Research on Feature Selection and kNN Classification Method in Chinese Text Classification

XIAO Chao[1, a], WU Ping[2,b *]

[1]School of Information Science and Technology, East China Normal University

No. 629, Zhongjiang Road, Shanghai 200062, China

[2] School of Information Science and Technology, East China Normal University

No. 3663, North Zhongshan Road, Shanghai 200062, China

[a]xfxc88@126.com, [b]pwu@cc.ecnu.edu.cn

Keywords: Chinese text classification, feature selection, text similarity, kNN, unbalanced degree of term distribution

Abstract. Scholars at home and abroad have done lots of research on feature selection methods in Chinese text classification, such as document frequency (DF), information gain (IG), and a χ^2-test (CHI). On the basis of their work, we propose a new selection method of counting the unbalanced degree of term distribution, compare it with other feature selection methods using the k-nearest-neighbor (kNN) algorithm, and find that the new method performs as well as CHI and IG. Experiments have shown that whatever the feature selection method we choose, after the number of features reaches a certain value, the gain of classification accuracy becomes very slight. Keep increasing the feature dimension can hardly improve the classification performance, while the time consumed doubles. In that case, we attempts to improve the kNN method by counting the text similarity differently. The improved method will quantify each feature's weight using a bit string, count the similarity of two documents under their bits mode, and finally remarkably reduce the space required for storing documents and the time consumed by counting their similarity. Experiments have confirmed that the new kNN method can greatly accelerate the speed of classification at the expense of a little loss of classification accuracy.

Introduction

With the rapid development of information technology, retrieving information becomes very simple. People can get access to information through the Internet conveniently. But when they take a closer look at the information downloaded from the Internet, they will probably find that there are too many useless, repetitive and outdated information and it takes lots of time to filter out the useful information manually. In order to get the valuable information efficiently, we use text categorization to solve this problem. Text categorization is the problem of automatically assigning predefined categories to free text documents. As to Chinese text categorization, the first difficulty is how to divide one document into a list of meaningful terms (words or phrases), since unlike English, Chinese text does not have a natural delimiter between words. Chinese word segmentation methods are not the focus of our work, and we use a segmentation software offered by Professor Zhang Huaping[1] instead. In addition, there is an important thing to mention that perfect precision and disambiguation in Chinese word segmentation can not be reached. As a result, the inherent errors caused by word segmentation always remain as a problem in Chinese information processing. However, we can focus on the performances of different feature selection methods as long as the errors caused by word segmentation are reasonably low. The second difficulty of Chinese text categorization is the high dimensionality of the feature space. The native feature space may consist of tens or hundreds of thousands of unique terms for even a moderate-sized text collection that is prohibitively high for many classification algorithms. It is highly desirable to reduce the native space without sacrificing categorization accuracy. The process of feature selection is designed to achieve such a goal. The final difficulty or characteristic is the applications of different classification algorithms to Chinese text

categorization. In this paper, we examine the capabilities of the k-nearest-neighbor (kNN) algorithm in mining categorization knowledge from high dimensional document feature vectors. Then we find out an easy way to calculate the text similarity and make the process of classification more quickly with a little loss of classification accuracy.

Feature Selection Methods

Feature selection has become the focus in the field of machine learning research. Its task in text classification is to use a term-goodness criterion thresholded to eliminate a large number of terms from the vocabulary of a corpus. Document frequency (DF), information gain (IG), and a χ^2 statistic (CHI) are the three most commonly used criteria in text classification[2]. At the last part of this section we propose a new criterion based on the unbalanced degree of term distribution inspired by the CHI method. All feature selection methods follow the same general module structure[3] as described by the pseudo code below:

Input	a dictionary D, a class c, and the number of feature words k
Output	a list of feature words belonging to the class c donated as Sel
Step 1	select valid words from the dictionary D to form a vocabulary V;
Step 2	let L to be an empty list;
Step 3	for each word $t \in V$, repeating the following sub-steps;
Step 3.1	compute the weight of t belonging to the class c denoted as $A(t,c)$;
Step 3.2	make a pair of t and its weight $A(t,c)$, then insert it into the list L in order of weight;
Step 4	select the first k elements in list L, and return the corresponding k feature words as Sel

