *n* is set to 1, 2, 3 or 4) that can be generated for a given document is basically the result of moving a window of *n* characters along the text. The window moves one character at a time. Then, the number of occurrences of each n-gram is counted [2].

There are several advantages of using n-grams in

TC tasks [3]. One of them is that by using n-grams, we do not need to perform word segmentation. In addition, no dictionary or language specific techniques are needed. However, n-gram extraction on a large corpus will yield a large number of possible n-grams. Only some of them will have discriminating frequency values in vectors representing the texts and good discriminate power. As a result, feature dimension reduction becomes more important for TC task based on n-grams.

Previous research mainly focuses on the value of "n". Lelu et al. discussed $n = 2$, namely, only using 2-gram to represent Chinese text because they regarded that most Chinese words are composed of two characters[3]. Zhou et al. gave more detailed experiments by using respectively 1-gram, 2-gram, 1-, 2-gram, 2, 3, 4-gram and 1, 2, 3, 4-gram as items and gave the conclusion that the best result is obtained by using 1-, 2-, 3-, 4-grams, the second best is obtained by using 1-,2-grams, the third best is obtained by using 2-grams, the case of using 2-, 3-, 4-grams follow and the worst one is obtained by using 1-grams[4]. Wei et al. compared various cases by using n-grams as text representation, for example, 0/1 weight and frequency weight[5]. They also analyzed different feature selection methods[6].

All of these works got some useful conclusions. There are still many problems to solve. Compared with famous weight Tf*idf, what about the performance of TF weight? What are the feature distributions when using different feature selection strategies? How do n-grams affect the miss-classified texts? This paper aims at solving above problems.

3. Text Representation Using Various Weights

We adopt the VSM (Vector Space Model), where each document is considered as a vector in the feature space. Thus, given a set of N documents, d_1, d_2...d_N, the table of "document by term" is constructed shown in table 1, where T_i is n-gram and each document is represented by a score "w_{ij}". Generally, "w_{ij}" has be any of the following: a dot:

- w_{ij} = frequency of term j in document i, that is, TF;
- w_{ij} = 0 or 1, $w_{ij} = 1$, if term j appears in document i, otherwise, w_{ij}=0.
- $w_{ij} = tf_{ij} \times idf_j$, where:
  $tf_{ij} = \frac{n_{ij}}{\sum_k n_{ij}}$, with n_{ij} is the number of occurrences of the considered term in document d_j, and the denominator is the number of occurrences of all terms in document d_j.
  $idf_i = \log \frac{|D|}{\left|\{d_j : t_i \in d_j\}\right|}$, with $|D|$ is the total number of documents in the corpus and $\left|\{d_j : t_i \in d_j\}\right|$is the number of documents where the term t_i appears.

Three kinds of weight "w_{ij}" are compared in this paper.

Table 1. "Document by feature" vector table

	T_1	T_2	...	T_j	...	T_M	Class
d_1	w_{11}	w_{12}	...			w_{1M}	A
d_2	w_{21}	w_{22}	...			w_{2M}	B
...		...					...
d_i	w_{i1}	w_{i2}	...	w_{ij}		w_{iM}	C
...			...				...
d_N	w_{N1}	w_{N2}	...			w_{NM}	A

4. Two Steps in Feature Selection

Feature selection is a space reduction method which attempts to select the more discriminant features from preprocessed documents in order to improve classification quality and reduce computational complexity. As many n-grams are extracted from Chinese texts, we perform two steps of feature selection. In the first step, we reduce the number of features within classes. In the second step, we choose the most discriminant features among all classes in the training set.

We extract the 1-, 2-grams in the texts of the corpus and divide the corpus into the training set and the testing set. In our work, 70% texts in each class are selected randomly to constitute the training set and the 30% left are used for the testing set. The two steps in feature selection are performed only on training set.

4.1. Some definitions

In text classification, the text is usually represented as a vector of weighted features. The difference between various text representations comes from the definition of "feature". This work explores four kinds of feature building methods which are defined as follow.

