An Improved Feature Selection Method in Chinese Text Categorization

Yuxia Sun\textsuperscript{1,2}, Weiguang Qu\textsuperscript{1,2,*}, Junsheng Zhou\textsuperscript{1,2}, Xuri Tang\textsuperscript{3}, Ying Di\textsuperscript{1,2}, and WeiCheng Wu\textsuperscript{1,2}
\textsuperscript{1}School of Computer Science and Technology
Nanjing Normal University
No.1, Wenyuan Road, Nanjing 210046, P. R. China
\textsuperscript{2}Jiangsu Research Center of Information Security and Privacy Technology
Nanjing Normal University
No.1, Wenyuan Road, Nanjing 210046, P. R. China
eve_hello@126.com; *Corresponding author: wgqu@njnu.edu.cn
\textsuperscript{3}College of Foreign Studies
Wuhan Textile University
No.1, Fangzhi Road, Wuhan 430073, P.R. China

Received May 2011; revised July 2011

ABSTRACT. This paper reports a method which has improved performance in feature selection in Chinese Text Categorization. The paper first uses the improved information gain (IG) to select the initial features, and experiments with two methods for feature selection. The first method chooses the top $n$ features of all classes as vector space, while the second method chooses the top $k$ features for each class $c_i$ ($i=1,\ldots,m$), where $m$ is the number of class and $k$ is the result of $n$ dividing $m$. For both methods, the Support Vector Machine (SVM) is used as classifier. The result shows that we have achieved great improvement by optimizing parameter. With the first method, the system gets best F-score 97\% with parameter optimization and 93.1\% without parameter optimization. The experiments also show that the second method is able to get better performance with fewer feature number. Compared with RON system, the methods in the paper is about 4.1\% higher in F-score.

Keywords: Text Categorization, Improved Information Gain, SVM, Parameters Optimization

1. Introduction. Over the recent years, text categorization has become one of the key techniques for organizing online information. It can be used to organize document databases, filter spam emails, or learn users’ reading preference. However, as hand-coding text-categorization is costly and thus impractical, automatic text categorization is needed.

Text categorization is to assign one or multiple categories from a predefined category set to documents. The key technologies include text representation, feature selection and classifier. The common feature selection methods are Information Gain (IG), $\chi^2$ (CHI), document frequency (DF), and mutual information (MI) etc. Among those methods, $\chi^2$
and IG are most popular\cite{2}. In recent years, the main-stream classifiers are from the field of machine learning, including SVM, kNN, Naive Bayes and Neural Network, etc. Among them, SVM and KNN perform better\cite{2}.

Text representation is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. At present, the most popular method to represent documents is Vector Space Model (VSM). VSM is proposed by Salton G in the 1960s\cite{5}, which uses the bag of words to represent documents, ignoring the sentence structure and grammar. Though VSM loses much information of documents, it simplifies the problem effectively and makes text categorization possible.

However, the studies on Chinese text categorization are less sufficient, compared with English text categorization. And Chinese text has its own characteristic. For example, we need to segment the sentence for further process.\cite{7} studies term weighting factors’ effects on text categorization.\cite{8} and\cite{9} concentrate on the study of feature weight and have improved the performance of text categorization.\cite{1} uses the improved IG to select features and applies Naive Bayes on text classification.\cite{3} uses the resource-optimizing neural networks for text categorization and performs well.\cite{2} adjusts the VSM and Naive Bayes classifier using the weight adjustment.

This paper uses two methods to select features with improved IG and optimizes parameters of SVM classifier automatically to find the most suitable Model. The results have showed that our experiment performs well.

The remainder of this paper is structured as follows. We first present overview of text category in section 2 and then describe feature selection based on improved information gain in section 3. Section 4 describes the data sets and evaluation methodology used in our experiments, and the result is also discussed. In the end, we conclude our work in section 5.

2. Text Categorization. Text categorization is to automatically assign the fixed semantic categories to documents. Each document can fall into one, or more, or none of the categories. This paper focuses on the problem of documents with one category, that is to say, we assume that each document just belongs to one category.

