International Journal of Hybrid Information Technology
Vol.9, No.2 (2016), pp. 47-56
http://dx.doi.org/10.14257/ijhit.2016.9.2.04

Automatic Identification of Chinese Dirty Word Texts

Xiaoxu Zhu, Peide Qian

School of Computer Science & Technology, Soochow University, Suzhou 215006, China
xiaoxzhu@suda.edu.cn, pdqian@suda.edu.cn

Abstract

As non-formal language, texts containing dirty words are widespread in Web reviews. Due to their bad effects on users of communication, it is essential to perform automatic analysis on Chinese texts containing dirty word. In this paper, we first crawled over millions of evaluating sentences which contain a lot of dirty words from the Web. Second, we manually annotated 40 typical dirty words with weights. And then proposed a machine learning-based approach for collecting dirty word texts corpus. Overall, more than 6000 sentences were collected from the huge amount of Web reviews to form a corpus on Chinese texts containing dirty words. With the corpus, we present SVM (Support Vector Machine) and ME (Maximum Entropy) classifiers to automatic detect Chinese texts containing dirty words. Empirical studies demonstrate that the SVM and ME classifiers are both effective for this task and the recall and precision are both over 97%.

Keywords*: dirty word texts, corpus, text classification, automatic identification*

1. Introduction

Recently, online forum, blog and micro-blog platforms become more and more popular with the developing of Web 2.0 and people have changed their roles from accepting information passively into manufacturing information actively. A large number of people take part in producing information on the Web through online forum, blog and micro-blog platforms. Under this environment, information is under-going a veritable explosion of growth at this moment. Therefore, it becomes increasingly important to automatically analyze and mine these information.

As a non-formal language, texts containing dirty words are widespread in modern society. Briefly, we consider the texts containing dirty words as dirty word texts. Dirty word texts are popularly used by people to express their *angry* emotion for insulting, abusing, or assaulting others. Because it often needs no real name in many network occasions, people can irresponsibly use dirty word texts. However, dirty word texts have bad effects on user communication and obviously hurt the network civilization.

Therefore, automatic identification of dirty words texts becomes an essential work in practice. In this paper, we focus on constructing a high-quality and large scale corpus on dirty word texts with active machine learning technologies, and then the corpus are employed to perform automatic identification on dirty word texts.

The rest of this paper is structured as follows. Section 2 introduces the related work. Section 3 present our approach to constructing a corpus on dirty word texts. In Section 4, we present experimental results of automatic identification of Chinese dirty words. We conclude in Section 5 with some discussion of our method and the directions for future work.

ISSN: 1738-9968 IJHIT

2. Related Work

Nowadays, many on-line games, forums have provided some functions for detecting dirty words. Their methods are mainly based on a dirty word dictionary, and the user's input words will be matched with the words in the dictionary. First, a word segmentation system is employed to split the user's input text. Then, the segmentation results will be matched with the words in a dirty word dictionary. If a dirty word is found, the entire user's input text would be abandoned, or the dirty words would be replaced with asterisks. Due to the limited coverage of the dirty word dictionary and the complexity of expressing dirty word in languages, the dictionary matching approach often suffers from both low precision and low recall. Thus it sometimes confuses the normal users' information.

Let's give an example of mistaken identification as follows. Figure.1 shows the results of handling dirty words in a well-known gaming platform. The last sentence in Figure.1 is "你不要觉得我傻，逼着我下棋". The character "逼" is considered as a dirty word in the dictionary, so it is amended by replacing two asterisks. However, it isn't a dirty word in this sentence due to the word disambiguation.

Automatic identification of dirty word texts can be considered as a specific application of sentiment analysis. As a pioneering work, Pang, *et al.*, [1] apply machine learning methods to perform sentiment classification. They use different features and three machine learning methods to perform sentiment classification: NB (Naive Bayesian), ME (Maximum Entropy), and SVM (Support Vector Machine). They find that using unigram features yields the best performance in sentiment classification. Furthermore, they find that using the Boolean weights performs better than using TF-IDF (term frequency - inverse document frequency) weights. Cui's [2] experiments show that when the training corpus is small, unigram is the dominant, but when the training corpus increases, bigram, trigram and n-gram (n > 3) features play an increasingly important role. Ng's [3] experiments find that using bigram and trigram features together with unigram entries can improve the classification performance of SVM.