In the training set, each text in corpus D belongs to one class c_i. Here, $c_i \in C$, $C = \{c_1, c_2 ... c_i ... c_n\}$, C is the class set defined before classification. a dot:

- Absolute text frequency is noted as $Text\_freq_{ij}$, which is the number of texts that include n-gram j in class c_i;
- Relative text frequency is noted as $Text\_freq\_relative_{ij}$, which is got from $Text\_freq_{ij}/N_i$, here, N_i is the quantity of texts in class c_i in training set;
- Absolute n-gram frequency is noted as $Gram\_freq_{ij}$, which is the number of n-gram j in all texts in class c_i in training set;
- Relative n-gram frequency is noted as $Gram\_freq\_relative_{ij}$, which is got from $Gram\_freq_{ij}/N'_i$, here, N'_i is the total number of occurrence of all n-grams in all texts in class c_i in training set.

4.2. Term preprocessing within class

We can extract the 1-, 2-grams in the texts of the training set. However, the n-grams frequency in each class is greater than 15,000 in average. Most of them occur only one or two times. It is necessary to cancel some features in a class before further feature selection. We adopt relative text frequency method to reduce the number of terms in a class which gives better results than the method using absolute frequency [6]. Algorithm 1 describes this process.

Algorithm 1.

Begin

For $c_i \in C$, $C = \{c_1, c_2 ... c_i ... c_n\}$,

$Term'_i = \emptyset$, $Term = \emptyset$;

For $n-gram_j \in Term_i$,

If $Text\_freq\_relative_{ij} > \alpha$, then $n-gram_j \in Term'_i$.

$Term = \{Term'_1, Term'_2 ... Term'_i ... Term'_n\}$.

End.

Here, $Term_i$ includes all the n-grams extracted in the class c_i , $Term'_i$ includes all the n-grams selected in the class c_i and $Term$ is n-gram set in all classes selected by Algorithm 1. We choose $\alpha = 0.02$ as the threshold in order to keep features as many as possible in each class. After this selection, there are 7000 features in each class in average which are enough for text classification task. In the case of $Text\_freq\_relative_{ij} > 0.03$, there are only 4,000 features left in each class in average. It is not enough for the second step.

4.3. Feature selection among classes

There are many methods for feature selection. Yang concluded some methods based on statistic [7]. There are also some methods considered text semantic information[8,9].The choice of the feature selection method in this work is the CHI-Square test which is often cited as one of the best methods for the feature selection[7,10]. It gives a similar result as Information Gain because it is numerically equivalent[11].

In this work,we construct "feature by class" matrix (noted as $Matrix_{cf}$) by Algorithm 2 to select discriminant features. In $Matrix_{cf}$, each feature "j" is assigned to a numeric score based on its occurrence within the different document classe c_i.According to CHI-Square algorithm, the score of n-gram "j" is:

$$\sum_i \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{1}$$

Where "i" is the class, "j" is the n-gram and O_{ij} is the observed value. E_{ij} represent the expectation value in the hypothesis of independence of classes and features:

$$E_{ij} = \frac{O_{i+} * O_{+j}}{O_{++}} \tag{2}$$

Here, we define four kinds of values on O_{ij} described in Algorithm 2. Their performance will be compared in different experiment scenarios in Section 5. According to the result of CHI-Square, we perform the classification using the 200, 500, 800, 1000, 2000... 5000 features respectively.

Algorithm 2.
Begin
For $c_i \in C, C = \{c_1, c_2 ... c_i ... c_n\}$,
For $n-gram_j \in Term$,
If $n-gram_j \notin Term_i$, $\{O_{ij}\} = 0$
Else $\{O_{ij} = Text\_freq_{ij}$ or $Text\_freq\_relative_{ij}$ or $Gram\_freq_{ij}$ or $Gram\_freq\_relative_{ij}\}$,
End.

