A Text Classification Method with an Effective Feature Extraction based on Category Analysis

Li Yun Sheng Yan Luan Luan

(Institute of Information Engineering，Yangzhou University，Jiansu Yangzhou 225009)

E-mail: shengyan1985@gmail.com

Abstract: Text classification refers to determine the class of an unknown text according to its content in the given classification system. In order to extract fewer features to express the information in the text as much as possible, the paper analysis the various features' statistical properties and to extract the features in the whole according to Zipf's law firstly; Secondly, based on the statistical analysis of the features' classified information, we compute the contribute of a feature and extract the efficient feature; After that, we improve the traditional TF-IDF formula using category frequencies and put forward a new formula for calculating the feature weight named by TF-IDF-CF; Next, Text classification method is advanced on the base of the first two steps; Finally test the algorithm in the Newsgroup data set. This experiment result proved that feature extraction methods advanced in the paper are reasonable and the formula for calculating the feature weight has higher classification accuracy.

Key word: Feature Extraction; Zipf Law; Category Frequency; Feature Weight

1 Introduce

Web resources, especially text information, expand increasingly with the continuous development of modern web. It costs people more time to organize and manage information. Also it is quite inconvenient to classify so much information manually. Therefore, automatic classification has become an important research domain in data mining with high commercial value. Simply, the main task of text classification[1] is that, classifier construct by given classification system automatically confirms the category of an unknown text based on its content. The most of existing classification system used the vector space model in order to express the set of text, namely, one text is extracted into a feature word sequence, then compute the weight of these features, so the text can be expressed as the weight vector. Finally, classifier can be constructed by dealing with the weight vector space.

Usually the text contains lots of features and the dimension of vector space is high, which costs much time or space of classifying texts. So we need the methods of feature extract or feature select to obtain the more useful and less features without losing any knowledge as much as possible. Wang[2], Feng[3] and Luo[4] improved on the traditional TF-IDF method and Luo advanced a feature extract method based on the word frequency differentia to improve the quality of feature select and the accuracy of text classification. Hu[5] improved CHI formula to express category contribution degree of features. Zhe[6] merged the features with close relations based on many feature extract methods to reduce the feature dimension. Wu[7], Xu[8] and Wang[9] selected the features with high capability to distinguish between categories after analyzing the category information. We can see that, in order to get better feature vector space, there are two ways: 1) Based on statistical properties of every feature, universal or noise feature need be removed in order to reserve the significative feature; 2) Analyzing the category information to obtain the category contribution of every feature can get hold of the classification result better. However, the most of literature[2~9] above are just from one single point of view, namely, improve the classical formula, or just use feature category contribution, or select feature based on word frequency with no further processing. But the whole process need two steps, which are feature extraction and text

classification. If the selected features are suitable to classify, the result will be better, otherwise worse. In the same case, if we just improve the second step, the final classification result will be not satisfied. So we should take two steps into consider together.

The paper first extract features which don't contain the usual or noisy features in the overall situation according to the Zipf[10~13] law; Second, we filter features partially after analyzing the feature category information statistically; Finally, the paper advanced a complete and unitive classification method using category frequency to improve TF-IDF formula after choosing the effective features. In the last paper, we tested on the standard dataset Newsgroup to demonstrate that the feature extraction method advanced in this paper is reasonable and the classification method has a higher accuracy.

2 Backgrounds

2.1 Zipf's Law

In 1932, linguistics expert Zipf[10] found that there are the word frequency in the descending order has the simple relationship that is the frequency of each word is inverse to the constant power of its ranking when he did some research on the English word frequency in Harvard University. In the formal, it can be expressed as below:

$P(r) = \frac{C}{r^{\alpha}}$, this is Zipf's Law. Thereinto, C is a constant bigger than 0, usually is 1, α is called Zipf exponential that is also a constant bigger than 0 and its value is decided by the detail case, in English is about 1. The law shows that in language, the words used frequently are only a small portion of total words while the most of words are used rarely. Its frequency-ranking curve[13] is in figure 1. Its log-log curve corresponding to frequency-ranking curve is a line with slope $-\alpha$ in figure 2.