Document Frequency (DF). Document frequency is the number of documents in which a term occurs. We compute the document frequency for each term in the training corpus and remove those terms whose document frequency is less than some predetermined threshold. DF thresholding is the simplest feature selection method and it can easily scales to large corpus, with a computational complexity approximately linear in the number of training documents. The principle of this method is based on an easy assumption that rare terms are either non-informative for category predication, or not influential in global performance[4]. As a result, we can aggressively remove rare terms without losing much categorization information and select the terms occurring more frequently to construct the feature space.

Information Gain (IG). Information gain measures the number of bits of information obtained from category prediction by knowing the presence or absence of a term in a document[5]. The information gain of term t can be defined as:

$$G(t) = -\sum_{i=1}^{m} p_r(c_i)\log p_r(c_i) + p_r(t)\sum_{i=1}^{m} p_r(c_i|t)\log p_r(c_i|t) + p_r(\bar{t})\sum_{i=1}^{m} p_r(c_i|\bar{t})\log p_r(c_i|\bar{t}) \quad (1)$$

$p_r(c_i)$ denotes the probability of a document belonging to category c_i. $p_r(t)$ denotes the probability of term t occurring in a document while $p_r(\bar{t})$ denotes the probability of term t not occurring in a document. $p_r(c_i|t)$ is the conditional probability of category c_i given term t, and $p_r(c_i|\bar{t})$ denotes the conditional probability of category c_i when term t is absent in a document. m is the number of categories. Given a training corpus, we compute the information gain for each term, remove those terms whose score is less than some predetermined threshold and finally get the feature space we wanted.

χ^2 Statistic (CHI). The basic idea of the χ^2 statistic method is to determine whether the assumption we made is correct or not by comparing the theoretical value and the practical value. First we assume that a term t and a category c are independent of each other. Then we calculate the two-way contingency table of them theoretically, and get the practical value of the contingency table

statistically. Finally, we compare the two different tables and compute the difference according to Eq. 2. The larger the value is, the more likely term t is associated with category c.

$$\chi^2(D,t,c) = \sum_{i=1}^{n} \frac{(x_i - E)^2}{E} \tag{2}$$

x_i is the i th pratical value of the contingency table, E is the corresponding theoretical value. For a term t and a category c, N_{11} denotes the number of documents which belong to category c and term t occurs in; N_{10} denotes the number of documents not belonging to category c and in which term t occurs; N_{01} denotes the number of documents belonging to category c and in which term t does not occur; N_{00} denotes the number of documents not belonging to category c and in which term t does not occur. Substitute these values into Eq. 2, then we get the more practical equation[6]:

$$\chi^2(D,t,c) = \frac{(N_{11}+N_{10}+N_{01}+N_{00}) \times (N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11}+N_{01}) \times (N_{11}+N_{10}) \times (N_{10}+N_{00}) \times (N_{01}+N_{00})} \tag{3}$$

Feature Selection Method Based on Term Unbalanced Degree (TUD). Inspired by the χ^2 statistic method, we assume that each term t and category c are independent of each other. Then the probability of a document which contains term t belonging to each category should be the same. We use Eq. 4 to calculate the unbalanced degree of each term.

$$Unbalance(t) = \frac{\sum_{c \in C} \left| n_{t,c} - ave_t \right|}{DF(t)} \tag{4}$$

t presents a term. c is a category. C denotes the set of categories. $n_{t,c}$ denotes the number of documents which contain term t and do not belong to category c. ave_t denotes the average number of documents which contain term t for each category, and $DF(t)$ is the document frequency of term t. We can calculate each term‘s unbalanced degree, make them a list and sort them in ascending order. After setting a reasonable threshold, we get the category-sensitive feature words.