5. Experiment

5.1. Experiment Setting

We use TanCorp-12 corpus, a collection of 14,150 texts in Chinese language. It has been collected and processed by Songbo Tan [12]. It contains 12 categories (art, car, career, computer, economy, education, entertainment, estate, medical, region, science and sport)as shown in Table 2. The largest class contains 2865 texts (4.17M) and the smallest class contains 150 texts (0.49M).

Table 2. Distribution of TanCorp-12 (M= megabyte).

Class name	Num of texts	Size of class
Art	546	1.42M
Entertainment	1500	2.89M
Car	590	0.89M
Estate	935	1.80M
Career	608	1.78M
Medical	1406	2.64M
Computer	2865	4.17M
Region	150	0.49M
Economy	819	2.60M
Science	1040	1.97M
Education	808	1.41M
Sport	2805	4.20M

In this paper, we perform experiments using two classifiers: Naive Bayes and C-SVC. C-SVC classifier which was introduced in LIBSVM[13] is the extension of SVM algorithm for the multi-classification tasks. For C-SVC, irrelevant attributes weakly disturb the learning process. That is, the effect of different feature selections and different text representation weights can be obvious. Learning parameters are set to a linear kernel, gamma=0 and penalty cost=1. We conduct our experiments in the platform TANAGRA which is a free data mining software for academic and research purposes developed by Ricco Rakotomalala[14].

We use the F1 measure which combines recall and precision in the following way for the bi-class case[15].

$$Recall = \frac{number\ of\ correct\ positive\ prediction}{number\ of\ positive\ examples} \quad (3)$$

$$Precision = \frac{number\ of\ correct\ positive\ prediction}{number\ of\ positive\ predictions} \quad (4)$$

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \quad (5)$$

For more than two classes, the F1 scores are summarized over the different categories using the Micro-averages and Macro-averages of F1 scores. a dot:

- Micro - F1 = average in documents and classes
- Macro - F1 = average of within - category F1 values

5.2. Experiments on Comparison Four methods of Feature Selection

1-, 2-gram combination has better performance than 1-, 2-, 3-gram combination[6]. Consequently, we set our experiments by comparing four kinds of feature selection methods by using 1-, 2-gram combination. That is, both 1-grams and 2-grams in corpus are extracted as terms. We design eight experiment scenarios shown in Table 3 by adopting four kinds of features defined in Section 4.3 and three kinds of vector weights referred in Section 3.

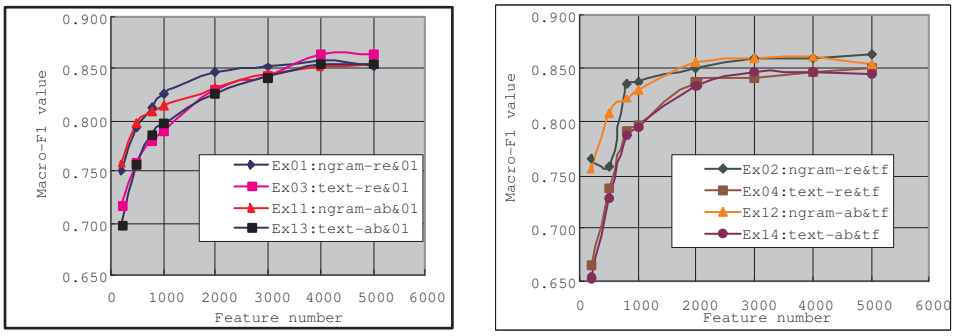
Table 3. Experiment scenarios list.

