

A ME Model Based on Feature Template for Chinese Text Categorization

Li Pei-feng^{*+}

Zhu Qiao-ming^{*+}

Li Jun-hui^{*}

^{*}School of Computer Science & Technology
Soochow University
Suzhou, Jiangsu, China

⁺Key Lab of Computer Information Processing
Technology of Jiangsu Province
Suzhou, Jiangsu, China

Abstract - With entering into information society and the Internet developing rapidly, people could acquire more and more information. How to utilize Internet information efficiently and promptly, has become a hotspot in information technology. Text categorization is an important component to help getting useful message from tremendous amount of vast information. And it assigns new documents to pre-defined categories automatically based on their content. In this paper, we puts forward propose an improving approach to apply ME model to Chinese text categorization, and its main points include constituting feature functions and pre-disposing documents. In order to make the feature items more reasonable, each document is undergone four steps, namely removing html tags, word segmentation, filtering empty words and removing stop words. In addition, feature-template is brought forward, which is combined with feature item's weight while constituting feature function. The results of experiments show that the performance with ME model based upon the combination of feature-template and weight outperforms that with ME model based upon word-frequency, and both of their performance is better than that of Naïve Bayes and KNN.

Key Words: ME model, Feature Template, Chinese Text Categorization

1.0 Introduction

With entering into information society and the Internet developing rapidly, people could acquire more and more information. How to utilize Internet information efficiently and promptly, has become a hotspot in information technology. Text categorization is an important component to help getting useful message from tremendous amount of vast information. And it assigns new documents to pre-defined categories automatically based on their content.

A variety of approaches towards text categorization have been researched in the past few years. The most popular approaches to text

categorization include Naïve Bayes and related Bayes learning methods [12, 13, 14], clustering techniques such as the kNN(k-Nearest Neighbor) algorithm [14, 16], decision trees, SVM(support vector machines) [14, 15, 16], neural networks[14], and so on.

In this paper, we explore the use of maximum entropy model for text categorization, which offers comparable performance to KNN or SVM [5, 6]. The ME framework has proved to be a powerful modeling tool in many areas of NLP, including sentence boundary disambiguation, part-of-speech tagging, chunk parsing [1], keyword indexing [2] and machine translation. Adwait R. and Kamal Nigam[3, 4] used ME for English text categorization; Chen Xue-Tian and Li Rong-Lu[5, 6] used ME for Chinese text categorization, which using word-frequency as the value of feature function and their micro-average precision was 92.73%. In this paper, ME model is also applied to Chinese text categorization. We pre-dispose documents with four steps, and then combine feature-template and weight to generate feature functions. In experiment, we compared our method with that in [5, 6], and the results of experiments show that the performance with ME model based upon the combination of feature-template and weight outperforms that with ME model based upon word-frequency, and both of their performance is better than that of Naïve Bayes and KNN.

This paper proceeds as follows. Section 2 presents an overview of ME. Then the application of ME model to Chinese text categorization is described in Section3. Experiments and their results are showed in Section 4. Finally, Section 5 summarizes the paper and highlights some problems in future work.

2.0 An Overview of ME

The ME model is one of mature statistics models, and it is fit for text categorization. The basic idea of ME is that we should prefer the most uniform models that also satisfy any given constraints, which are mined from the known event collection.

In text categorization, each document is deemed as

an event. For examples, there is an event collection which presented as $\{(d_1, c_1), (d_2, c_2), (d_3, c_3), \dots, (d_N, c_N)\}$, where $d_i (1 \leq i \leq N)$ denotes a document and $c_i (1 \leq i \leq N)$ is the category of document d_i . We let any real-valued function of the document and the category be a feature function of $f_i(d, c)$. The expected value of $f_i(d, c)$ with respect to the empirical distribution $\tilde{p}(d, c)$ is exactly the statistic we are interested in and it can be denoted as:

$$\tilde{p}(f_i) \equiv \sum_{d,c} \tilde{p}(d, c) f_i(d, c) \quad (1)$$

The expected value of f_i with respects to the model $p(c|d)$ is

$$p(f_i) \equiv \sum_{d,c} \tilde{p}(d) p(c|d) f_i(d, c) \quad (2)$$

where $\tilde{p}(d)$ is the empirical distribution of d in the training samples. We constrain this expected value to be the same as the expected value of f_i in the training sample. The constraint is

$$p(f_i) = \tilde{p}(f_i) \quad (3)$$

Suppose that we are given n feature functions. We would like our model to accord with these constraints. That is, we would like p to lie in the subset C of P defined by

$$C \equiv \{p \in P \mid p(f_i) = \tilde{p}(f_i) \text{ for } i \in \{1, 2, \dots, n\}\} \quad (4)$$

