Comparing different text representation and feature selection methods on Chinese text classification using Character n-grams

Zhihua Wei [1, 2], Jean-Hugues Chauchat [1] and Duoqian Miao [2]

[1] ERIC, Université Lumière Lyon 2, 5 avenue Pierre Mendès-France
69 676 Bron Cedex France

[2] Key laboratory “Embedded System and Service Computing” Ministry of Education
Tongji University, 4800 Cao’an Road, 201 804, Shanghai, China

Abstract

In this paper, we perform Chinese text categorization using n-gram text representation on TanCorpV1.0 which is a new corpus, special for Chinese text classification of more than 14,000 texts divided in 12 classes. We use a combination of methods, including between inter-class feature reduction methods and cross-class feature selection methods. We use the C-SVC classifier (with a linear kernel) which is the SVM algorithm made for the multi-classification task. We perform our experiments in the TANAGRA platform.

Our experiments concern: (1) the performance comparison between using both 1-, 2-grams and using 1-, 2-, 3-gram in Chinese text representation; (2) the performance comparison between using different feature representations: absolute text frequency, relative text frequency, absolute n-gram frequency and relative n-gram frequency; (3) the comparison of the sparseness in the “text*feature” matrix between using n-gram frequency and frequency in feature selection; (4) the performance comparison between two text coding methods: the 0/1 logical value and the n-gram frequency numeric value. We found out that in the case of using less than 3,000 features, the feature selection methods based on n-gram frequency (absolute or relative) always yield better results.

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

Résumé

Dans ce papier, nous travaillons au classement (discrimination) des textes de l’ensemble TanCorpV1.0, en les codant par les n-grams ; cet ensemble est un nouveau corpus en langue chinoise de plus de 14,000 textes étiquetés. Nous comparons différentes combinaisons de méthodes pour – la réduction du nombre de variables dans chaque classe, et – de sélection des variables discriminantes des classes entre elles. Dans la plateforme de logiciel libre TANAGRA, nous utilisons la méthode des vecteurs de support (SVM) adaptée au cas multi classe : le classifieur C-SVC (avec un noyau linéaire). Nos résultats montrent que le codage en n-grams donne des résultats comparables à celui des « sacs de mots » pour la catégorisation de textes en langue chinoise.

Comparée à la représentation des textes en « sacs de mots », le codage en « n-gram de caractères » est simple et efficace, notamment pour les textes en langue chinoise dans laquelle il n’y a pas de séparateurs entre les mots ; le découpage automatique en mots d’une suite de caractères chinois est très difficile et donne lieu à de nombreuses erreurs. Cependant les n-grams extraits des textes sont souvent plus nombreux que les mots et le vecteur de représentation est dans un espace de grande dimension.

Nous aboutissons à plusieurs résultats : (1) comparaison du codage en 1 et 2-grams comparé à celui en 1, 2 et 3-grams pour les textes chinois ; (2) comparaison de 4 formes de fréquences : fréquences absolues, ou relatives, dans chaque class, et fréquences absolues, ou relatives, des n-grams dans chaque class ; (3) comparaison du nombre de cellules vides dans la matrice « textes * variables » ; (4) comparaison des performance du classifieur en utilisant deux caractérisation des textes : valeurs logiques (le texte contient ou non le n-gram) ou numériques (fréquence absolue du n-gram dans le texte). Enfin, nous analysons les raisons de « mauvaise classification » de certains textes, et proposons plusieurs pistes de solution.

1. Introduction

In recent years, much attention has been given to the Chinese text classification (TC) with the rapidly increasing quantity of web sources and electronic texts in Chinese. The problem of Chinese TC is difficult and challenging. In addition to the difficulties existing in the counterpart problem 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 words frequently used in Chinese, which are much more than the number of words used in English.