We fist use stop words list to filter noise words which have no content meanings. And then we can use feature selection methods to abstract useful features, here we use improved information gain which will be introduced in the section 3. Next, texts are transformed into VSM. VSM considers that documents consist of a collection of words \((t_1, t_2, \ldots, t_n)\), each word is assigned a weight \(w_i\) \((i=1,2,\ldots,n)\). Documents are mapped into a vector which is in the vector space composed of a set of word vectors. Each document is regarded as a feature vector: \(d_i = \{t_1:w_1, t_2:w_2, \ldots, t_n:w_n\}\), where \(t_k\) indicates words, \(w_k\) indicates the weight of \(t_k\) in the document \(d_i\).\cite{1}

Weight describes the importance of words in documents\cite{5}. The most popular computing method of feature weight is \(tf*idf\). This paper uses \(tf*idf\) and normalizes feature vector as formula 1\cite{6}:

\[
weight(t) = \frac{\sum_i tf(t) * \log[ idf(t) + 0.01]}{\sum_i [tf(t) * \log[ idf(t) + 0.01]]^2}
\]

where \(tf\) is Term Frequency, which indicates the frequency of a word in a document ;\(idf\) is
Inverse Document Frequency, which indicates the distribution of a word in the documents collection, and is calculated by formula 2.

\[
\text{idf}(t) = \log \frac{N}{n}
\]

(2)

where \( N \) indicates the number of all documents, and \( n \) indicates the number of documents that includes word \( t \).

Finally, we will train the classifier from the training corpus automatically and then apply the classifier to our test corpus. The classifier model used here is Support Vector Machine (SVM) which is very popular in text categorization. This paper chooses RBF as the kernel of SVM, at the same time, we optimize parameter of RBF to obtain a more suitable SVM model.

3. Feature Selection Based On Improved Information Gain.

3.1. Information Gain. Information gain (IG)\(^2\) is an entropy-based assessment methods and is widely used in the field of machine learning. Information gains represents the entropy difference before and after a certain feature occurs in the given document. The basic idea to select features is as follows: first we calculate IG of each feature, and sort features in descending order according to the value of IG, then choose features whose values are larger than the predefined threshold or rank in the top predefined number. Given feature number \( n \), we use two methods to select features. Method1 chooses the top \( n \) features of all classes as vector space. Method2 chooses the top \( k \) features for each class \( c_i \) \((i=1,…,m)\), here \( m \) is the number of class and \( k \) is the result of \( n \) dividing \( m \), that is to say, each class has the same number of feature. We use formula 3 to calculate IG.

\[
\text{IG}(t) = - \sum_{c=1}^{m} p(c) \log p(c) + p(t) \sum_{c=1}^{m} p(c | t) \log p(c | t) + p(\bar{t}) \sum_{c=1}^{m} p(c | \bar{t}) \log p(c | \bar{t})
\]

(3)

where \( p(c) \) indicates the probability of texts belong to class \( c \) in the whole corpus, \( p(t) \) indicates the probability of texts including term \( t \), \( p(c | t) \) indicates the conditional probability of texts belong to class \( c \) while texts include term \( t \), \( p(\bar{t}) \) indicates the probability of texts not including term \( t \), \( p(c | \bar{t}) \) indicates the conditional probability of texts belong to class \( c \) while texts don’t include term \( t \), \( m \) indicates the number of class.

3.2. Improved Information Gain. The shortage of IG is that it considers the absence of features. Though the absence of features will contribute to text categorization, there is experiment that proves the contribution is less than the distraction it takes.

So, this paper uses the improved IG to select features \(^2\). The improved IG removes the contribution of feature’s absence, at the same time, adds concentration information (considering the contribution of features to some one category) and distribution information (considering the degree of association between categories and features).