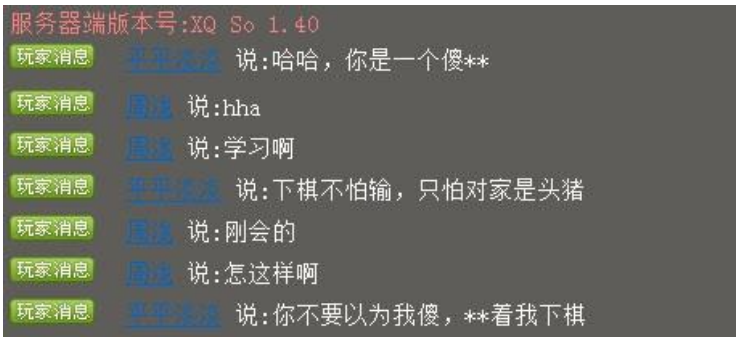

Figure 1. A Typical Example of Mistaken Identification

Generally, corpus plays an important role in sentiment analysis. Swapna Somasundaran [4] introduces the detail, precautions and specification of manual annotation on emotion corpus, and they explain how to train the annotators of work. They also describe the evaluation methods to measure the degree of unity between the annotators. Wiebe [5] describe the MPAQ (multiple-perspective QA) corpus' manual annotating process in detail.

Manually annotating corpus is very expensive and time-consuming. To reduce the annotation cost, we exploit active learning approach to help constructing the corpus on dirty word texts. Here, active learning is an effective technology for data mining from lots of unlabelled data. This method is reported to be rather effective in text classification [6].

3. Constructing the Corpus on Dirty Word Texts

The corpus on dirty word texts corpus is beneficial to the research on sentiment analysis. Manually annotating the corpus is a time-consuming working, but it is easy to get a large number of review texts through Web. It is possible to build variety corpus with machine learning. Here, a high-quality and large-scale corpus on dirty word texts is constructed through active machine learning.

3.1. Dirty Word Texts

Generally, Chinese dirty words can be summarized as the following five categories as reported by Liao [7]: (1) The text is about sex organs and sexual behavior of male or female. (2) The text is about the speaker considers himself as the other side's eldership. (3) The text is about the speaker abuses the others as animal. (4) The text is associated with the others with foul things. (5) The text is about blasphemy and curses. Among them, (1-5) are considered as apparent dirty words and (2) is considered as implicit dirty words. This paper mainly focuses on apparent dirty words.

3.2. Collecting Review Texts

We collect our data from Baidu Tieba [8] which is a huge Chinese Web community and an online communication platform for those peoples who are interested in the same topic. There are plenty of actual evaluating texts which include a large amount of dirty word texts.

The data structure of Baidu Tieba's post is analyzed, and then a crawler is designed and implemented to automatically get evaluating texts from Baidu Tieba. A number of HTML pages are automatically collected by the crawler. The regular expression technology is used to extract evaluating texts from the HTML pages. Some html tags in evaluating texts are handled. For example, the tag " " is replaced by a space character. The collected texts are split into sentences with period, question mark, exclamation mark or a line break. If a sentence has more than 1000 words or hasn't any Chinese characters, it will be abandoned. Finally, we obtain 1,736,722 review sentences which constitute a review text corpus.

These sentences are segmented by the Stanford Word Segmenter [9]. We further check the distribution of words frequencies in the corpus after word segmentation. A plot of word frequencies is presented in Figure 2. From this figure, we can see that the distribution of word frequencies follows Zipf's law [10], which confirms a proper characteristic of the collected corpus.

3.3. Solution

Natural language words are often ambiguous and the combinations between words are full of variant. Therefore, it is difficult to achieve a high quality of dirty words recognition only based on keywords. If a high quality dirty word texts corpus is constructed on the basis of mass texts, the features of dirty word texts will be extracted and mined easily. A novel method is designed for collecting dirty word texts from the mass texts depending on active machine learning. Figure 3 gives the architectural overview of our method.