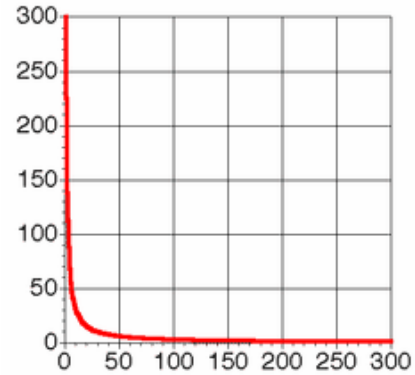

Figure 1 Zipf Law's frequency-ranking curve

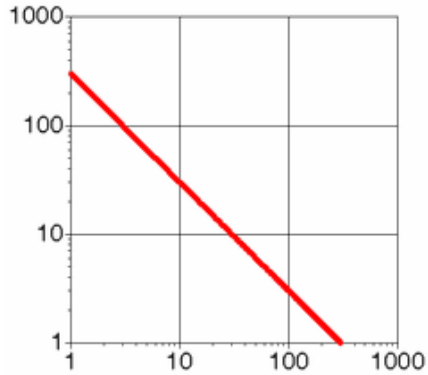

Figure 2 Zipf Law's log-log curve

Zipf Law shows that in English, only a very small number of words have been frequently used, while the vast majority of words have been used rarely. In fact, most of human language has met this law. This paper extract effective feature using this characteristic in text corpus.

2.2 Classical Feature Extract Method

At present there are many feature extract algorithms[5], such as Document Frequency(DF), Information Gain(IG) , Mutual Information(MI), χ^2 (CHI).

1) Document Frequency(DF). Document frequency refers to the number of texts which has certain word in the training dataset. Some terms with low DF must be removed from the original feature space in order to reduce the dimension.

2) Information Gain(IG). The IG formula is below:

$$IG(t) = P(t)\sum_{i=1}^{M} P(C_i \mid t)\log\frac{P(C_i \mid t)}{P(C_i)} + P(\overline{t})\sum_{i=1}^{M} P(C_i \mid \overline{t})\log\frac{P(C_i \mid \overline{t})}{P(C_i)}$$

t standards for a term, P(t) is the probability of text contains t in the training dataset, $P(C_i)$ is the probability of category C_i, $P(C_i|t)$ is the conditional probability of text contains t belongs to the category C_i. $P(\overline{t})$ is the probability of text that don't contain t, $P(C_i \mid \overline{t})$ is the conditional probability of text don't contains t belongs to the category C_i. By all appearances, the bigger IG(t) the term has, the stronger it has the ability to distinguish categories.

3) Mutual Information(MI). MI fomula is:

$$\text{Mutual Info Txt}(t,\ C) = \log\frac{A\times B}{(A+C)(A+B)}$$

A is the number of t contained by text that belongs to C_i, B is the number of t contained by text that don't belongs to C_i, C is the number of text don't contain t belongs to C_i.

4) χ^2 (CHI). χ^2 formula is:

$$\chi^2(t,\ C) = \frac{N(AD-BC)^2}{(A+C)(B+D)(A+B)(C+D)}$$

$$\chi^2_{AVG}(t) = \sum_{i=1}^{m} P(c_i)\chi^2(t,c_i)$$

$$\chi^2_{Max}(t) = \max\nolimits_{i=1}^{m}\{\chi^2(t,c_i)\}$$

A is the number of text contained t and belonged to c, B is the number of text contained t but don't belong to c, C is the number of text don't contain t but belong to c, D is the number of text don't contain t and don't belong to c, N is the number of total texts.

2.3 Feature Weight Computing

In the vector space model, the method for computing feature weight is using TF-IDF, as follow:

$$\omega(t,d) = \frac{TF(t,d)\times\log(IDF(t)+\delta)}{\sqrt{\sum_{t\in d}(TF(t,d)\times\log(IDF(t)+\delta))^2}}$$, TF(t, d) is the frequency of t contained by d, IDF(t) is inverse document frequency, namely, $IDF(t) = \frac{1}{DF(t)}$, δ is an adjustment coefficient, about 0.01 usually. If the term t has a higher document frequency in one text and has a lower document frequency in other text, it can more able to distinguish documents and its weight is bigger. The category frequency can be take into consider because the category information also play a important role in classification as usual.

3 Feature Extract Strategy

The result of feature extraction plays an important role in text classification but the existing feature extraction methods are just carried out not from the global view but a single point. A novel feature extraction method is put forward in the paper, which has two steps are global and partial feature extraction. The former removed the widespread and, in particular, the noise feature from the whole Training Corpus(TC for short) according to the Zipf Law; The latter extract the effective feature among the class after obtaining the category information of each feature. In the whole process, beside of the word frequency(TF for short) and document frequency(DF for short), the Overall Frequency(OF for short) which is used for the global feature extraction and the Category Frequency(CF for short)which is use for the partial feature extraction need be compute.