Among the models $p \in C$, the maximum entropy philosophy dictates that we select the most uniform distribution. And the mathematical measure of the uniformity of a conditional distribution $p(c|d)$ is provided by the conditional entropy:

$$H(p) \equiv - \sum_{d,c} \tilde{p}(d) p(c|d) \log p(c|d) \quad (5)$$

The ME principle presents us with a problem in constrained optimization: find the $p_* \in C$ that maximizes $H(p)$. It can be shown that p_* is always well defined and is always of the exponential form [7]:

$$p_1(c|d) = \frac{1}{Z_1(d)} \exp\left(\sum_i I_i f_i(d, c)\right) \quad (6)$$

where

$$Z_1(d) = \sum_c \exp\left(\sum_i I_i f_i(d, c)\right) \quad (7)$$

is simply a normalizing factor determined by the requirement of that $\sum_c p_1(c|d) = 1$ for each document d , f_i is a feature function, I_i is the weight assigned to feature f_i . Two algorithms specifically tailored to calculate the parameters of a ME classifier are generalized iterative scaling

algorithm [7] and improved iterative scaling algorithm [8].

In this section, we briefly outline the GIS algorithm. Algorithm 1 (Generalized Iterative Scaling (GIS)). The following procedure will converge to p_* :

$$I_i^0 = 0 \quad (8)$$

$$I_i^{(n+1)} = I_i^n + \frac{1}{c} * \log\left(\frac{E_{\tilde{p}} f_i}{E_{p^{(n)}} f_i}\right) \quad (9)$$

where

$$E_{p^{(n)}} f_i = \sum_{d,c} \tilde{p}(d) p^{(n)}(c|d) f_i(d, c) \quad (10)$$

$$p^{(n)}(c|d) = \frac{1}{Z_{(n)}(d)} \exp\left(\sum_i I_i^{(n)} f_i(d, c)\right) \quad (11)$$

Darroch J. N.[7] considered that the algorithm is convergent.

3.0 Using ME for Chinese Text Categorization

3.1 The Process of Chinese Text Categorization

There are two components in text categorization, namely classifier training and testing. While using ME model for text categorization, classifier training includes pre-disposing documents in training data and feature denotation, and classifier testing contains pre-disposing documents in test data and categorizing them.

Figure 1 shows the process of applying ME model for categorization. Firstly, we pre-dispose training documents in four steps, namely removing html tags, word segmentation, filtering empty words and removing stopwords, and then the feature items for each document are generated. Secondly, each feature items in a document is transformed to feature function according to the way of generating feature function. Lastly, use GIS or IIS algorithm to calculate parameters. While testing documents, the two initial steps are as same as above, and the last step is the classifier outputting results.

3.2 Pre-disposing Document

While using ME for text categorization, each document is treated as an event. Unlike English and some other languages, Chinese text does not have a natural delimiter between words. So word segmentation with which POS tagging is accompanying is an essential step in our processing. After word segmentation and POS tagging for a document, each word would have corresponding features, so the number of features would be very

large. According to the idea that different part-of-speech tags pay various roles in text categorization, we filter any word whose part-of-speech tag is conjunction, or exclamation, or preposition, or interpunction and so on. The left words' part-of-speech tags are almost noun, verb etc. Then we remove stopwords that carry no or little information from the left words. The experiment shows that filtering part-of-speech tags and removing stopwords speed up text categorization while using ME, but also improve the recall and precision.