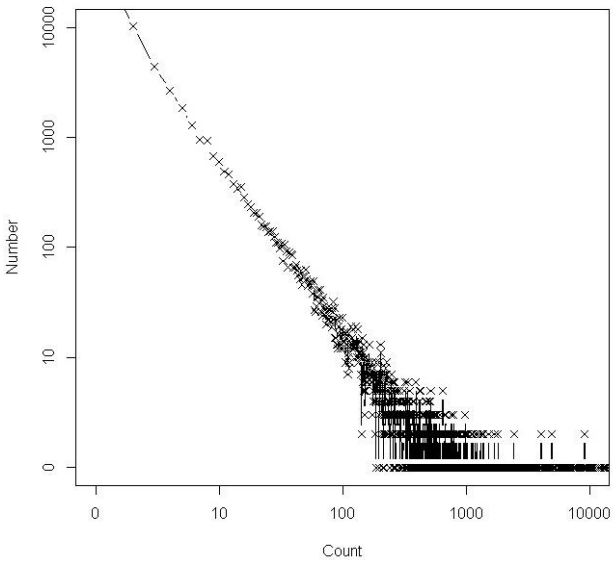

Figure 2. The Distribution of the Word Frequencies

As showed in Figure 3, we manually collect a small amount of typical features which constitute a dirty word features library. Then, a high-precision classifier is used to scan sentences in mass texts. Each sentence is assigned a weight value by the classifier. If a sentence's value was larger than a threshold value, it would be considered as a dirty word sentence and be added into the dirty word sentences collection. If a sentence's value equal 0, it would not be considered as a dirty word sentence, which would be added to the no dirty word sentences collection. All the other sentences would be considered as unsure dirty word sentences, and would be added into the unsure collection. The dirty word sentence collection and the no dirty word sentence collection are processed separately in order to find high-frequency and typical features. An algorithm is designed to identify the most possibly features of dirty words, and the features would be checked manually. If a feature is considered as a dirty word, it would be added to the dirty word feature library. Then, these steps will be looped many times until the features which need to be manually checked become very few. Apparently, at the beginning, there are only a few of dirty word seeds, thus the sentence collection of no dirty word has a lot of dirty word sentences in fact. As the loop is executed, the collection of dirty word sentences would keep growing, and the collection of no dirty word sentences becomes smaller and smaller.

3.4. A High-Precision Classifier

We find that POS features and syntactic features of dirty word sentences have no apparent difference with ordinary sentences through statistics and observations. Unigram discards the orderly relationship between words, and the complexity of n-gram (n>3) is too high. Thus we analyze dirty word text through unigram, bigram and trigram of phrase.

According to the rules mentioned above, 400 dirty words sentences are selected manually from Baidu Tieba. These sentences are segmented by the Stanford Word Segmenter. Then, the word segment results are analyzed using the SRILM (The SRI Language Modeling Toolkit) tool [11]. Finally, 40 phrases are discovered.

Five annotators are invited to annotate each weight value of the 40 phrases. These weight values are among 1 to 10. Higher values mean the greater likelihood of the dirty words. The weight values of each word annotated by five annotators are calculated averagely. These phrases are regarded as the seeds of dirty words features. Table [1] presents all the seeds and their weight values.

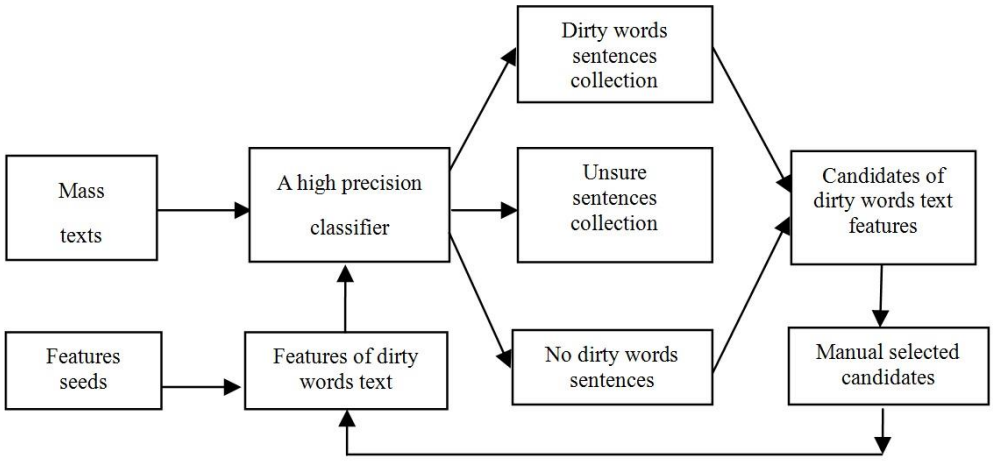

Figure 3. The Architectural Overview of our Method

Table 1. The Dirty Word Seeds and their Weight Values

Word	Value	Word	Value	Word	Value	Word	Value
SB	10	婊子养的	10	狗臭屁	8	尼玛	8
JB	8	狗比样	10	他妈	5	赶	2
我艹	10	操	5	鸟人	4	2	2
SB.....	10	阴道	10	死去	5	王八蛋	5
贱	5	发情	8	TM	4	二	2
逼	2	草	3	滚	3	屄	10
狗	2	母狗	8	杂种	10	CNM	10
B	2	泥马	10	全家	2	妈的	6
你个	1	妓女	10	日	2	死	5
婊子	10	鸡巴	10	傻逼	10	肏	10