3.1 Global Feature Extraction

Definition 1: In the whole training corpus(TC), the frequency of feature t in TC is called the Overall frequency(OF), defined as: $OF(t) = \frac{\text{the times of t appears in TC}}{\text{the number of all words in TC}}$.

After computing all features' OF, a frequency-rank curve obtained according to all OF sorted by size is accord with the Zipf Law in most case. So we can enactment two threshold: the overall percentage of low-frequency feature(z-low) and the overall percentage of high-frequency feature(z-high) used for removed the lower z-low*number of all features and the higher z-high*number of all features separately, that is, reserve the features whose rank is between the two thresholds. In addition, after computing the document frequency(DF) of each feature, if the feature whose DF is less than certain threshold, it must be removed from the original feature space in this step. The total procedure can be described by Algorithm 1 below:

Algorithm 1: Overall_Selected(TC, z-low, z-high, df-threshold)

Input: the training corpus: TC, the overall percentage of low-frequency feature: z-low, the overall percentage of high-frequency feature: z-high, DF threshold: df-threshold

Output: the effective feature set: TSet

Description:

Step 1: Obtain all feature set TSet and compute the global frequency of each feature t based on TC

Step 2: Sort TSet according to the frequency from high to low

high_OF = the first $\lfloor |TSet| \times z - high \rfloor$ features in TSet # the overall high-frequency feature want to be removed

low_OF = the last $\lfloor |TSet| \times z - low \rfloor$ features in TSet # the overall low-frequency feature want to be removed

Get rid of the feature in high_OF or low_OF from TSet

Step 3: # Remove the feature in high_OF or low_OF from document vector space and compute the DF of each feature

```
for each file in TC:
    for each term in file:
        if term in high_OF or term in low_OF:
            remove term from file
    for each t in TSet:                # compute DF
```

if feature t is in file:

increase the frequency count of t of 1

Step 4: # Get rid of the term with low DF

Convert and get each t's DF

Remove the t whose DF is less than df-threshold and return the final TSet

Algorithm 1 is used for getting rid of highly or lowly frequent features from the whole feature space in the overall situation and can obtain the most of effective features and reduce the most of ineffective features.

3.2 Partial Feature Extraction

Definition 2: In the whole training corpus(TC), the category set is $C=(c_1,c_2,...c_m)$, m is the number of category, every text in TC has a category info $c_i \in C$, the frequency of feature t appears in the text belongs to $c_i \in C$ is called as the category frequency(CF) of t, defined as: $CF(t,c_i)=\frac{\text{the times of t appears in } c_i}{\text{the number of all words in } c_i}$.

If the feature t appears in the lesser categories and also have higher category frequency, it shows that the feature t has a special meaning for these categories which must be kept down; But if the feature t appears in the most categories and the category frequencies are very close, it shows that the feature t must be removed from feature space because of lower ability to distinguish categories. In the end, the features have bigger differentia among the category frequencies will be kept down. The detail process is below.

Algorithm 2: Category_Selected(TC，cf-rate，cf-threshold)

Input: the training corpus: TC, category emergence rate: cf-rate, word frequency differentia threshold: cf-threshold

Output: the effective feature set: TSet

Description:

Step 1：# CF Compute the category frequency CF of feature t

for each file in TC:

Obtain the category of the file

for each t in file:

Increase CF(t,c) of 1

Convert and compute the CF of t

Step 2：CF_min = the total number of category*cf-rate # the threshold of inner-category frequency

for each t in CF:

if |the number of category of feature t | >= CF_min and _IsInThreshold(t's CF，cf-threshold): # If t appears in the most of category with similar CF, t will be removed

Remove t

Step 3：Return the final TSet

_IsInThreshold(cf，cf-threshold)

Input: a series of category frequency value cf, word frequency differentia threshold: cf-threshold

Output: if cf value fluctuation is in cf-threshold, return True, otherwise return False.

Description:

Step 1: # Find the maximum and minimum of cf

```
cf_min is minimum of cf, initialized with the first cf
cf_max is maximum of cf, initialized with the first cf
for each otc in the rest of cf:
    if otc < tc_min:
        tc_min = otc
        continue
    if otc > tc_max:
        tc_max = otc
```

Step 2：# If cf value fluctuation is in cf-threshold, return True, otherwise return False.

```
if tc_max-tc_min < CF_THRESHOLD:
    return True
return False
```

After the algorithm 2, the feature set with good ability to distinguish categories will be obtained, which has fewer number of features. In this way, after two steps above, we are able to maximize to reduce the dimension of features in the case of retaining the original text information as much as possible, thereby enhancing the efficiency of automatic text classification.