Concentration Information: features that can effectively represent category should appear in documents of the specific category instead of appearing in documents of all categories equably. A feature, which just comes up in few categories and doesn’t appear in the other categories, is a better feature. \( \text{CI}(c,w) \) indicates the contribution of word \( w \) to the category \( c \).

\[
\text{CI}(c,w) = \frac{\text{Num}(c,w)}{N(w)}
\]

(4)

where \( \text{Num}(c,w) \) indicates the number of documents including word \( w \) in category \( c \), \( N(w) \)
indicates the number of documents including word w in the whole training corpus.

Distribution Information: In documents of a particular category, those feature words which occurs evenly enjoy a high association with the category. That is, the more frequent and more scattered a word is found in the documents of a category, the more information it bears about the category. Thus it is more reliable for categorization. DI(c,w) indicates the association of class c and word w.

\[
\text{DI}(c,w) = \frac{\text{Num}(c,w)}{M(c)}
\]

(5)

where M(c) indicates the number of documents in category c.

IG’(w) indicates the improved IG of word w.

\[
\text{IG}'(w) = \sum_{i=1}^{k} CI(ci,w) * DI(ci,w) * p(ci,w) \log \frac{p(ci,w)}{p(ci) * p(w)}
\]

(6)

where p(ci,w) indicates the probability of documents including word w in category ci, p(ci) indicates the probability of documents in category ci, p(w) indicates the probability of documents including word w.

4. Experiments.

4.1. Data Sets. This paper uses the corpus provided by the Computer Information and Technology Department of Fudan University \[10\], which contains 20 categories of documents. For experimental comparison, we choose the categories that are used in the system RON \[3\] and the same scale of documents. We select a total of 2,370 documents of Art, Space, Computer, Environment, Agriculture, Economy, and Sports, and select the training corpus and testing corpus randomly according to the ratio of 2:1 in each category. Then we use ICTCLAS \[11\] of Chinese Academy of Sciences for word segmentation and POS tagging. Finally, we obtain 20,023 feature items, and select difference feature number to have comparison.

4.2. Evaluation Measures. Recall (number of categories correctly assigned divided by the total number of categories should be assigned) and precision (number of categories correctly assigned divided by total number of categories assigned)\[4\]. \(r_c\) is recall of category \(c\), and \(p_c\) is precision of category \(c\).

\[
r_c = \frac{a}{a + b}
\]

(7)

\[
p_c = \frac{a}{a + d}
\]

(8)

Where a indicates the number of documents which belong to category \(c\) and at the same time is assigned to \(c\), b indicates the number of documents which belong to category \(c\) but is assigned to other category, d indicates the number of documents which don’t belong to category \(c\) but is assigned to \(c\).

Here recall and precision just evaluate for some one category. In order to evaluate the whole corpus, we use Macro-Averaging. Macro-Averaging is the arithmetic average of each category’s performance. \(\bar{r}\) is Macro-Averaging recall and \(\bar{p}\) is Macro-Averaging precision.
\[
\begin{align*}
\bar{r} &= \frac{1}{|C|} \sum_{c \in C} r_c \quad (9) \\
\bar{p} &= \frac{1}{|C|} \sum_{c \in C} p_c \quad (10)
\end{align*}
\]

Where \(|C|\) indicates the number of the total categories, \(r_c\) indicates recall of category \(c\), \(p_c\) indicates precision of category \(c\).

The performance measure used here is F-score which is the harmonic mean of recall and precision. \(F\)-score(c) reflects performance of text categorization in category \(c\), and \(\bar{F}\) reflects a system’s overall performance.

\[
F\text{-score}(c) = \frac{2 \times r_c \times p_c}{r_c + p_c}
\]

\[
\bar{F} = \frac{2 \times \bar{r} \times \bar{p}}{\bar{r} + \bar{p}}
\]

4.3 Results.

4.3.1. **Results of SVM with Different Parameter.** Here, we use Libsvm\textsuperscript{[14]} as our classifier, and choose RBF as the kernel of SVM. RBF function is as formula 13.

\[
RBF = e^{-\gamma \|u - v\|^2}
\]

This paper has optimized parameter \(\gamma\) in [0,1] and names this method MWO. At the same time, we also have experiment with the default value of parameter \(\gamma\) which is 0 in Libsvm and name it MWD. From table 1, we can see that F-score has achieved great improvement after optimizing parameter \(\gamma\). After optimizing, the best result of system is 97\% which is 3.9\% higher than the best result without optimizing. What’s more, F-score of MWO is much higher than MWD all along, no matter how many features there are. So, we can see that it is an efficient way to optimize the parameters in the model, and it can bring unexpected results.