3.3 Feature in ME Model

Li Su-Jian et al. [2] argued that the key point while using ME model is that how to constitute feature collection in a specific task. Use simple feature to denote the complicated language phenomenon, and acknowledge the facts that we could deduce from

training data, and make no independent hypothesis. Those deduced facts are denoted as feature collection in ME model.

Because we have to assign category label for each document, and the label process is treated as an event, we use current document to generate the event's feature collection. Chen Xue-Tian and Li Rong-Lu[5, 6] choose "word--categorization" as a feature, and word frequency as the value of the feature. So for each word-category ($W - c'$) pair, its feature is denoted as:

$$f_{w,c'}(d,c) = \begin{cases} num(d,w) & c = c' \\ 0 & otherwise \end{cases} \quad (12)$$

where $num(d,w)$ is the number of times word w occurs in document d .

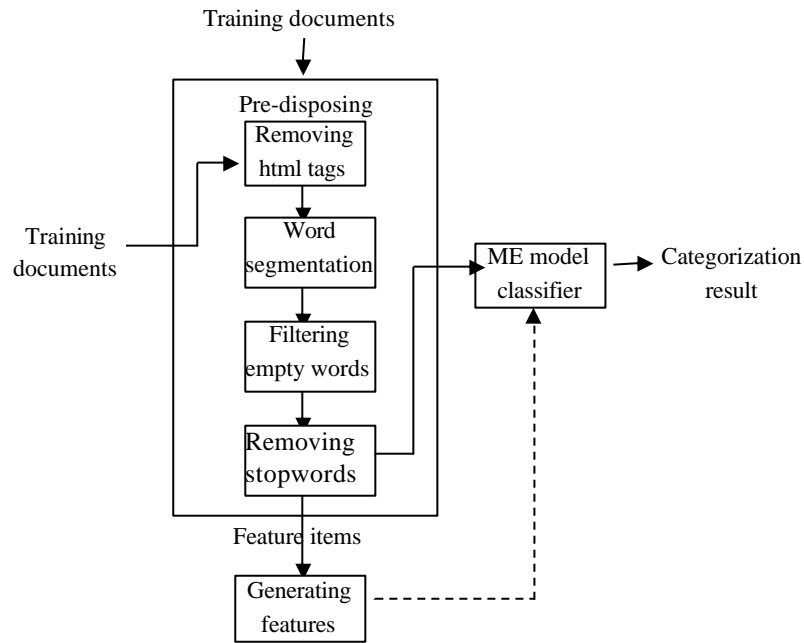


Figure 1 The process of Chinese text categorization based upon ME model

Table 1 The feature-template

Template-num	Memo	Template-name	Scope of value
1	Number of word occurs	F	{A, B, C, D} A: 1—3, B: 4—6, C: 7—9, D: >10
2	Position where word first appears	P	{T, B} T: document title, B: document body
3	Labeled result	L	{The collection of categories}

In this paper, we propose another way to generate features, which is the combination of feature-template and weight. While establishing the feature-template, we consider two aspects for each

word w in document d : the times word w occurs in document d and the position where word w first appears in document d . We define the

scope for each template item's value. Table 1 shows the feature-template.

In Table 1, the template item 3 is special, which denote the labeled result. The template will be instantiated as long as the value of template function is defined. While template item 1 or 2 has a certain value and the labeled category is known, a feature is created. For example, word w occurs 5 times in document d whose category is Sports, and word w first appears in the document's title, so it generates two features that are only binary valued:

$$f_j(w, c) = \begin{cases} 1 & F(w) = B \text{ \& } c = 'Sports' \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$f_k(w, c) = \begin{cases} 1 & P(w) = T \text{ \& } c = 'Sports' \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

In following experiments, we find that the categorization performance of only using feature-template whose feature is binary valued is inferior to that of using word-frequency as feature's value. Therefore, we optimize the feature by merging formula (13) and (14) to yield a new feature, in which word-frequency weight and word-position weight are given respectively according to their practical value. We call the new method as ME model based upon the combination of feature-template and weight. The new feature is as:

$$f_j(w, c) = \begin{cases} freq(w) * posi(w) & c = 'Sports' \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where $freq(w)$ is called word-frequency weight and its value is related to $F(w)$, $posi(w)$ is called word-position weight and its value is related to $P(w)$. Another advantage by merging features is that the total number of features is decreased greatly, so speed up the categorization process.