The kNN Classification Method and Its Improvement

In the field of text categorization, the most commonly used algorithms include the Bayes classifier, the k-nearest neighbor (kNN) classifier, and support vector machine (SVM)[7]. Support vector machine (SVM) is a kind of novel machine learning methods, based on statistical learning theory, which have performed well in solving classification problems[8]. The Bayes classifier is a simple and effective classification method based on probability theory, but its attribute independence assumption is often violated in the real world[9]. The inherent defect remains a problem when applied in text classification. The k-nearest neighbor classifier is a traditional statistical pattern recognition algorithm. It has been studied extensively for text categorization applications, we will introduce it in detail and try to make it more efficent in text categorization.

Vector Space Model and the Traditional kNN. Vector space model is first proposed by G.Salton in 1975[10], and is widely used to represent the text in text auto classification. A document d_i is regarded as a n dimensional vector $(t_{i1}, w(t_{i1}), t_{i2}, w(t_{i2}), , t_{in}, w(t_{in}))$, while t_{ik} denotes the k-th feature, and $w(t_{ik})$ is the weight of the k-th feature, often a funtion of term frequency. The most famous weighting formula is called the tf-idf formula that is $w(t_{ik}) = tf(t_{ik}) \times idf(t_{ik})$ [11]. $tf(t_{ik})$

means the term frequency of the k-th feature word in the document d_i. $idf(t_{ik}) = \log(N / df(t_k))$, and N is the total number of documents. $df(t_k)$ denotes the document frequency of term t_k.

In the traditional k-nearest neighbor classifier, we calculate the similarity between two documents by using the cosine value of the angle between the vector space models of them, which is described as Eq. 5.

$$y(\vec{x}, c_j) = \sum_{\vec{d}_i \in KNN} sim(\vec{x}, \vec{d}_i) y(\vec{d}_i, c_j) - b_j \tag{5}$$

$\vec{x}$ denotes the feature vector of a document to be classified; $\vec{d}_i$ denotes the feature vector of a document in the train set; c_j means a category; $y(\vec{d}_i, c_j) \in \{0,1\}$, when $\vec{d}_i \in c_j$ gets value 1, otherwise gets value 0; b_j is a previously set threshold value of category c_j; $sim(\vec{x}, \vec{d}_i)$ denotes the similarity between the document to be classifed and a training document instance, which can be computed through Eq. 6.

$$\cos < \vec{x}, \vec{d} >= \frac{\vec{x} \bullet \vec{d}}{\left|\vec{x}\right|\left|\vec{d}\right|} \tag{6}$$

The Improved kNN Method Based on Quantified Feature Weight. A large number of experimental results have shown that after the feature dimension reaches a certain value (enough large), it has little impact on the classification result; whatever the feature selection method we choose, the classification evaluation measures are close to 90% using the kNN classification method; the simplest selection method DF has a remarkable performance on high dimensions, almost as well as IG and CHI. On the basis of these facts, we try to simplify the feature weight measures, make the values discrete in their bit mode, and reduce the calculating quantity of text similarity largely.

The measure of feature weight does not use the tf-idf formula directly. We calculate the actual weight of a term by using the tf-idf formula, then quantify it to a 32-bit mode. Commonly an integer in computer has 32 bits, so we use an integer to present a term's weight. In order to simplify the calcultion of text similarity, we determine that the integer can only have 32 different values according to Eq. 7.

$$w(t) = \underbrace{0...0}_{p}\underbrace{1...1}_{q}, 0 \le p \le 32, 0 \le q \le 32, p + q = 32 \tag{7}$$

After calculating each term's weight, we get the new vector space model of a document. Then we can easily computing the text similarity between two documents using Eq. 8.

$$sim(d_i, d_j) = \sum_{t \in T} ones(w_i(t) \wedge w_j(t)) \tag{8}$$

t is a term. T denotes the featuer space. $w_i(t)$ is the weight of term t in document d_i. $ones$ is a function computing the number of 1 in a bit string.