**Table 1. F-score of MWD and MWO**

<table>
<thead>
<tr>
<th>Feature number</th>
<th>14</th>
<th>49</th>
<th>210</th>
<th>350</th>
<th>490</th>
<th>798</th>
<th>1498</th>
<th>3500</th>
<th>4998</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWD</td>
<td>55.7%</td>
<td>87.1%</td>
<td>90.9%</td>
<td>91.1%</td>
<td>92%</td>
<td>93.1%</td>
<td>92.3%</td>
<td>91.4%</td>
<td>91.8%</td>
</tr>
<tr>
<td>MWO</td>
<td>60.2%</td>
<td>92.4%</td>
<td>95.9%</td>
<td>96.2%</td>
<td>96%</td>
<td>97%</td>
<td>96.4%</td>
<td>96.7%</td>
<td>96.9%</td>
</tr>
</tbody>
</table>

Table 2 has shown \(\gamma\) of MWO with different feature numbers, we can see that 0.7778 and 0.5556 are appropriate values for our corpus.

**Table 2. \(\gamma\) of MWO with different feature number**

<table>
<thead>
<tr>
<th>Feature number</th>
<th>14</th>
<th>49</th>
<th>210</th>
<th>350</th>
<th>490</th>
<th>798</th>
<th>1498</th>
<th>3500</th>
<th>4998</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\gamma)</td>
<td>0.4444</td>
<td>0.7778</td>
<td>0.5556</td>
<td>0.5556</td>
<td>0.7778</td>
<td>0.5556</td>
<td>0.7778</td>
<td>0.5556</td>
<td>0.7778</td>
</tr>
</tbody>
</table>
Now, we list the top 7 key features for each class (Table 4). Those features are appropriate indeed.

<table>
<thead>
<tr>
<th>Classes of Texts</th>
<th>Main Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Art</td>
<td>创作 (Create), 作品(Works), 文学(Literature), 艺术(Art), 审美(Appreciation of the beauty), 文艺(Literature and art), 美学(Aesthetics)</td>
</tr>
<tr>
<td>Space</td>
<td>航空(Aviation), 航天(Space flight), 发动机(Engine), 宇航(Astronavigation), 精度(Precision), 动力学(Kinetics), 飞行(fly)</td>
</tr>
<tr>
<td>Computer</td>
<td>计算机科学(Computer science), 算法(Algorithm), 软件(software), 集合(Assemble), 存储(Storage), 分布式(Distributed), 人工智能(Artificial Intelligence)</td>
</tr>
<tr>
<td>Environment</td>
<td>浓度(Density), 污染(Pollute), 污染物(Pollution), 排放(Discharge), 测定(Assay), 大气(Atmosphere), 离子(Ion)</td>
</tr>
<tr>
<td>Agriculture</td>
<td>农村(Country), 农业(Agriculture), 农民(Peasant), 农产品(Agriculture products), 耕地(Plantation), 农户(Peasant household), 粮食(Food)</td>
</tr>
<tr>
<td>Economy</td>
<td>资本(Capital), 经营(operate), 货币(Currency), 价格(Price), 经济运行(Economic operation), 投资(Investment), 金融(Finance)</td>
</tr>
<tr>
<td>Sports</td>
<td>体育(Sports), 身体(Body), 比赛(Competition), 竞技(Athletics), 锻炼(Exercise), 运动(Take exercise), 运动员(Sports man)</td>
</tr>
</tbody>
</table>

4.3.2. Results of Two Feature Selection Methods. Here, given feature number $n$, we use two methods to select features. Method1 chooses the top $n$ features of all classes as vector space. Method2 chooses the top $k$ features for each class $c_i$ ($i=1,...,m$), here $m$ is the number of class and $k$ is the result of $n$ dividing $m$, that is to say, each class has the same number of features.