3.4 Smoothing Processing

The term smoothing refers to the adjustment of the maximum likelihood estimator of a language model so that it will be more accurate. At the very least, it is required to not assign a zero probability to unseen words.

To the best of my knowledge, there is no smoothing technology for ad hoc ME model. Chen Xue-Tian and Li Rong-Lu[5, 6] apply absolute-discounting smoothing technology to ME model: the feature's value is ad when the word-frequency is zero. By doing this, feature (12) is converted to the follow form:

$$f_{w,c}(d, c) = \begin{cases} num(d, w) & c = c' \\ ad & \text{otherwise} \end{cases} \quad (16)$$

We apply this smoothing technology for the other two ME model analogously to view the

categorization result. In this paper, the value of ad is 0.1.

4.0 Experiment Results

4.1 Corpora

To explore the performance of ME models, we need Chinese documents whose title are well defined. We collected two corpora for the experiments.

First, we collect 8557 category-defined and title-defined news documents from Sina.com web site etc. The eight categories are {Economics(财经), Environment(环境), Education(教育), Military(军事), Sports(体育), Medicare(医保), Entertainment(娱乐), Politics(政治)}. The number of each category document is a little more than 1000. We choose 300 documents randomly for each category as test documents, and the rest as training documents. The proportion between training documents and test documents is about 2.6:1.

The other public corpus which we used is the FuDan Chinese document corpus that contains 2,186 documents originally. It is an open corpus and used widely for Chinese text categorization. For the reason that we need documents whose titles are well defined, we tidy up these documents in order to specify each sample's title and body, and filter documents that either are derived from BBS or have no title at all. Finally 2,260 documents are left and one-third documents for each category are chosen randomly as test documents, and the rest are as training documents. The categories in the corpus are { Environment(环境, 102, 1), Computer(计算机, 41, 2), Transportation(交通, 151, 3), Education(教育, 204, 4), Economics(经济, 225, 5), Military(军事, 242, 6), Sports(体育, 447, 7), Medicine(医药, 104, 8), Art(艺术, 239, 9), Politics(政治, 505, 10)}. Each category's Chinese name, document number, and serial number are shown in the parentheses respectively.

4.2 Software Package

Instead of implementing word segmentation and POS tagging algorithms, we used Hai-Liang Intelligent Word-segmentation System [9] developed by Tian-Jing Han-Liang Science and Technology Development Ltd.

The ME model toolkit prototype [10] we used in our experiment was implemented by Jason Baldrige et al. Some of codes in it are modified for our experimental studies.

While calculating the parameters of a ME model the toolkit uses GIS algorithm.

4.3 Evaluation Measures

In our experiments we report recall (r) and precision (p) for each category, and micro-average p (or simply micro- p) for each text categorization approach. These measures are defined as follows:

$$r(c) = \frac{\text{the number of documents correctly assigned to class } c}{\text{the number of documents contained in class } c}$$

$$p(c) = \frac{\text{the number of documents correctly assigned to class } c}{\text{the number of documents assigned to class } c}$$

$$\text{micro-}p = \frac{\text{the number of correctly assigned test document}}{\text{the number of test documents}}$$

4.4 Results and Discussion

After several tests, $\text{freq}(w)$ and $\text{posi}(w)$ are constituted eventually as Table 2 and Table 3.

Table 2 word-frequency weight table

$F(w)$	A	B	C	D
$\text{freq}(w)$	1	2	3	4

Table3 word-position weight table

$P(w)$	T	B
$\text{posi}(w)$	3	1

The experiment results with the first corpus are showed in Table 4 and Table5, where AD denotes that the ME model uses absolute-discounting smoothing technology, NS means that no smoothing technology is used. While using ME model, the number of iteration in GIS algorithm is 100 times. In KNN model, the value of k is 10.

With the second corpus, we conduct experiments

by using ME models, Naïve Bayes, and KNN. The recall and precision for each category are shown in Figure 2, Figure3 and Table 6.