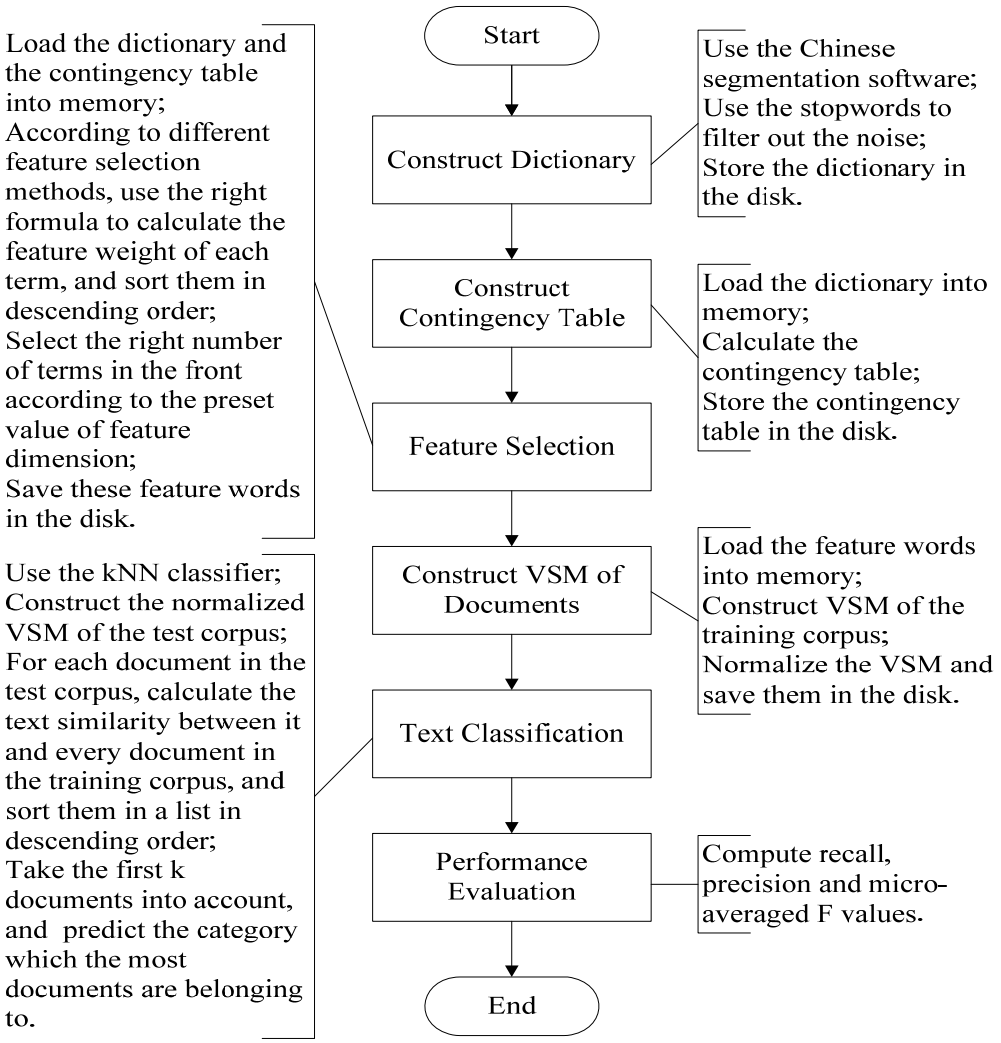
Experiments

Data Collections and the Experimental Process. There is not a standard corpus for Chinese text classification. This paper uses the sohu news articles during June 2012 and July from domestic, international, social, entertainment, etc, totally 18 different channels as the original corpus which contains 304359 articles, and has been divided into 15 categories such as finance, military, culture, and education. After processing the original corpus, we delete the categories which have a few

articles or have little demand of classification, and get a corpus with 10 categories that is entertainment, health, information, education, culture, women, finance, sports, military and others. The training corpus includes 10000 articles, each category having 1000 articles; the test corpus has 5000 articles, with 500 articles in each category. The table mode in database is (ArticleId, ArticleTitle, AritcleText, Categorization).