Experiment	Feature selection	Weight
Ex01	$Ngram\_freq\_relative$	0/1
Ex02	$Ngram\_freq\_relative$	TF
Ex03	$Text\_freq\_relative$	0/1
Ex04	$Text\_freq\_relative$	TF
Ex11	$Ngram\_freq$	0/1
Ex12	$Ngram\_freq$	TF
Ex13	$Text\_freq$	0/1
Ex14	$Text\_freq$	TF

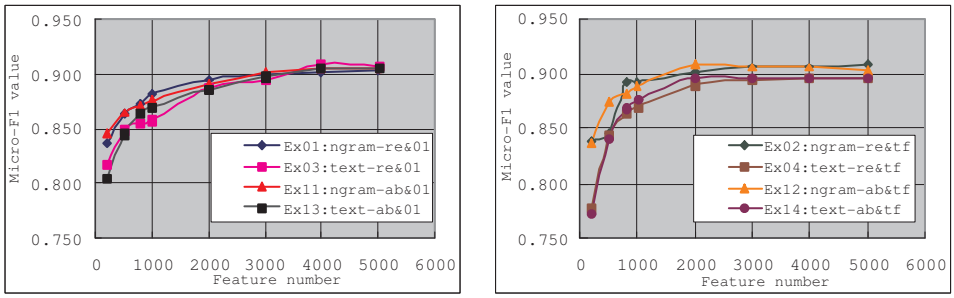
Fig.1 and Table 4 show the results using C-SVC classifier. Ex02 and Ex12 have the best performance, Ex01 and Ex11 are the second, Ex03 and Ex13 follow and the Ex04 and Ex14 get the worst results. When the number of features exceeds 3,000, all the experiments have quite similar performance. Whether 0/1 weight or TF weight, both Micro-F1 and Macro-F1 results show the performance ranking of four feature selection methods: *Gram_freq or Gram_freq_relative* > *Text_freq or Text_freq_relative*. That is, in four feature selections, n-gram frequency is better than text frequency and relative frequency does not get better results than absolute frequency.

Table 4. The scope of Macro-F1 and Micro-F1 in eight experiments (using 1,000 to 5,000 n-grams).

No. of Experiment	Macro-F1	Micro-F1
Ex02,Ex12	0.83-0.86	0.89-0.91
Ex01,Ex11	0.82-0.85	0.88-0.90
Ex03,Ex13	0.79-0.86	0.87-0.91
Ex04,Ex14	0.79-0.85	0.87-0.90



Comparison of Macro-F1 results on four feature selection methods



Comparison of Micro-F1 results on four feature selection methods

Fig. 1. Performance comparison on four kinds of feature selection methods (on C-SVC classifier).

We further use Naive Bayes classifier to testify the results of experiments Ex02, Ex04, Ex12 and Ex14. They represent four kinds of feature selection methods by using only TF weight for text representation. However, Naive Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variables. To get the independent variables (features), we should filter the correlated features first. Here, we adopt FCBF algorithm which can quickly identify relevant features as well as redundant ones[16]. From experiments above, we could find that Macro-F1 and Micro-F1 always give consistent results. As a result, here, we could only compare the Micro-F1 among these experiments. As shown in Fig.2, Ex02 and Ex12 have better performance than Ex04 and Ex14, which in-

dicates that feature selection based on n-gram frequency (absolute or relative) is better than that based on text frequency (absolute or relative). It is consistent with the results provided by C-SVC classifier.

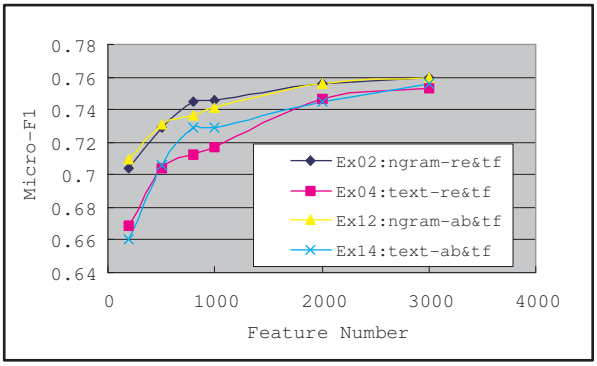


Fig. 2. Performance comparison on four kinds of features selection methods (on Bayes classifier).