From these experiment results, we could see:

- (1) The performance by using ME model based upon the combination of feature-template and weight is better than that upon word frequency. The main reason may be that the former takes account of the position of word in a document, and give the weight according its position. In addition, the frequency of word is partitioned into four sections. So as two word-frequencies are in the same section though their values are distinguished, their effect is same.
- (2) The performance by using ME model based only upon feature-template is inferior to both of them discussed above. And it is reasonable for the feature in it is binary valued. Therefore each two different sections of word-frequency are independent. In the approach of using word-frequency, the feature value with bigger word-frequency would be bigger than the one with smaller word-frequency. And in the approach of combination feature-template and weight, word-frequencies in different section is discriminating by define word-frequency weight.
- (3) The effect by using absolute-discounting smoothing is not obvious. And in several experiments, the results with absolute-discounting smoothing are unpredictable: maybe it performs well, but sometimes performs badly. It illuminates that it is unreliable to using absolute-discriminating smoothing as employed in this paper.

Table 4 Experiment results using ME model in the first corpus

Category	Feature-template + weight				Only feature-template				Word-frequency			
	AD		NS		AD		NS		AD		NS	
	r	p	r	p	r	p	r	p	r	p	r	p
Economics	97.33	99.32	98.33	98.99	95	95.96	97	96.68	97.33	95.74	98.33	97.04
Environment	97	95.72	97.33	96.69	93.67	96.23	93.33	94.92	94.67	95.62	96.33	96.66
Education	97.33	96.05	98.33	95.16	95.33	95.65	96.67	93.25	97.67	93.91	99	93.10
Military	92.67	92.36	91.67	93.86	89	92.07	87.67	91.96	93.33	91.50	91.67	91.67
Sports	98.67	98.67	99.33	99.33	98	99.66	98.67	99.33	99	99.66	99	99.33
Medicare	99	98.34	98.33	98.99	98.33	97.04	97	97.32	98.33	98.33	98.33	98.33
Entertainment	95.33	96.30	97.33	97.01	97.67	93.61	97.67	95.75	96.67	97.97	96.33	98.97
Politics	92.33	92.95	92.66	93.29	94.33	91.29	91.33	90.13	88.67	93.01	89.67	93.73

Table 5 Micro-average precision by using different approaches in the first corpus

	Feature-template + weight		Only feature-template		Word-frequency		Only feature-template (no filtering words)	Naïve Bayes
	AD	NS	AD	NS	AD	NS		
micro-p	96.21	96.67	95.17	94.92	95.71	96.08	94.63	93.83

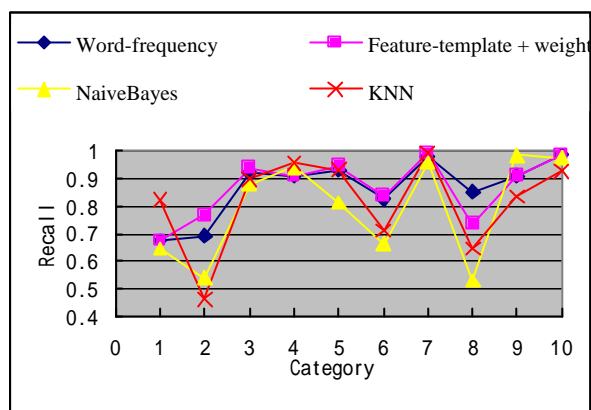


Figure 2 Recall vs. Category in FuDan corpus. The reason for recall values of category 2 and 8 is low may be that their training documents are fewer than others.