The experimental process are shown in Fig. 1.

Fig. 1. The Experimental Process of Chinese Text Classification

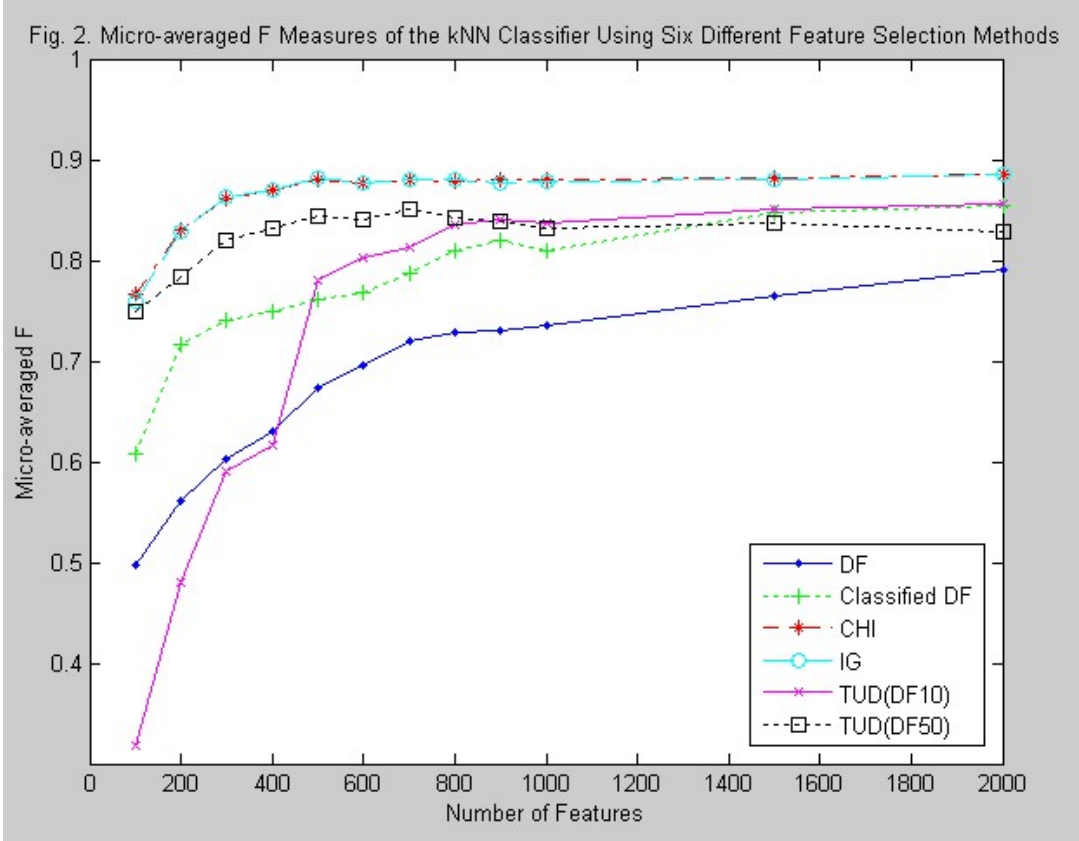


Performance Measures. There are three main performance measures used in text classification, that is recall, precision, and F measures. Recall (R) is the percentage of the documents for a given category that are classified correctly. Precision (P) is the percentage of the predicted documents for a given category that are classified correctly. F measure is to combine recall and precision in some way so that both of them are taken into account when we valuate a classifier‘s performance. It is commonly defined as

$$F = \frac{2PR}{P+R}. \tag{9}$$

These scores are calculated for a series of binary classification experiments, one for each category. Micro-averaged scores on the whole corpus are then produced across the experiments. With micro-averaging, the performance measures are produced by globally adding up all the documents counts across the different tests, and calculating using these summed values.