4 Text Classification Algorithm with an Effective Feature Extraction based on Category Analysis

4.1 The Feature Weight Formula based on Category Frequency

The traditional formula TF-IDF shown in 2.3 is widely used in many classification systems or information retrieval systems in practice because of its simple computation. But the TF-IDF just takes TF and IDF into consideration without category frequency which is always important to text classification. Therefore, the paper advanced a improved feature weight formula, named as TF-IDF-CF:

$$\omega(t,d)=\frac{TF(t,d)\times\log(IDF(t)+\delta)\times\max\limits_{c\in C}CF(t,c)}{\sqrt{\sum\limits_{t\in d}(TF(t,d)\times\log(IDF(t)+\delta)\times\max\limits_{c\in C}CF(t,c))^2}}$$

, TF(t, d) is the frequency of t in document d, IDF(t) is the inverse document frequency, namely, δ is an adjustment coefficient, about 0.01 usually. CF(t, c) is the category frequency in category c which is the similar to algorithm 2. Because a feature t may have many category frequencies, the weight of feature t should use the max of category frequencies in order to reflect its category contribute. If the feature t has a higher category frequency, its category contribute is bigger, otherwise smaller. From the overall look, the new formula shows that a feature t with higher category frequency has a higher weight and is more useful.

4.2 Improved kNN Classification Algorithm

We advanced an text classification algorithm which is improved from the kNN[14] after extracting the effective features by algorithm 2. The main idea is that, after extracting the useful features by algorithm 2 and computing the feature weight by formula TF-IDF-CF advanced in section 4.1, a target document vector space is build. Then a new document vector will be obtained from an unknown document in the test corpus the same way. At last, the K neighbors of new document will be found out then the category label obtained by counting these neighbors is as the

category of the new document. Hereinto, the similarity of document vector is measured by the cosine similarity[15]. The detail algorithm is below:

Algorithm 3: Classify(TC, Test Corpus, TSet, K)

Input: the training corpus: TC, the test corpus: Test Corpus, the effective feature set: TSet, the number of neighbors: K

Output: the category of the unknown document in Test Corpus

Description:

Step 1: # Compute the weight of t in TSet obtained in Algorithm 2

for each t in TSet:

for each d in TC:

Compute w(t, d) according to TF-IDF-CF formula

Step 2: # Classify the document in Test Corpus

for each file in Test Corpus:

Compute the effective features in file

Ergodic the whole TC, compute the similarity between file and the training corpus using cosine similarity, then find the K most adjacent documents.

From the K documents, choose the largest category of document as this file's category.

5 Experiments and Analysis

5.1 Dataset

The paper adopt the international universal dataset named NewsGroup[16], which is consist of 19997 messages placard in Usenet. These messages belongs to twenty different news groups which have 1000 messages. Each news group corresponds to one text category. We adopt the whole data and chose 12000 texts as the training dataset and the rest is as test data.

5.2 Performance Evaluation Criterion

There are two ordinary evaluation criterions: recall(r) and precision(p) to evaluate the classifier.

$$r = \frac{a}{a+c},\ if\ a\ +\ c\ >\ 0;\ otherwise\ r\ =\ 1,$$

$$p = \frac{a}{a+b},\ if\ a\ +\ b\ >\ 0;\ otherwise\ p\ =\ 1,$$

a is the number of texts classified by the category correctly, b is the number of texts classified by the category wrong, c is the number of texts belonged to the category but classified by the other category.

The other evaluation criterion is F-guideline, defined as: $F_{\beta}(r, p) = \frac{(\beta^2 + 1)pr}{\beta^2 p + r}$. β is used for measuring the importance of r and p.

5.3 Experiments Result and Analysis

The arguments set up in the experiments are as follows:

The overall percentage of low-frequency: z-low is 1%;

The overall percentage of high-frequency: z-high is 1%;

DF threshold: df-threshold is 1%;

Category emergence rate: cf-rate is 90%;

Word frequency differentia threshold: cf-threshold is 0.0005;

The adjustment coefficient in TF-IDF-CF: δ is 0.01;

The number of neighbors: K is 500;

In the F-guideline, β is 1.

We got 88336 terms totally and have about 70000 effective features after the overall and partial feature extraction. Each document contains 56 effective features on average.