Formally, a sentence is represented as $W = \{w_1, w_2, ..., w_n\}$, where w_i is a word segment phrase. Features set is represented as $F = \{(fea_1,value_1),(fea_2,value_2)...(fea_m,value_m)\}$, where $(fea_j,value_j)$ denotes the relationship between fea_j and $value_j$. $Fvalue(fea_j)$ is a function, which is used to get a specify feature's weight value. An algorithm is designed to detect whether a sentence is a piece of dirty word text or not. The algorithm is illustrated as follows.

Input: Sentence S_k
Input: Features
Input: Threshold
Output: Sentence Foul Score and Foul Detection
1: $Foul(S_k)=0$;
2: for i=1 to m
3: if $IsExist(S_k,fea_i)$=TRUE then
4: $Foul(S_k)=Foul(S_k)+ fvalue(fea_i)$
5: end if
6: end for
7: if $Foul(S_k)>$ Threshold then
8: S_k is classified foul.
9: else if $Foul(S_k)=0$ then
10: S_K is classified not foul.
11: else
12： S_k is classified uncertainty
13: end if

When the dirty word feature library only includes 40 features as listed in Table 1, and the threshold value is assigned to 20. Empirical study shows that the high-precision classifier's precision and recall are 96.29% and 20.47% respectively.

3.5. Preprocessing

Since the character "2(two)" and "日(day)" are often appeared in date string, they are listed in Table 1. They might be mistakenly considered as dirty words. Therefore, all the date string was caught by regular expressions and replaced with empty.

3.6. Features Recommendation

Using the high-precision classifier, we find 284 dirty word sentences from 1,736,722 sentences. These sentences are analyzed on the unigram, bigram and trigram feathers therein. The features are sorted according to their frequencies and they are considered as candidate features. The top 5% of unigram features, the top 2% of bigram features and the top 1% of trigram features are selected. The CHI-square test was applied to calculate the selected features. The CHI-square test is one of the most commonly used statistical methods [12].

$$\chi^2(features_i, C_j) = \frac{N \times (A \times D - C \times B)^2}{(A+C) \times (B+D) + (A+B) \times (C+D)} \quad (1)$$

Where, N represents the total count of the training set, including dirty words sentences and no dirty words sentences. C_j represents a specific category, $C_j \in$ {dirty words, no dirty words}. $Feature_i$ is a feature. The character A is the number of sentences which belong to the class C_j and contain $Feature_i$. The character B is the number of sentences which do not belong to the class C_j and contain $Feature_i$. The character C is the number of sentences which belong to the class C_j but not contain $Feature_i$, and the character D is the number of sentences which do not belong to the class C_j and not contain $Feature_i$.

3.7. Corpus Constructing

Constructing the dirty word text corpus is a process of multiple iterations. In the first round, the dirty word features library has 40 feature seeds. The high-precision classifier found 284 dirty words sentences from 1,736,722 sentences. Then, the frequency of unigram, bigram and trigram features are counted out by SRILM. The values of CHI-square test are ordered by descending, and the top 50 features are recommended as candidate features of dirty words. These features are manually filtered. The filtered features are manually tagged with the weight values and then added to dirty word feature library. This is a typical active learning period. The loop started up again and the process is repeated until the number of dirty words sentences was more than 6,000.

The whole process is repeated 6 rounds. Consequently, 162 features and 6232 sentences are obtained from 1,736,722 sentences. Then, a high quality dirty word text corpus of sentence is generated. The number of sentences and the filtered features in each round are illustrated in Table [2].

Table 2. Features Obtained in Each Round

Round	Features before round	Filtered sentences	Filtered features
1	40	284	21
2	61	2120	36
3	97	3434	38
4	135	4675	15
5	150	5846	13
6	162	6232	-----

4. Automatic Detecting Chinese Dirty Word Texts

Once the dirty words text corpus is ready, we can use the automatic text classification method to realize the dirty words recognition. Baidu Baike [13] is a website which allows people involved in editing items. There are lots of introduction of famous peoples in the Baidu Baike site, which usually do not contain dirty words. We randomly selected famous persons' introductory texts on Baidu Baike site. These texts were considered as none dirty word text set. Then we can use text classifier for experiments. NB, KNN (K-Nearest Neighbor), ME and SVM classifiers are applied in our experiment.