5.3. *Experiments on Analyzing Feature Sparseness and Correlation*

In Section 3, we concerned that each document is considered as a vector in feature space. There are mostly zero values in each vector. Thus "text*feature" matrices include much zero values. As we all know, a matrix populated primarily with zeros is a sparse matrix. Sparseness problem is an important reason for degrading classification results. Here,we use "sparseness degree" to describe a matrix, which is the percentage of empty cells. Moreover, different sparseness degree often result in different time and space cost in program implementation. We further analyze the sparseness degree of "text*feature" matrices in four feature selection methods by comparing the non-zero value distribution in these matrices.

Fig.3 shows the non-zero value distribution in the "text*feature" matrix in the experiments Ex02, Ex04, Ex12 and Ex14. Ex04 (text-re&tf) has about two times less non-zero cells than Ex02 (ngram-re&tf), which indicates that Ex04 will produce less dense matrices after feature selection. Similarly, Ex14 (text-ab&tf) has about two times less non-zero cells than Ex12 (ngram-ab&tf). We also find out that the matrices are denser when we use an absolute frequency than a relative frequency. Accordingly, the matrix sparseness degree which is obtained by four feature selection methods is: $Gram\_freq < Gram\_freq\_relative < Text\_freq < Text\_freq\_relative$.

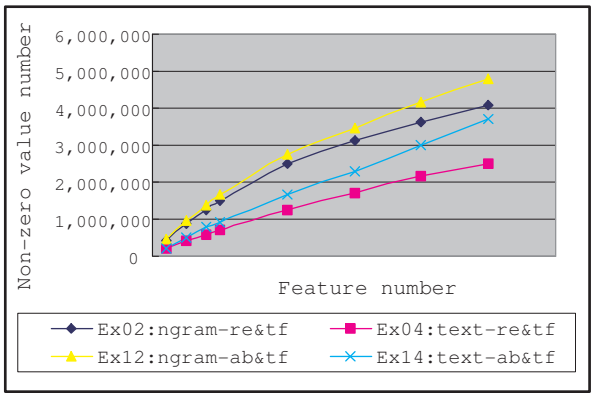


Fig. 3. Comparison on matrix sparseness.

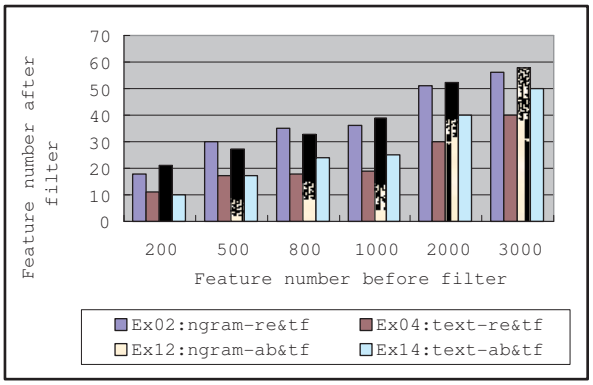


Fig. 4. Comparison on feature correlation.

Further, we analyze the matrix "text*feature". Which method of feature selection can we use for getting the less correlated features? Fig.4 compares the feature numbers after filtering by using FCBF algorithm in Naive Bayes classification process. We could find that Ex02 and Ex12 have more features than Ex04 and Ex14 after filtering. That is, there are more independent features using n-gram frequency (in Ex02: n-gram relative frequency and in Ex12: n-gram absolute frequency) than using text frequency (in Ex04 and Ex14).

5.4. *Experiments on Comparison Text Representation Weights*

According to previous experiment conclusions, we design three experiments by using 1-, 2-gram combination, n-gram relative feature selection method and three text representation weights (0/1 logical, TF and Tf*idf) in order to explore the performance difference of three weights. We still perform these experiments by using C-SVC classifier. From Section 5.2, we could find out that Macro-F1 and Micro-F1 always give consistent results. As a result, here, we could only compare the Micro-F1 among three experiments.