Usually, there are two steps in the construction of an automated text classification system. The first one is that the texts are being preprocessed into a representation more suitable for the learning algorithm that is applied afterwards. There are various ways of representing a text such as by using word fragments, words, phrases, meanings, and concepts (Thorsten Joachims, 2001). Different text representations have different dependence on the language of the text. The second step regards the learning algorithm that is chosen. In this work we focus on the first step.

In this paper, we use a method independent of languages which represents texts with n-grams. The performance of two kinds of n-gram combinations (1-, 2-gram and 1-, 2-, 3-gram) is given. We adopt two kinds of weight: logical weights (0/1: if the feature occurs in a text, it has a value “1”, otherwise “0”) and numerical weights (frequencies of features in the texts). We perform inter-class feature reduction and feature selection among all the classes in the training set. We compare the results using different absolute frequencies and relative frequencies in the process of CHI-Square feature selection. We adopt a multi-classification SVM as the learning algorithm.

In section 2, we briefly introduce the main difficulties in Chinese text representation and the advantage of using n-grams for text representation. In section 3, we give some definitions used in this paper. In section 4, the feature number reduction strategy inter-class and the feature selection method in the training set are presented. In section 5, the text representation methods in our work are introduced and, in section 6, the experiment dataset and the experiment scenarios are presented. In section 7, the experimental results are compared and the conclusion is given in section 8.

2. Difficulties in Chinese text representation and the advantage of n-gram representation

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 an n-gram. These features play the same role in Chinese TC. However, unlike most of the western languages, the 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. The word sense disambiguation issue and the 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) *物理学 / 起来 / 很 / 难。(error)*

Physics/ study / up / very / difficult. *Physics / up / very / difficult.*

Word sense disambiguation (WSD) is one of the most important and also 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 prevalent in winter.)”, “*流感*” is a abbreviation of a disease which is a kind of unknown word, whereas, “*感到*” is a word in dictionary. The above sentences may have two kinds of segmentations. But only in the first kind of segmentation, the word “flu” (which indicates that the document belong to the medical field) can be recognized.

流感/ 到/ 冬天/ 很/ 普遍/ 。(right)   *流/ 感到/ 冬天/ 很/ 普遍/ 。(error)*

Flu / arrive / winter / very / prevalent.   *Flow/ feel / winter / very / prevalent.*

This example points out the problem of automatically the segmentation task, since the relevant algorithm should decide which segmentation is the correct one.

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 is moved one character at a time. Then, the number of occurrences of each n-gram is counted (Jalam Radwan et Jean-Hugues Chauchat, 2002).

There are several advantages of using n-grams in TC tasks (Lelu et al., 1998). 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. N-grams are also language independent.

N-gram extraction on a large corpus will yield a large number of possible n-grams, but only some of them will have significant frequency values in vectors representing the texts and good discriminate power.

3. Some definition

In text classification, the text is usually represented as a vector of weighted features. The difference between various in text representations comes from the definition of “feature”. This work explores four kinds of feature building methods with their variations.

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

Absolute text frequency is noted as $Text\_freq_{ij.}$

$$Text\_freq_{ij} = \text{the number of texts which include n-gram } j \text{ in class } c_i. \quad (1)$$

Relative text frequency is noted as $Text\_freq\_relative_{ij}$.

$$Text\_freq\_relative_{ij} = \frac{Text\_freq_{ij}}{N_i}, \text{ here, } N_i \text{ is the quantity of texts in class } c_i \text{ in training set;} \quad (2)$$

Absolute n-gram frequency is noted as $Gram\_freq_{ij.}$

$$Gram\_freq_{ij} = \text{the number of n-gram } j \text{ in all texts in class } c_i \text{ in training set;} \quad (3)$$

Relative n-gram frequency is noted as $Gram\_freq\_relative_{ij.}$

$$Gram\_freq\_relative_{ij} = \frac{Gram\_freq_{ij}}{N'_i} \quad (4)$$

Here, N'_i is the total of occurrence of n-gram in all texts in class c_i *in training set.*

4. Feature number reduction inter-class and discriminate feature selection cross-class

Feature selection is a term space reduction method which attempts to select the more discriminative 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. The first is reducing the number of features inter-class. The second is choosing the more discriminate features among all the classes in training set.