4.1. SVM and ME Classifiers

SVM [14] seeks a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set. SVM want to find the hyperplane which not only separates the training data but also has a maximal margin. A simple linear support vector machine example is showed in Figure 4. Previous studies report that SVM is very effective for the task of automatic text classification [15].

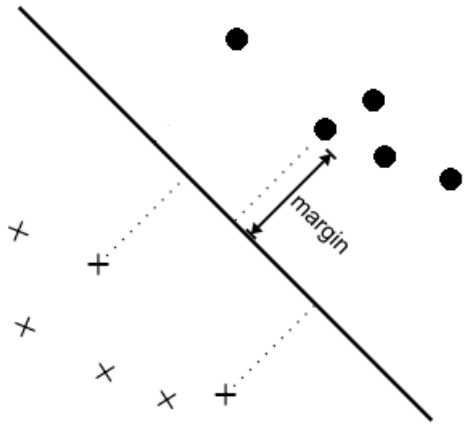

Figure 4. A Simple Linear Support Vector Machine Example

Maximum entropy (ME) model is a probability estimation technique widely used for a variety of natural language tasks. ME makes no assumptions about the relationships between features, and so might potentially perform better when conditional independence assumptions [1]. The objective of this mode is to estimate of the probability, *i.e.*, P(c|d), with the following formula:

$$P_{ME}(c \mid d) := \frac{1}{Z(d)} \exp(\sum_{i} \lambda_{i,c} F_{i,c}(d, c')) \qquad (2)$$

Where Z(d) is a normalization function. $F_{i,c}$ is a feature/class function for feature f_i and class c. It is defined as follow:

$$F_{i,c}(d, c') := \begin{cases} 1 & n_i(d) > 0 \quad and \quad c' = c \\ 0 & otherwise \end{cases} \qquad (3)$$

4.2. Feature Selection

A good feature selection is not only beneficial to reduce the computational complexity, but also to improve the performance. There are several familiar feature selection methods,

such as DF (document frequency), TF-IDF, IG (information gain), MI (mutual information) and CHI [16].

DF is simply count the number of documents containing the feature. TF–IDF is a metric that multiplies the two quantities TF and IDF. Here, TF provides a direct estimation of the occurrence probability of a term when it is normalized by the total frequency. On the other hand, IDF can be interpreted as 'the amount of information' in conventional information theory [17]. IG is used to calculate the number of bits of information obtained for category prediction given a feature. CHI can measure the lack of independence between a term and the category.

In this study, we employ CHI as our feature selection method due to its good performance in previous studies. Similar to the corpus constructing, we use word unigrams, bigrams and trigrams as the features in the experiments of performing automatic detection on dirty word texts.

4.3. Experiments

SVM and ME are used to detect Chinese dirty word texts. The SVM classifier is implemented with the tool LIBSVM [18], which is currently one of the most widely used SVM software. It is designed and implemented by Taiwan University's Lin. The ME classifier is implemented with the tool by Zhang [10] in Northeastern University China. 5-folds cross test is performed and the results are averaged.

The 6232 sentences are used as the data set of dirty word texts. We collected 6656 sentences from the Baidu Baike personal introduction, which was regarded as the data set of none dirty word texts. First, two text sets are segmented by the Stanford Word Segmenter. Then, stop words are removed from word segmentation results. Then, unigram, bigram and trigram features are counted out, and these results were sorted by frequency. Afterwards, the top 5% of unigram features, the top 2% of bigram features and the top 1% of trigram features were selected. CHI-square test is applied to calculate the selected features again. The experiments are conducted with the top 200, 400, 600, 800, 1000, 1500 and 2000 features. Table [3] shows the results of the classification experiments.

From Table [3], we can see that:

- Both SVM and ME are effective for automatic detecting dirty words text.
- The precision of classifying slightly increase when the number of features increases.
- The recall of classifying slightly decreased when the number of features increases.
- F-Measure is stabilized at over 98% in each experiment.