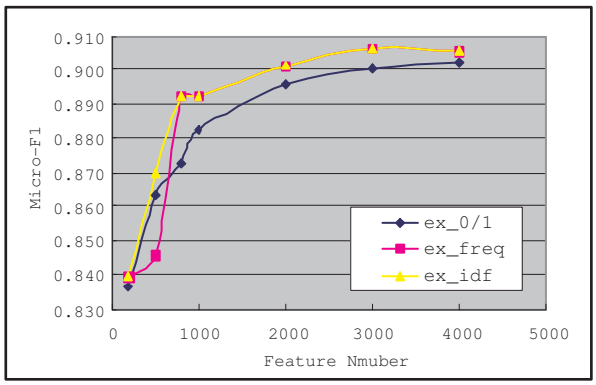


Fig. 5. Comparison on text representation weights.

As shown in Fig.5, Tf*idf weight always indicates the best performance. When we use more than 800 features for classification, Tf*idf and TF weight give almost the same results. Obviously, it is more complicated to calculate Tf*idf value than to calculate TF value. From these analysis, we can have the conclusion that TF weight and Tf*idf weight have similar performance for text representation and TF is more efficient when we use high dimensional features.

5.5. *Miss-Classified Texts Analysis*

The confusion matrix presented in Table 5 shows the predictions made by our model. It is a result of classification on test set using 3,000 n-grams in Ex11. The rows correspond to the known classes of the data, i.e. the labels in the data. The columns correspond to the predictions made by the model. The diagonal elements show the number of correct classifications for each class, and the off-diagonal elements show the miss-classified text numbers.

The main reason for some misclassifications comes from the similarities between texts in real world. For example, the class “art” and the class “entertainment” are close to each other. More generally, the class “science” could refer to many subjects in other classes. As an example, texts of medical science should be assigned to the label of class “medicine”, as well as class “science”. The same case appears in “computer” and “economy”, “education” and “career” etc. Some kinds of misclassified texts could belong to different classes. In Table 5, the numbers with a label of “*” are the numbers of texts classified in the class closer to the correct class. So, it should be more reasonable to construct a multi-classifier with multi-label.

Another reason for the decrease of results is the unbalanced distribution among different classes. Table 1 shows that the largest class “computer” includes 2865 texts, while the smallest class “region” only includes 150 texts.

6. Conclusion

In this paper, we discussed Chinese text classification based on n-grams by using different feature selection methods and different text representation weights. From the experiments, we get many conclusions.
a dot:

- In the case of using less than 3000 features, the feature selection methods based on n-gram frequency (absolute or relative) always give better results than those based on text frequency (absolute or relative). Relative frequency is not better than the absolute frequency. In the situation with more than 3000 features, results in all cases with both methods are similar.
- Feature selection based on n-gram frequency produces denser “text*feature” matrices than the ones based on text frequency. Feature selection based on absolute frequency produces denser “text*feature” matrices than the ones based on relative frequency.

- Feature selection based on n-gram frequency produces features which have less correlation than the ones based on text frequency.
- Text representation using TF weight has similar performance to those using Tf*idf weight. They have better performance than 0/1 logical weight.

In this paper, we also analyze the reason for the error rate. It mainly comes from the similarity between some classes. It would be better to construct a multi-label classifier. The other reason for the performance decrease is the unbalanced class distributions. Our future work will try to solve these problems. In addition, this paper mainly discuss text feature selection based on statistic methods using n-grams. In the situation of using words as terms, text semantic could be considered. We will continue to compare the performances between using n-grams and using words.

Acknowledgments

This paper is sponsored by the National Natural Science Foundation of China (No.60775036), (No.60970061) and (No.60475019), the Research Fund for the Doctoral Program of Higher Education of China (No.20060247039). Our colleagues, Professor Rakotomalala Ricco, Proffessor Annie MORIN and Feifei XU gave many good advices for this paper. We really appreciate their helps.