4.1. Feature number reduction inter-class

We can extract all the 1-gram and 2-gram in all the texts of the training set. However, all the items of (n-gram, its frequency) in each class will be more than 15,000 in average. Most of them are the n-grams which occur only one or two times. So it is necessary to reduce some inter-class features. We adopt two kinds of strategies: relative text frequency method and absolute frequency method.

4.1.1. Relative text frequency method

In each class, we removed the n-gram *j* which has $Text\_freq\_relative_{ij}>0.02$. Then all the n-grams extracted from each class are put together for feature selection cross-class.

4.1.2. Absolute frequency method

In each class, we removed the items which occur less than 3 times in a class. When we calculate all the n-grams in the training set, we only keep the n-grams which appear more than 100 times in the whole training set for feature selection cross-class. In our experiments, about 18,000 n-grams are left in this step.

4.2. Feature selection cross-class

In this step, each feature "*j*" is assigned a numeric score based on the occurrence of features within the different document classes "c_i". The choice of the scoring method in this work is the CHI-Square test. There are many other tests available as summarized in (Fabrizio Sebastiani, 2002) but the CHI-Square is often cited as one of the best tests for the feature selection. It gives a similar result as Information Gain because it is numerically equivalent as shown by (Benzécri, 1973).

The score of n-gram "*j*" is $\sum_i \frac{(O_{ij}-E_{ij})^2}{E_{ij}}$

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: $E_{ij} = \frac{O_{i+} \times O_{+j}}{O_{++}}$.

Here, we define four kinds of values on $O_{ij.}$

1) Relative text frequency $Text\_freq\_relative_{ij}$.

2) Relative n-gram frequency $Gram\_freq\_relative_{ij.}$

3) Absolute text frequency $Text\_freq_{ij.}$

4) Absolute n-gram frequency *Gram_freq*$_{ij.}$

One must perform the feature selection only on the learning set and not on the whole corpus. In our work, 70% texts in each class are selected by random to constitute the learning set and the rest 30% are used for the testing set. According to the result of CHI-Square, we separately perform the classification using the 200, 500, 800, 1000, 2000 …5000 features.

5. Text representation

We adopt the VSM (Vector Space Model), where each document is considered to be a vector in feature space. Thus, given a set of N documents, {d_1, d_2…d_N}, the table of “document*feature” is constructed such as that shown in table 1, where each document is represented by a core “w_{ij}” in relation to each of M chosen features. In our work, two kinds of value of “w_{ij}” are applied.

1) w_{ij} = absolute frequency of feature *j* in document *i.*

2) w_{ij} =0 or 1, w_{ij}=1, if feature *j* appears in document *i*, otherwise, w_{ij}=0.

D	F_1	F_2	…	…	…	F_M	Class	Status
d_1	w_{12}	w_{13}	…			w_{1M}	A	Learning
d_2	w_{21}	…				w_{2M}	B	Learning
…		…					…	Learning
d_i	w_{i1}	…	w_{ij}			w_{iM}	C	…
…		…					…	Testing
d_N	w_{N1}	…				w_{NM}	A	Testing

*Table 1: Document*feature vector table*

6. Experiment

6.1. Chinese text classification benchmark

For a long time, there was no special benchmark for Chinese text categorization. Many reported results were achieved in some benchmark of information retrieval which did not include the class information. In order to use this kind of benchmark, it is necessary to clustering the datasets before the task of categorization. The TanCorp corpus, a collection of 14,150 texts in Chinese language, has been collected and processed by Songbo Tan (Songbo Tan, 2005). The corpus is divided in two hierarchical levels. The first level contains 12 big categories (art, car, career, computer, economy, education, entertainment, estate, medical, region, science and sport) and the second consists of 60 subclasses. This corpus can serve as three categorization datasets: one hierarchical dataset (TanCorpHier) and two flat datasets (TanCorp-12 and TanCorp-60). In our experiment, we use TanCorp-12. Table2 shows the distribution of TanCorp-12.