Table 3. The Results of Detecting Dirty Word Texts

Features number	SVM			ME		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
200	99.35%	97.75%	98.54%	97.64%	99.60%	98.61%
400	99.43%	97.51%	98.46%	97.25%	99.28%	98.25%
600	99.59%	97.35%	98.46%	96.12%	99.44%	97.75%
800	99.59%	97.11%	98.33%	96.88%	99.60%	98.22%
1000	99.83%	96.79%	98.29%	97.09%	99.20%	98.13%
1500	99.92%	96.55%	98.20%	97.40%	99.12%	98.25%
2000	100.00%	96.39%	98.16%	98.01%	99.04%	98.52%

These experimental results show that machine learning method is a practical approach for automatic identification of dirty word texts. In practice, we need to adjust the balance of precision and recall.

5 Conclusion

In this paper, we focus on analyzing dirty words in Chinese texts. Specifically, we propose an active learning approach to constructing a Chinese text corpus containing many dirty word sentences from a mass review collection. On this basis, we use the corpus to train machine learning-based classifier to automatically detect dirty word texts. Empirical studies show that our approach achieves outstanding results for this task.

In our future work, we will apply our classifiers in many online forum, blog or micro blogging platforms to filter dirty word texts. Meanwhile, we would like to employ our corpus and detection methods in sentiment classification to help detect negative comments. Moreover, we would like to apply our approach of constructing corpus to some other task in the community of natural language processing.

References

[1] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up?: sentiment classication using machine learning techniques", Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Association for Computational Linguistics, vol. 10, **(2002)**, pp. 79-86.
[2] H. Cui, V. Mittal and M. Datar, "Comparative experiments on sentiment classification for online product reviews", AAAI, vol. 6, **(2006)**, pp. 1265-1270.
[3] V. Ng, S. Dasgupta and S. M. Arin, "Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews", Proceedings of the COLING/ACL on Main conference poster sessions, Association for Computational Linguistics, **(2006)**, pp. 611-618.
[4] S. Somasundaran, J. Wiebe, P. Hoffmann and D. Litman, "Manual annotation of opinion categories in meetings", Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora, Association for Computational Linguistics, **(2006)**, pp. 54-61.
[5] J. Wiebe, T. Wilson and C. Cardie, "Annotating expressions of opinions and emotions in language", Language resources and evaluation, vol. 39, **(2005)**, pp. 165-210.
[6] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification", The Journal of Machine Learning Research, vol. 2, **(2002)**, pp. 45-66.
[7] L. De-Ming, "Sexual Consciousness Orientation of Dirty Words", Journal of Eastern Liao ning University (Social Sciences), vol. 11, **(2009)**, pp. 25-30.
[8] http://tieba.baidu.com/.
[9] H. Tseng, P. Chang, G. Andrew, D. Jurafsky and C. Manning, "A conditional random field word segmenter for SIGHAN bakeoff 2005", Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, **(2009)**, pp. 171-172.
[10] L. Q. Ha, E. I. Sicilia-Garcia, J. Ming and F. J. Smith, "Extension of Zipf's law to words and phrases", Proceedings of the 19th international conference on Computational linguistics, Association for Computational Linguistics, vol. 1, **(2002)**.
[11] A. Stolcke, "SRILM-an extensible language modeling toolkit", INTERSPEECH, **(2002)**.
[12] N. Mantel, "Chi-square tests with one degree of freedom; extensions of the Mantel-Haenszel procedure", Journal of the American Statistical Association, vol. 58, no. 303, **(1963)**, pp. 690-700.
[13] http://baike.baidu.com/.
[14] V. Vapnik, "The nature of statistical learning theory", Springer Science & Business Media, **(2000)**.
[15] T. Joachims, "Transductive inference for text classification using support vector machines", ICML, vol. 99, **(1999)**, pp. 200-209.
[16] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization", ICML, vol. 97, **(1997)**, pp. 412-420.
[17] A. Aizawa, "An information-theoretic perspective of TF-IDF measures", Information Processing & Management, vol. 39, **(2003)**, no. 1, pp. 45-65.
[18] C. Chih-Chung and C.-J. Lin, "LIBSVM: a library for support vector machines", ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, **(2011)**, pp. 27-27.
[19] Z. Le, "Maximum entropy modeling toolkit for Python and C++", Natural Language Processing Lab, Northeastern University, China, **(2004)**.

Authors

Xiaoxu Zhu, He was born in 1975. He received the B.S. degree in computer application technology from Wuhan University of Technology of China, Hubei in 1997, and the M.S. degrees in computer application technology, Soochow University, Jiangsu in 2003. His current research interests include natural language processing and information system. He is currently a lecturer in Soochow University.

Peide Qian, He was born in 1947. He received the B.S degree from Nanjing University in China. Now, he is a professor in Soochow University. His research interests include operating system, Chinese character information processing, and distributed computation.