Categorizing Web Information on Subject with Statistical Language Modeling

Xindong Zhou[1], Ting Wang[2], Huiping Zhou, and Huowang Chen

National Laboratory for Parallel and Distributed Processing,
Changsha, Hunan, P.R.China 410073
[1] zhouxindong@sohu.com, [2] wonderwang70@hotmail.com

Abstract. With the rapid growth of the available information on the Internet, it is more difficult for us to find the relevant information quickly on the Web. Text classification, one of the most useful web information processing tools, has been paid more and more attention recently. Instead of using traditional classification models, we apply *n*-gram language models to classify Chinese Web text information on subject. We investigate several factors that have important effect on the performance of *n*-gram models, including various order *n*, different smoothing techniques, and different granularity of textual representation unit in Chinese. The experiment result indicates that bi-gram model based on word and tri-gram model based on character outperform others, achieving approximately 90% evaluated by *F1* score.

1 Introduction

With the volume of information available on the Internet continues to increase, the need for tools that can help users easy to find, filter and manage the information quickly on the Web is growing accordingly [1]. Many tasks related to free-text processing, such as document retrieval, categorization, routing and filtering systems or agents, are usually based on text classification. A number of classification models have been developed and worked well in practice, which include Rocchio relevant feedback algorithm [2], Naive Bayes classifier [2], Decision Tree classifier [3], Regression methods [3], Neural networks [3], and Example-based classifier [3].

Statistical language modeling [4] has been widely used in many fields, including speech recognition, OCR, and machine translation. Most of the previous works use statistical language modeling in the field of text categorization and mainly focus on exploiting collocation or co-occurrence with the forms of bi-gram or tri-gram so as to present documents textual more precisely, or avoiding the word segmentation issues that always arise in Chinese or other Asian language. Peng [5] proposed Chain Augmented Naive Bayes classifier (CAN), which allows a local Markov chain depended on the context in calculating the class conditional probability of documents belong to categories in Naive Bayes model, relaxing some of the independence assumptions.

This paper applies the statistical language modeling—*n*-gram models [6] as text classifiers directly, treating documents as random observation sequences while traditional classifying models look documents as bag of words, the category in which a document can be observed with the biggest probability is considered as the result of

X. Zhou et al. (Eds.): WISE 2004, LNCS 3306, pp. 403–408, 2004.

classification. The goal of learner is to construct a language model for each category on its training corpus, which is quite different to traditional classifying models that based on representing documents as points in a multi-dimension vector space.

We investigate several factors that have impact on the performance of the *n*-gram models, including various order *n*, alternative granularity unit in Chinese textual representation, and the effect of different smoothing methods. Especially for Good-Turing smoothing [4], we adapt it to being used independently in language models.

2 Statistical Language Modeling

Statistical language modeling [6] has been widely used in the research of natural language processing. From the view of statistical language modeling, any string (can be imagined as the combination of words) could be accepted, and the distinction among them is the different possibility of acceptance. There are a few types of statistical language modeling, including *n*-gram Language Models [6], Hidden Markov Model [7], Probabilistic Context Free Grammar [6], and Probabilistic Link Grammar [8]. Among them the most widely used model, by far, is *n*-gram models. In our experiment, we will apply it as text classifier.

2.1 *n*-Gram Language Models

The *n*-gram language model has been used successfully in many fields; it captures the local constraint of natural language successfully [9]. Assuming there is a document *d*: $w_1 w_2 \ldots w_n$, where w_i means the textual presentation unit in a specified language. How can we calculate its observed probability in a certain category *c*: $P_c(d)$? In *n*-gram models, according the chain rule, we can calculate it as follow:

$$P_c(d) = P_c(w_1 w_2 \cdots w_n) = p_c(w_1)\, p_c(w_2 \mid w_1) \cdots p_c(w_n \mid w_1, \cdots, w_{n-1}) \tag{1}$$

2.2 Smoothing Methods

Now the problem we should deal with is how to calculate the conditional probability: $p(w_i|w_1 \ldots w_{i-1})$. Usually we use the maximum likelihood estimate (MLE):

$$p(w_i \mid w_1 \cdots w_{i-1}) = \frac{p(w_1 \cdots w_i)}{p(w_1 \cdots w_{i-1})} \approx \frac{C(w_1 \cdots w_i)}{C(w_1 \cdots w_{i-1})} \tag{2}$$

where $C(s)$ means the frequency of *s* occurs in training corpus. As the number of parameters that need to be estimated is tremendous, an unavoidable and crucial problem is how to deal with the case that $C(s)=0$, namely the problem of sparse data [9]. Smoothing techniques, tending to make distribution more uniform by adjusting low probabilities such as zero probabilities upward and high probabilities downward, are needed. There are several types of smoothing techniques: additive [4], discounting [4][5], back-off [4][5], and interpolation [4].