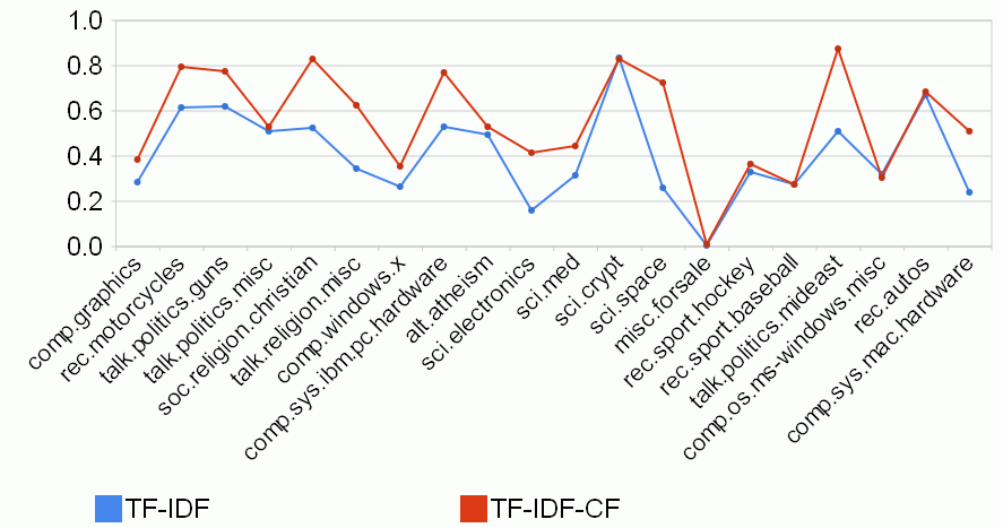

Figure 3 the result of recall(r)

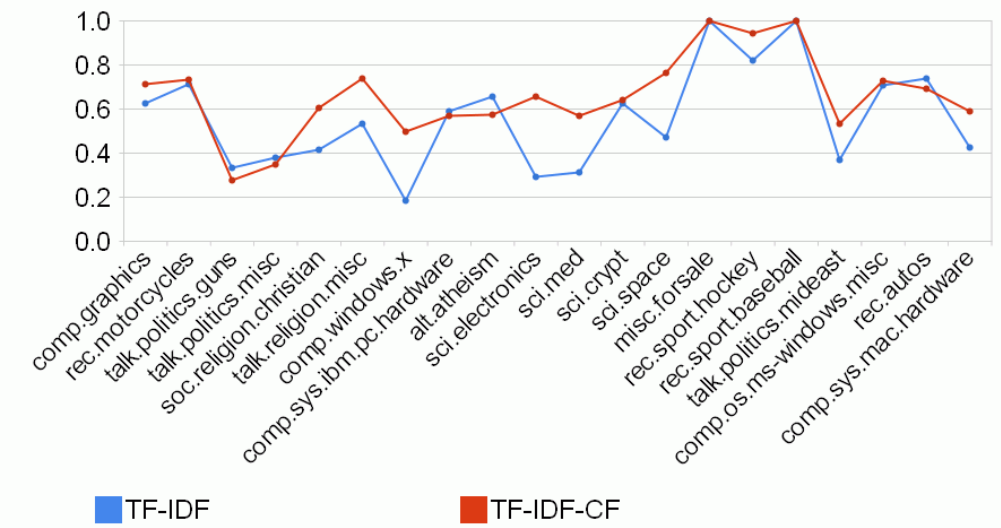

Figure 4 the result of precision(p)

Figure 3 and 4 shows the different result of recall and precision separately by two methods TF-IDF-CF and TF-IDF. We can see in most categories, the methods advanced in the paper is better than the traditional methods.