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

Table 2: the distribution of TanCorp-12 (M= megabyte)

6.2. Experiment description

In order to test the results given from different kinds of methods in feature selection and text representation we set different experiment scenarios, as described in table 3. In the following section, we use a number (e.g. Ex_01) and a long name (e.g. 1,2-gram&inter-relative&cross-ngram-re&01) to describe each experiment scenario. The first part of the long name can be "1,2-gram" or "1,2,3-gram" and it notes the items extracted from texts as features. The second part can be "inter-relative" or "inter-absolute" and it notes the feature number reduction method inter-class, the third part "cross-text-re", "cross-ngram-re", "cross-ngram-ab" or "cross-text-ab" notes the feature selection method cross-class and the last part "01" or "freq" notes the text representation method.

	n-gram combination	Feature number reduction inter-class	Feature selection cross-class	Text representation
Ex_01: 1,2-gram&inter-relative&cross-ngram-re&01	1 +2-gram	Relative frequency	Relative n-gram frequency	0/1
Ex_02: 1,2-gram&inter-relative&cross-ngram-re&freq	1 +2-gram			n-gram frequency
Ex_03: 1,2-gram&inter-relative&cross-text-re&01	1 +2-gram		Relative text frequency	0/1
Ex_04: 1,2-gram&inter-relative&cross-text-re&freq	1 +2-gram			n-gram frequency
Ex_11: 1,2-gram inter-absolute&cross-ngram-ab&01	1 +2-gram	Absolute frequency	Absolute n-gram frequency	0/1
Ex_12: 1,2-gram&inter-absolute&cross-ngram-ab&freq	1 +2-gram			n-gram frequency
Ex_13: 1,2-gram&inter-absolute&cross-text-ab&01	1 +2-gram		Absolute text frequency	0/1
Ex_14: 1,2-gram&inter-absolute&cross-text-ab&freq	1 +2-gram			n-gram frequency
Ex_21:1,2,3-gram&inter-absolute&cross-ngram-ab&freq	1+2+3-gram	Absolute frequency	Absolute n-gram frequency	n-gram frequency
Ex_22: 1,2,3-gram&inter-relative&cross-text-re&01	1+2+3-gram	Relative frequency	Relative text frequency	0/1

Table 3: Experiment scenarios list

We use the C-SVC classifier which was introduced in LIBSVM (R.-E. Fan et al., 2005). It is the SVM algorithm proper for the multi-classification task. We use a linear kernel since text classification problems are usually linearly separable. Learning parameters are set to gamma=0 and penalty cost=1. We perform our experiments in the platform TANAGRA which is a free data mining software for academic and research purposes developed by Ricco Rakotomalala (Rakotomalala, 2005).

We use the F1 measure introduced by (van Rijsbergen, 1979). This measure combines recall and precision in the following way for the bi-class situation.

$$\mathrm{Re}\,call = \frac{number\_of\_correct\_positive\_predictions}{number\_of\_positive\_examples}$$