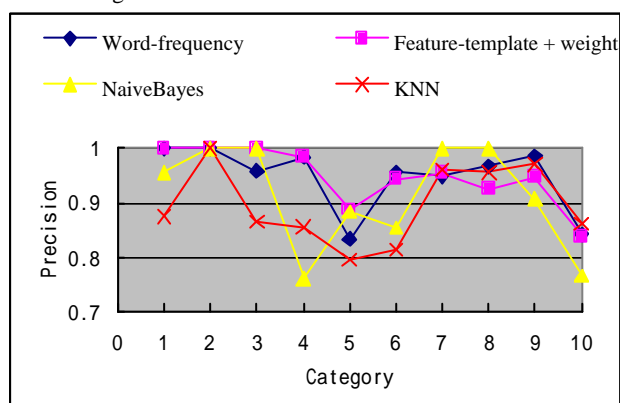


Figure 3 Precision vs. Category in FuDan corpus

Table 6 Micro-average precision by using different approaches in FuDan corpus

	Feature-template + weight	Word frequency	Naïve Bayes	KNN
micro-p	92.86	92.13	87.20	88.40

Furthermore, how to apply smoothing technology to ME model is under study.

- (4) Filtering part-of-speech tags and removing stopwords may improve the performance to a certain extent.
- (5) ME model outperforms Naïve Bayes model in text categorization, and it is also better than KNN in our experiments.

5.0 Summary and Future Work

We have introduced the method applying ME model to Chinese text categorization. While constituting feature, feature-template is proposed, and combined with weight. We have evaluated three different ME model for Chinese text categorization. Based on the empirical experiments conducted on the 8557 category-defined news documents and FuDan corpus, our main conclusions are:

Overall performance of only feature-template is worse than that of word-frequency. But combining the feature-template with weight, its performance is improved a lot and is better than that of word-frequency.

In the future, in addition to conducting more extensive experiments, especially with public text corpora and large-scale corpora, we plan to optimize the feature function and consider more aspects while designing the feature-template.

6.0 References

- [1] Li Su-Jian, Liu Qun, and Yang Zhi-Feng. Chunk Parsing with Maximum Entropy Principle[J]. *Chinese Journal of Computers*, 2003, 26(12):1722-1727.
- [2] Li Su-Jian, Wang Hou-Feng, Yu Shi-Wen, and Xin Cheng-Sheng. Research on maximum entropy model for keyword indexing[J]. *Chinese journal of computers*, 2004, 27(9):1192-1197.
- [3] Adwait R. Maximum entropy models for natural language ambiguity resolution [D]. PhD thesis. University of Pennsylvania, 1998.
- [4] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification[C]. In: *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999.
- [5] Chen Xue-Tian, and Li Rong-Lu. Using maximum entropy model for text categorization[J]. *Computer Engineering and Applications*, 2004, 40(35):78-79.
- [6] Li Rong-Lu, Wang Jian-Hui, Chen Xiao-Yun, Tao Xian-Peng, and Hu Yun-Fa. Using maximum

- entropy model for Chinese text categorization[J]. Journal of Computer Research and Development, 2005, 42(1):94-101.
- [7] Darroch J. N., and Ratcliff D. Generalized iterative scaling for log-linear models. Annals of Mathematical Statistics, 1972, 43(5): 1470-1480.
- [8] Pietra S D, Pietra V D, and Lafferty J. Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(4): 180-393.
- [9] Hai-Liang Intelligent Word-segmentation System. Available online at <http://www.hylanda.com/production/SegmentStudy.htm>
- [10] The OpenNLP Maximum Entropy Package. Available online at https://sourceforge.net/project/showfiles.php?group_id=5961
- [11] Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. A maximum entropy approach to natural language processing[J]. Computational Linguistics, 1996, 22(1): 38-73.
- [12] Yu Fang. A web document classifier based on Naive Bayes Method: WebCAT[J]. Computer Engineering and Applications, 2004, 40(13):195-197.
- [13] Andrew McCallum, Kamal Nigam. A comparison of event models for Naive Bayes text Classification. In AAAI/ICML-98 Workshop on Text Categorization. 1998:41-48.
- [14] Yang Y. AND LIU, X. A re-examination of text categorization methods. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), 1999:42-49.
- [15] James Tin-Yau Kwok. Automated Text Categorization Using Support Vector Machine. In Proceedings of the International Conference on Neural Information Processing(ICONIP), Kitakyushu, Japan, 1999:347-351.
- [16] Ji He, Ah-Hwee Tan, Chew-Lim Tan. A comparative study on Chinese text categorization methods[J]. PRICAI Workshop on Text and WebMining. 2000, 24-35.