Here we introduce a novel variation of Good-Turing (GT) smoothing method, and our purpose is making this smoothing method can be used independently and generating probabilities more detailed and reliable. We pre-define a threshold of frequency for observed events, for the grams with frequency higher than it, we use MLE, for others, we use GT smoothing to discount probabilities a little. Let n_i denotes the number of events, which occur exactly i times in training corpus. As the case $n_i = 0$ would not occurs for infrequent grams, thus it can be used independently, Let k be the threshold, for a specified gram $w_1 \ldots w_n$, if its frequency is r and $0<r<k$, then the discounting probability that gets from this specified gram is:

$$\tilde{p}_r = \frac{r - r^*}{N} = \frac{r \times n_r - (r+1)n_{r+1}}{N \times n_r} \tag{3}$$

Finally the mass of discounting probability is: $\sum_{0<r<k} n_r \times \tilde{p}_r$

How to determine the best threshold value k? The optimal value of k is chosen based on empirical observations and related to the size of training corpus closely.

3 Applying *n*-Gram Models as Text Classifiers

Text classification is a task that assigns documents to a certain category according to its contents. Since we look document as a random observed sequence, well then the appearance of words and the sequence of them can be seen as a type of language combining modes, which is very closely related to the document contents itself. The *n*-gram models are exactly the tools that attempt to reveal or capture the language combining modes for different categories. The formal definition can be described as follow: for a new document d, $d = w_1 w_2 \ldots w_n$, we can calculate the probability of d that would appears in category c as:

$$P_c(d) = P_c(w_1 \cdots w_n) \approx \prod_{i=1}^{n} p_c\left(w_i \mid w_{i-N+1}^{i-1}\right) \propto \sum_{i=1}^{n} \log p_c\left(w_i \mid w_{i-N+1}^{i-1}\right) \tag{4}$$

and the decision rule is $\arg\max_{c \in |C|} P_c(d)$.

4 Experiment Results

The corpus used in our experiments is obtained from Fudan University. Most of them are Chinese Web pages collected from Internet, consisting of news, papers, and articles. All experiment results that given below are evaluated by *F1* score.

The implementing of various smoothing techniques in our experiments is: in additive smoothing methods, the size of vocabulary is the number of distinct events occurred in training corpus, and the adjustment coefficient λ is 0.1. For absolute discounting, d is assigned to $n_1/(n_1+2*n_2)$. For adaptive Good-Turing smoothing, the threshold k is 5 for tri-gram models, and 10 for bi-gram models.

Table 1. Results on categorizing Web information (CB, additive and discounting smoothing)

n	Laplace	Lidstone	Absolute	Linear	Good-Turing	Witten-Bell
1	83.72%	83.85%	83.44%	83.29%	**83.89%**	83.61%
2	84.78%	86.6%	86.78%	86.73%	86.66%	**86.93%**
3	81.43%	86.79%	89.88%	89.37%	89.14%	**90.11%**
4	76.48%	79.07%	80.85%	81.43%	81.02%	**81.58%**

Table 2. Results on categorizing Web information (WB, additive and discounting smoothing)

n	Laplace	Lidstone	Absolute	Linear	Good-Turing	Witten-Bell
1	85.16%	85.21%	85.28%	85.49%	86.04%	**86.12%**
2	80.03%	85.09%	89.35%	89.09%	89.72%	**89.94%**
3	77.33%	82.17%	85.06%	85.18%	85.42%	**86.24%**

4.1 Experiments on Character-Based (CB) Models

One advantage of using CB *n*-gram models is avoiding the problem of word segmentation in Chinese. Another advantage is that there are fewer parameters need to be estimated compared with Word-Based language models, so the reliability of model parameters is higher than the WB language models with the same order *n*.

4.2 Experiments on Word-Based (WB) Models

In Chinese, words have been considered as the smallest unit that can carry meanings; so breaking words into characters lead to lose information inevitably. To make a language model capture more information, we investigate WB language models.