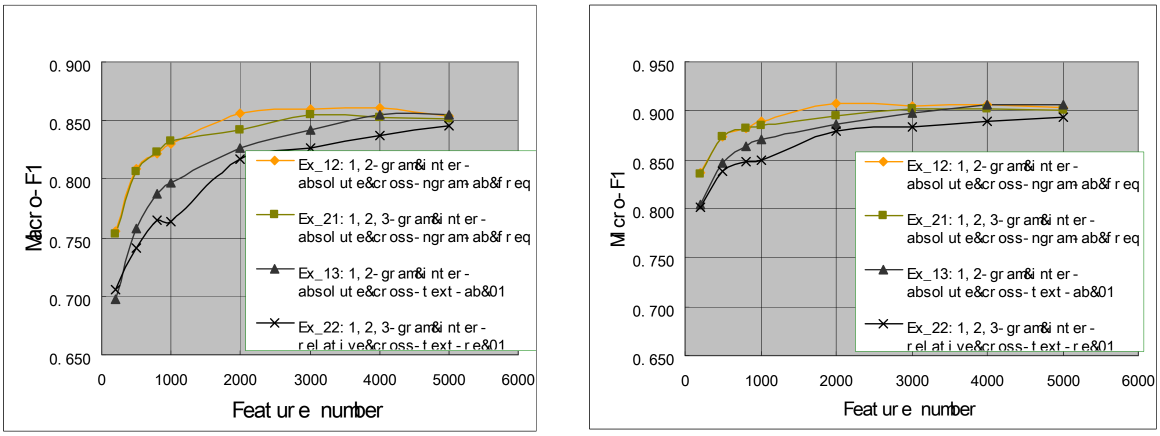
$$\mathrm{Pr}\,ecision = \frac{number\_of\_correct\_positive\_predictions}{number\_of\_positive\_predictions}$$

$$F1 = \frac{2 * \mathrm{Re}\,call * \mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,ecision}$$

For more than 2 classes, the F1 scores are summarized over the different categories using the Micro-averages and Macro-averages of F1 scores.
Micro - F1 = average in documents and classes
Macro - F1 = average of within - category F1 values

7. Results and discussions

7.1. Comparison between 1,2-gram and 1,2,3-gram



Macro-F1 comparison on 1,2-gram(Ex_12, Ex_13) and 1,2,3-gram(Ex_21, Ex_22) (Feature selection by ABSOLUTE Text or N-gram Frequency)

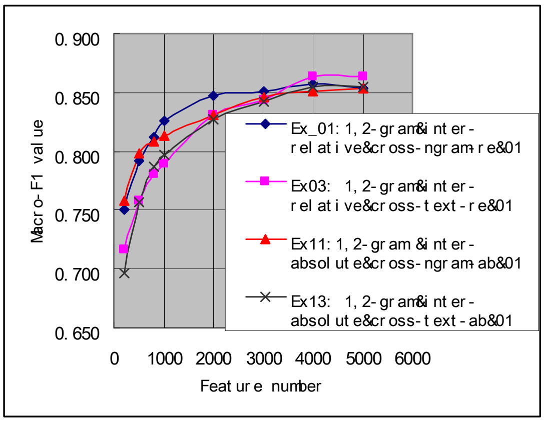
Micro-F1 comparison on 1,2-gram(Ex_12, Ex_13) and 1,2,3-gram(Ex_21, Ex_22) (Feature selection by ABSOLUTE Text or N-gram Frequency)

Graph1: Performance comparison between1,2-gram and 1,2,3-gram

Either for macro-F1 or Micro-F1, graph1 shows that Ex_12 (1,2-gram) and Ex_21 (1,2,3-gram) have similar results and Ex_13 (1,2-gram) and Ex_22 (1,2,3-gram) have worse performance than Ex_12 and Ex_21 when feature number is less than 4000. It indicates that both combination of 1, 2-gram and 1, 2, 3-gram can well represent Chinese texts. However, there will be many more candidate features when using 1, 2, 3-grams than 1, 2-grams. So, we prefer to use the combination of 1, 2-gram.