Results and Discussions. The performances of six selection methods are shown in Fig. 2.



The top two curves present the variation of micro-averaged F values of CHI or IG along with the changes of feature dimension. They perform the best and almost overlap each other. TUD (DF50) is the TUD method when the document frequency of feature words must be above 50. It performs as well as CHI and IG in the beginning, but declines after the feature dimension reaches 800, which means the latter words we choose have an opposite effect. Accordingly, TUD (DF10) means a term's document frequency must be above 10. Its micro-averaged F value begins very low and increases sharply when the number of features is between 300 and 600. That indicates although the feature words we choose in the beginning are sensitive to category, their document frequencies are too low to have a remarkable effect on classification. The last two curves present the performances of DF and classified DF which takes the document frequency of a term into account for each category. They performs poorly and the final micro-averaged F values are close to 0.8.

Table 1 compares the traditional kNN method and the improved method from the perspective of the consumption of time. Method 1 is the traditional kNN method using the cosine value of the angle to calculate text similarity. Method 2 is the improved method based on quantified feature weight.

Table 1 A Comparison of Two Methods

Feature Dimension	Time Consumed by Method 1[sec.]	Micro-averaged F of Method 1	Time Consumed by Method 2[sec.]	Micro-averaged F of Method 2
100	149.766	0.7406	80.843	0.6792
500	396.657	0.8708	98.938	0.7846
1000	729.969	0.8764	123.766	0.798
1500	998.125	0.8824	157.469	0.8092
2000	1265	0.8804	161.781	0.8098
2500	1596	0.8801	176.327	0.8218

From Table 1, we can see that the time consumed by the traditional kNN method increases almost exponentially, while the time consumed by the improved method has a approximately linear relation with the grouth of feature dimension at a little loss of classification accuracy, within 10%.

Summary

We have evaluated four feature selection methods, namely DF, IG, CHI and TUD for Chinese text categorization using the kNN classifier. Based on empirical experiments we find that IG and CHI are the two best methods, and TUD can perform as well as them. In the other way we try to simplify the kNN classifier to make the process of classification more efficient at a little loss of classification accuracy. We quantify feature weights and make the calculation of text similarity very simple. Experiments have confirmed that the improved classifier can cut down a large amount of time without losing much classification accuracy.

References

[1] Qun Liu, Huaping, Zhang and Hongkui Yu: Journal of Computer Research and Development Vol. 41 (2004), p. 1421-1429

[2] Xu Yao, Xiaodan Wang and Yuxi Zhang: Control and Decision Vol. 27 (2012), p. 161-166

[3] C.D. Manning, P. Raghavan: *Introduction to Information Retrieval* (Cambridge University Press, England 2008).

[4] A. Srivastava and M. Sahami: *Text Mining Classification, Clustering and Applications* (CRC Press, the USA 2009).

[5] Songwei Dan, Shicong Feng and Xiaoming Li: Computer Engineering and Applications Vol. 22 (2003), p. 146-148

[6] Yunfei Qiu, Wei Wang and Dayou Liu: Application Research of Computers Vol. 29 (2012), p. 1303-1306

[7] Yong Li: Modem Computer Vol. 15(2012), p. 3-7

[8] Haijun He, Jianfen Wang and Qing Zhou: Computer Engineer Vol. 29 (2003), p. 47-48

[9] Yujin Hu, Xiaoling Zhou and Ling Ling: Computer & Digital Engineering Vol. 32 (2004), p. 28-32

[10] Yong Mao, Xiaobo Zhou and Zheng Xia: Pattern Recognition and Artificial Intelligence Vol. 20 (2007), p. 211-218

[11] Xi Zhou and Mingsheng Zhao: Journal of Chinese Information Processing Vol. 18 (2004), p. 17-23