Table 5. Miss-classified texts analyses .

	art	car	career	comp.	economy	edu.	ente.	estate	medical	region	sci.	sport
art	84	0	1	8	1	3	*57	0	0	0	9	0
car	0	161	0	5	4	0	1	1	1	0	1	3
career	1	0	158	6	3	*11	1	0	1	0	1	0
comp.	0	0	4	856	8	6	4	0	3	0	2	0
economy	0	0	9	*24	197	2	0	2	3	0	8	1
edu.	10	0	*11	*13	2	194	3	0	2	0	7	0
ente.	*20	0	0	6	1	0	419	0	0	0	2	2
estate	1	0	0	4	2	0	0	270	2	1	0	0
medical	0	0	0	1	1	3	1	0	386	0	*30	0
region	5	1	2	3	1	0	0	2	1	28	2	0
sci.	4	0	0	*13	5	0	2	1	*27	2	257	1
sport	0	1	0	1	1	0	2	1	1	0	2	830

References

1. T.Joachims, "Learning to Classify Text Using Support Vector Machines,", *University Dortmund, February*, (2001).
2. J.Radwan and J.Chauchat, "Pourquoi les n-grammes permettent de classer des textes? Recherche de mots-clefs pertinents à l'aide des n-grammes caractèristiques," *(JADT2002)Proc. 6th Intl. Conf. on Statistical Analysis of Textual Data*, 381–390 (2002).
3. A.Lelu, Mohamed Halleb and Bruno Delprat, "Recherche d'information et cartographie dans des corpus textuels à partir des fréquences de n-grammes," *(JADT1998)Proc. 4th Intl. Conf. on Statistical Analysis of Textual Data*, Université de Nice - Sophia Antipolis,France, 391–400 (1998).
4. S.Zhou et al., "A Chinese Document Categorization System Without Dictionary Support and Segmentation Processing," *J. Comput. Research and Development*, **38**(7), 839–844 (2001).
5. Z.Wei et al. , "Comparing different text representation and feature selection methods on Chinese text classification using Character n-grams," *(JADT2008)Proc. 9th Intl. Conf. on Statistical Analysis of Textual Data*, Lyon, France, **II**, 1175–1186 (2008).
6. Z.Wei et al. , "Feature Selection on Chinese Text Classification Using Character N-grams," *(RSKT2008) Proc. 3rd Intl. Conf. on Rough Sets and Knowledge Technology*, Chongqing, China, 500–507 (2008).
7. Y.Yang, J.P.Pedersen "A Comparative Study on Feature Selection in Text Categorization," *(ICML'97)Proc. 4th Intl. Conf. on Machine Learning*, 412–420 (1997).
8. X.Zhou, X.Zhang, X.Hu, Semantic smoothing for

Bayesian text classification with small training data, *(SIAM2008) Proc. Intl. Conf. on Data Mining*, Atlanta, Georgia, USA,289–300 (2008).
9. D.Lin et al. "An ontology-based document feature extraction," *Computer Science*, **35**(3),152–154 (2008).
10. F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, **34**(1), 1–47 (2002).
11. J.P.Benzécri , "L'Analyse des Données, T1 = la Taxinomie," *DUNOD, Paris*, (1973).
12. S. Tan et al., "A novel refinement approach for text categorization," *(CIKM'05)ACM 17th Conf. on Information and Knowledge Management*, 469–476 (2005).
13. R.E.Fan et al., "Working set selection using second order information for training SVM," *J. Machine Learning Research*, **6**,1889–1918 (2005).
14. R.Rakotomalala , "TANAGRA : un logiciel gratuit pour l'enseignement et la recherché," *(EGC'2005) Journées Francophones "Extraction et Gestion des Connaissances"*, **2**, 697–702 (2005).
15. C.J.Van Rijsbergen, "Information Retrieval," *Butterworths, London*, (1979).
16. L. Yu, H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Proc. IEEE Intl. Conf. on Machine Learning* , Washington DC, 856–863 (2003).