7.2. Result comparison in four kinds of methods in cross-class feature selection

Graph 2 and table 4 shows that when the feature number is less than 3,000, Ex_02 and Ex_12 have the best performance, Ex_01 and Ex_11 have the second best, Ex_03 and Ex_13 follow and the Ex_04 and Ex_14 have the worst results. When there are more than 3,000 features, all the experiments have quite similar performance. The best results are in the Ex_02, Ex_12, Ex_01 and Ex_11 which are all using **n-gram frequency** (relative or absolute) for feature selection. In the situation of absolute frequency and relative frequency, the results are similar. The results indicate that using n-gram frequency for feature selection is better than using text frequency. Also the relative frequency does not give better results than the absolute frequency. When using n-gram frequency in cross_class feature selection, the results produced by using logical weights 0/1 binary and absolute n-gram frequency for text representation are similar, as it is shown in experiments Ex_01 and Ex_02, Ex_11 and Ex_12. When using text frequency in cross_class feature selecting, the results produced when using 0/1 are better than those produced when using n-gram frequency.



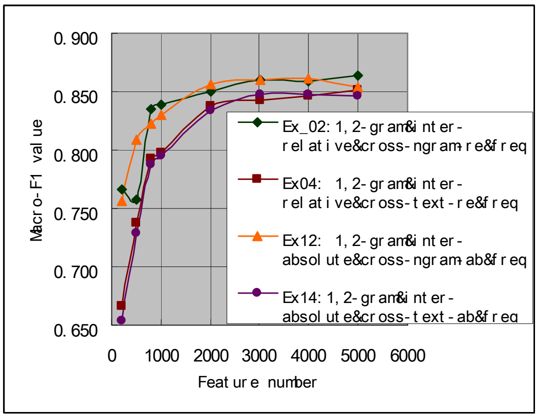
Graph2-1: Comparison of Macro-F1 results on four feature selection methods:

relative n-gram frequency(Ex_01),

relative text frequency(Ex_03),

absolute n-gram frequency(Ex_11);

absolute text frequency(Ex_13).



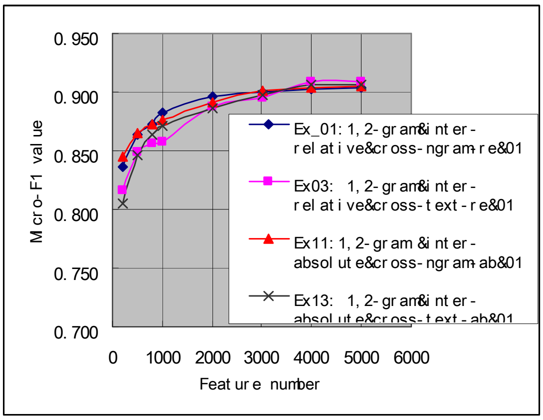
Graph2-2: Comparison of Macro-F1 results on four feature selection methods:

relative n-gram frequency(Ex_02),

relative text frequency(Ex_04),

absolute n-gram frequency(Ex_12);

absolute text frequency(Ex_14).



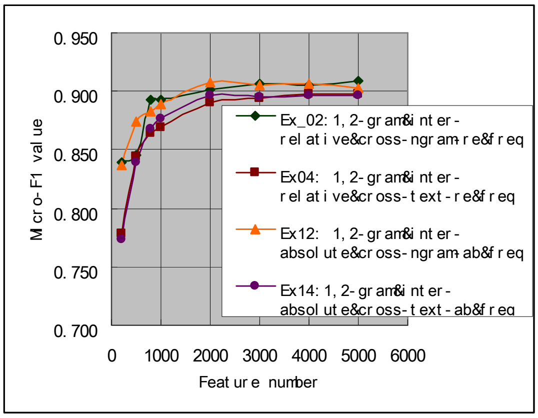
Graph2-3: Comparison of Micro-F1 results on four feature selection methods:

relative n-gram frequency(Ex_01),

relative text frequency(Ex_03),

absolute n-gram frequency(Ex_11);

absolute text frequency(Ex_13).



Graph2-4: Comparison of Micro-F1 results on four feature selection methods:

relative n-gram frequency(Ex_02),

relative text frequency(Ex_04),

absolute n-gram frequency(Ex_12);

absolute text frequency(Ex_14).