In order to check the performance of these two methods, we choose the different numbers of feature to have comparison. Table 3 shows results of two methods in choosing features while optimizing parameters of RBF in SVM. We can find that method2 obtains
better result with fewer features, method1 gets best F-score 97% when features number is 798, while method2 gets the same best score when it’s 490.

When features number is 14, F-score of method2 is about 26.4% higher than method1, and the difference is great. So, we check the top 14 features in method1. To our great surprise, the distribution of each class’s features is imbalanced deeply. Features of class “Art” account for 5/7, and class “Agriculture” for 2/7, class “Environment” and “Sports” for 1/7 respectively, features of other classes do not come up. However, though each class has the same number of texts in our corpus, the length of texts in class “Art” is much longer, which lead to the imbalanced distribution of features. With the increasing of feature’s number, our result shows a smoothing trend and performs well, both with method1 and method2. In a word, the balanced feature distribution is better for text categorization, when feature number is relatively small. And the performances of balanced feature distribution and imbalanced feature distribution are more similar with the increasing feature number.

4.3.3. Compared with Other System. Our system is named IIG here and RON is Zhang’s system [3]. Table 5 has showed the result of our system and the best result of RON system. RON system gets the best result when the number of features is 3000. And our system gets the best result with the top 490 features using method2, which is named IIG(490) in table 3, at the same time, we also list the result of our method2 with 2996 feature number, which is named IIG(2996) in table 5. From the table 5, we can see that our system does better in the most classes of the texts. Specially in the class of Environment, our system gets a great improvement, whose F-score is 96.7% with 490 features and 98.4% with 2996 features, while the result of RON system is 87.1%. The best F-score of our overall system is 97% which is about 4.1% higher than RON’s result.

<table>
<thead>
<tr>
<th>Classes of Texts</th>
<th>F-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RON</td>
</tr>
<tr>
<td>Art</td>
<td>98.4</td>
</tr>
<tr>
<td>Space</td>
<td>90.9</td>
</tr>
<tr>
<td>Computer</td>
<td>94.33</td>
</tr>
<tr>
<td>Environment</td>
<td>87.1</td>
</tr>
<tr>
<td>Agriculture</td>
<td>92.3</td>
</tr>
<tr>
<td>Economy</td>
<td>95.5</td>
</tr>
<tr>
<td>Sports</td>
<td>91.5</td>
</tr>
<tr>
<td>Macro-Averaging</td>
<td>92.9</td>
</tr>
</tbody>
</table>

5. Conclusions. This paper first uses the method of improved information gain (IG) to compute IG of each feature and uses two methods to choose features. And then, we use SVM as our classifier. The RBF is chosen as the kernel of SVM, and the parameters of the RBF are automatically selected in a range. The result shows that we have achieved great improvement by optimizing parameter. When choosing top 798 features, our system gets best F-score 97% with parameter optimization and 93.1% without parameter optimization, which both use method1. And among two methods of selecting features, method2 can get better F-score with fewer features. Method1 gets the best F-score 97% when features number is 798, however method2 gets the same best result when 490. That is to say, the
balanced features distribution is good for text categorization. At last, we compare the best result of our method2 with the best result of RON system, and our F-score is 97% which is about 4.1% higher than RON system’s.

In order to check whether our method is appropriate for different Chinese corpus, we will apply it to other Chinese corpus such as TanCorp[12], the corpus of sogou[13] and so on. Certainly more parameters of SVM will be optimized to get the better model.

Acknowledgment. This work is partially supported by Chinese National Fund of Natural Science under Grant 60773173, 61073119, and Chinese National Fund of Social Science under Grant 11CYY030 and Grant 10CYY021, Jiangsu Province Fund of Natural Science under Grant BK2010547 and the Fund of Social Science of the Ministry Department in Jiangsu Province under Grant 06SJB71007.

REFERENCES