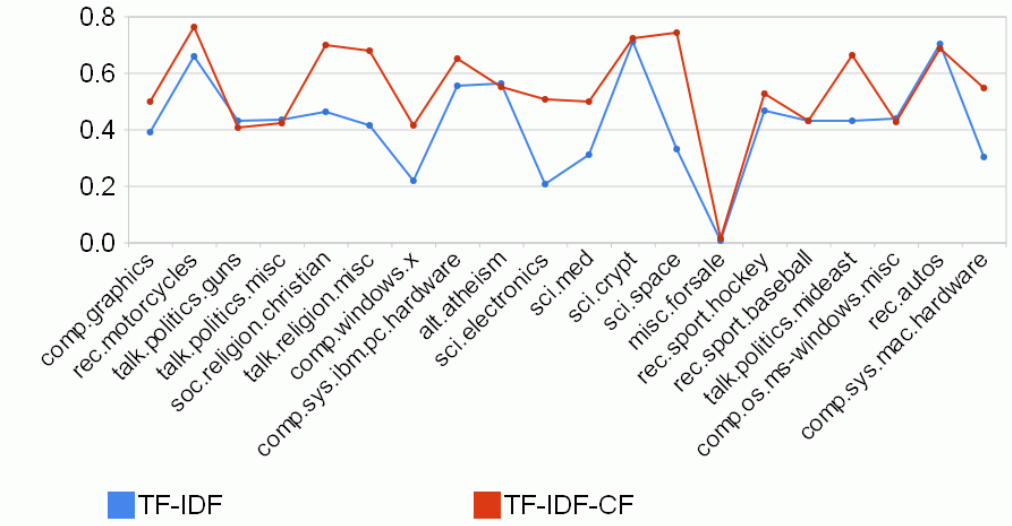

Figure 5 the result of F-guideline

Figure 5 shows the result of F-guideline which is the integrate evaluation of recall and precision. Also we can get better result than traditional algorithm.

6 Summarize

This paper extracts the effective features from two level, that is overall and partial feature extraction. After obtaining the effective features, a new weight measurement named TF-IDF-CF is put forward based on the category frequency. Finally, the result of experiment shows that the feature extraction method advanced in the paper is more effective and can get higher classification accuracy.

Reference:

[1] Ronglu Li, Jianhui Wang, Xiaoyun Chen. Using Maximum Entropy Model for Chinese Text Categorization[J]. Journal of Computer Research and Development, 2005,42(1):94~101

[2] Meifang Wang, Peiyu Liu, Zhengfang Zhu. Feature selection method based on TFIDF. Computer Engineering and Design,2007,12,Vol.28,No.23

[3] Changyuan Feng, Jiexing Pu. Research about Algorithm of Web Text Feather Selection. Application Research of Computers,2005,7,Vol.22,No.7:36~38

[4] Xin Luo, Deling Xia, Puliu Xia. Improved feature selection method and TF-IDF formula based on word frequency differentia. Journal of Computer Application,2005,9,Vol.25,No.9

Improved feature selection method and TF-IDF formula based on word frequency differentia

[11] Zipf Curves and Website Popularity. http://www.useit.com/alertbox/zipf.html

[5] Shengbo Hu, Xiyu Liu. The Comparison and Improvement of Feature Selection Method in Web Page Classification. Journal of Shandong Normal University(Natural Science),2008,9,Vol.23,No.3

[6] Li Zhe, Shiyong Zhang. A Text Feature Selection Method Based on Integration and Combination.. Computer Applications and Software,2008,10,Vol.25,No.10

[7] Di Wu, Yaping Zhang, Fuliang Yin, Ming Li. Feature Selection Based on Class Distribution Difference and VPRS for Text Classification. Journal of Electronics & Information Technology, 2007,12,Vol.29,No.12

[8] Yan Xu, Jintao Li, Bing Wang, Chunming Sun. A Category Resolve Power-Based Feature Selection Method. Journal of Software,Vol.19,No.1,2008,1,pp.82-89.

[9] Qiang Wang,Yi Guan,XiaoLong Wang,Zhiming Xu. A Novel Feature Selection Method Based on Category

Information Analysis for Class Prejudging in Text Classification. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.1A, January 2006.
[10] G.Zipf,Human Behavior and the Principle of Least-Effort (Cambridge, Mass, 1949; Addison-Wesley,1965);
[11] Xiaojin Zhu.CS838-1 Advanced NLP: Words, Zipf's Law, Miller's Monkeys. 2007. http://pages.cs.wisc.edu/~jerryzhu/cs838/words.pdf
[12] Yanjiang Ji. Zipf Law and its Application. http://www.qiji.cn/eprint/abs/4.html
[13] Xiaopu Han. Zipf Law in Web information Seach.. http://www.qiji.cn/eprint/abs/840.html. [8]
[14] Ning Zhang, Ziyan Jia, Zhongzhi Shi. Text Categorization with KNN Algorithm. Computer Engineering. 2005, 4, Vol. 31, No. 8(171~173)
[15] Yiming Y. An evaluation of statistic approaches to text categorization[J]. Information Retrieval, 1999, 1(1/2):69-90.
[16] NewsGroup. 1999. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html