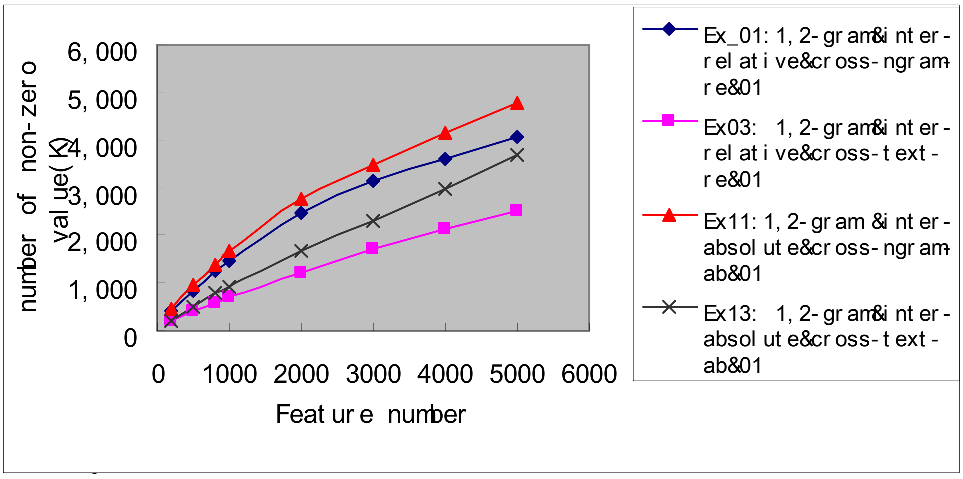
Graph 2: Performance comparison on four kinds of features selection methods

No. of Experiment	Macro-f1	Micro-F1
Ex_02, Ex_12	**0.83~0.86**	**0.89~0.91**
Ex_01, Ex_11	**0.82~0.85**	**0.88~0.90**
Ex_03, Ex_13	**0.79~0.86**	**0.87~0.91**
Ex_04, Ex_14	**0.79~0.85**	**0.87~0.90**

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

7.3. Sparseness comparison

(Šilić et al., 2007) shows that the computational time is more linked with the number of non-zero values in the cross-table (text*feature) than with its number of columns (features). Graph 3 shows the non-zero value distribution in the "text*feature" matrix for experiments Ex_01, Ex_03, Ex_11 and Ex_13. Ex_03 (1,2-gram&inter-relative&cross-text-re&01) has about two times less non-zero cells than Ex_01 (1,2-gram&inter-relative&cross-ngram-re&01), which indicates that it will produce less dense matrices after cross-class feature selection, so in this way the computation will be faster. Similarly, Ex_13 (1,2-gram&inter-absolute&cross-text-ab&01) has about two times less non-zero cells than Ex_11 (1,2-gram &inter-absolute&cross-ngram-ab&01). The matrices are denser when we use an absolute frequency than a relative frequency. For example, Ex_11 has more non-zero cells than Ex_01 and Ex_13 has more non-zero cells than Ex_03.



relative n-gram frequency(Ex_01), relative text frequency(Ex_03),

absolute n-gram frequency(Ex_11), absolute text frequency(Ex_13).

Graph 3: Comparison on matrix sparseness

7.4. Discussion on wrongly classified documents

The confusion matrix presented in table 5, shows the predictions made by our model. 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 made 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 many classes. As an example, texts of medical science should be assigned the label of class “medicine” AND/OR class “science”. The same stands for “computer” and “economy”, “education” and “career” etc. Some kinds of misclassified texts could undoubtedly belong to different classes. In the table 5, the numbers with a label of “*” are the numbers of texts classified in a class close to the correct class. So, it should be more reasonable to construct a multi-classifier with multi-label.

Another reason for the decrease of F1 values is the inclining distribution among different classes and different texts. Table 1 show that the biggest class “computer” includes 2865 texts, while the smallest class “region” only includes 150 texts.

		PREDICTED CLASSES											
		art	**car**	**career**	**com**	**eco**	**edu**	**ente**	**estate**	**med**	**region**	**sci**	**sport**
TRUE CLASSES	**art**	**87**	0	0	6	4	6	***46**	2	0	2	11	0
	car	0	**163**	0	4	1	0	0	0	4	0	2	3
	career	0	0	**160**	3	4	***14**	0	0	0	0	1	0
	Com	0	1	6	**845**	***16**	5	3	0	0	0	6	1
	Eco	1	2	9	***31**	**190**	0	0	5	0	0	7	1
	Edu	***8**	0	10	***13**	3	**193**	4	0	1	2	8	0
	Ente	***27**	1	1	6	0	0	**409**	0	0	1	3	2
	estate	0	0	1	0	0	0	0	**279**	0	0	0	0
	Med	1	0	1	5	2	2	0	0	**385**	0	***26**	0
	region	2	0	0	5	1	0	6	1	1	**28**	1	0
	Sci	3	1	0	***14**	3	4	1	2	***26**	1	**257**	0
	sport	0	0	0	4	2	1	4	0	1	0	0	**830**

Table 5: A result of classification on test set using 3,000 n-grams in Ex21: 1,2,3-gram&inter-absolute&cross-ngram-ab&freq (we use the abbreviation for some classes: computer-com, economy-eco, education-edu, entertainment-ente, medical-med, science-sci)

8. Conclusion

In this paper, we perform Chinese text categorization using n-gram text representation and a different combination of methods in inter-class feature reduction and cross-class feature selection. Our experiments show that using a combination of 1-, 2-grams and 1-, 2-, 3-grams have similar performance.

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. Methods based on n-gram frequency also produce denser "text*feature" matrices. The method of absolute frequency has competitive behavior compared to the method of relative frequency.

The main reason why the error rate increases is the similarity between some classes. It is more feasible to construct a multi-label classifier than a bi-label classifier. The other reason for the performance decrease is the different between the number of the texts in large classes and small classes, that is, data inclining. Our future work will try to solve these problems.

Acknowledgments

This paper is sponsored by the National Natural Science Foundation of China (No. 60775036) and the Research Fund for the Doctoral Program of Higher Education of China (No. 20060247039). Our colleague, Anna Stavrianou gave many good advices for this paper. We really appreciate her helps.

References

Benzécri J.-P. (1973). *L'Analyse des Données, T1 = la Taxinomie*. Dunod, Paris.

Fan R.-E., Chen P.-H. and Lin C.-J. (2005). Working set selection using second order information for training SVM. *Journal of Machine Learning Research 6*: 1889-1918.

Jalam R., Chauchat J.-H. (2002). Pourquoi les n-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques. In *JADT 2002, 6^es^ Journées internationales d'Analyse statistique des Données Textuelles*, pages 381-390.

Lelu A., Halleb M., Delprat B. (1998). Recherche d'information et cartographie dans des corpus textuels à partir des fréquences de n-grammes. In Mellet Sylvie (ed.), *4^es^ Journées Internationales d'Analyse statistique des Données Textuelles,* Université de Nice - Sophia Antipolis, pages 391-400.

Rakotomalala R. (2005). TANAGRA : un logiciel gratuit pour l'enseignement et la recherche. *EGC'2005, RNTI-E-3,* Vol. 2, pages 697-702.

Sebastiani F.(2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1): 1-47.

Šilić A. et al. (2007). Detailed experiment with letter n-gram method on Croatian-English parallel corpus. *EPIA'07, Portuguese Conference on Artificial Intelligence*, 3-7 Dec. 2007.

Songbo Tan et al. (2005). A novel refinement approach for text categorization. *CIKM'05*, pages 469-476.

Thorsten J. (2001). *Learning to Classify Text Using Support Vector Machines*. University Dortmund, February.

Van Rijsbergen C. J. (1979). *Information Retrieval*. Butterworths, London.