4.3 *n*-Gram Models Versus Traditional Classifying Models

In order to compare the performance between *n*-gram models and traditional classifying models, we also construct and test Rocchio classifier and Naive Bayes classifier on the same corpus.

Fig. 1 shows the *F1* scores achieved by WB bi-gram model (89.94%) and CB tri-gram model (90.11%) are higher than Rocchio classifier (86.38%) and Naive Bayes classifier (86.54%). For traditional models, feature selection is a crucial step, and there are several other factors such as weighting strategy, etc, can also affect the performance. On the other hand, all the factors mentioned above need not to be considered in *n*-gram models any more, so the stability of *n*-gram models is better than traditional models.

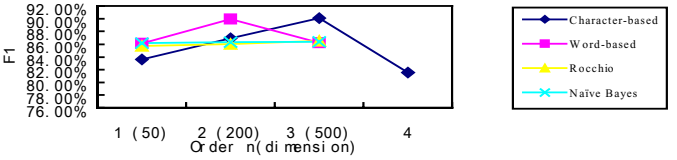

Fig. 1. Results of *n*-gram models (Witten-Bell smoothing), Rocchio classifier, and Naive Bayes classifier

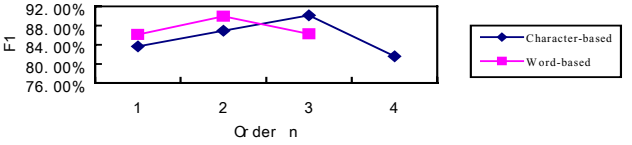

Fig. 2. Relations between *F1* score and order *n* (Witten-Bell smoothing)

5 Analysis of Experiment Results

There are several important factors that can significantly affect the quality of *n*-gram models in Chinese. From the experiment results, we will analyze the influence of each individual factor, and give the conclusions accordingly.

5.1 Effects of Order *n* and CB Models Versus WB Models

The order *n* is a key factors that relating to the quality of language models.

For the CB models, the sparse data problem is not as serious as the WB models', so the *F1* score increases up to the 3-gram, while WB models decreases at 3.

5.2 Effects of Smoothing Techniques

Smoothing technique is another important factor that can affect the performance of the language models, especially for the higher order models.

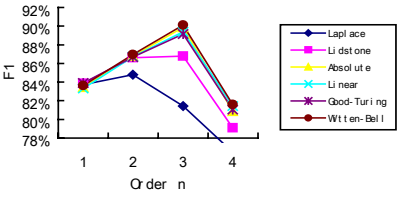

Fig. 3. Performance of smoothing techniques with CB models

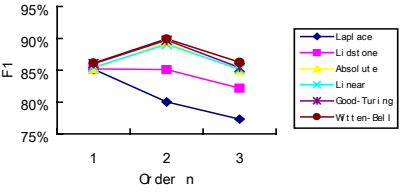

Fig. 4. Performance of smoothing techniques with WB models

We can find that Laplace smoothing is the most sensitive to the perplexity of model parameters. Lidstone smoothing has the moderate performance, and other smoothing methods' performance in parameters estimating have little difference, including our adaptive Good-Turing smoothing method.

6 Conclusions

In this paper, we apply *n*-gram language models as text classifiers directly. We also investigate several important factors that can affect the quality of statistical language models and compare the performance of *n*-gram models as classifiers with traditional classifiers on the task of Web information classification. The experiment result shows that the performance of *n*-gram language models in Chinese text classification is better than traditional classifying models.

Acknowledgement. This research is supported in part by the National High Technology Research and Development Program and the National Natural Science Foundation of China.

References

1. Aas, K. & Eikvil, L: Text Categorization: A Survey. Technical Report #941, Norwegian Computing Center, (1999)
2. Thorsten Joachim: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Processing of ICML-97□14th International Conference on Machine Learning. (1996) 143-151
3. Fabrizio Sebastiani: Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34, No.1, March (2002) pp. 1-47
4. Stanley F. Chen, Joshua Goodman: An Empirical Study of Smoothing Techniques for Language Modeling. Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics
5. Peng, F., Schuurmans, D. and Wang, S: Augmenting Naïve Bayes Classifiers with Statistical Language Models. Information Retrieval, September (2004) vol. 7, no. 3-4, pp. 317-345 (29)
6. Rosenfeld R. Two decades of Statistical Language Modeling: Where Do We Go From Here? Proceedings of the IEEE, (2000), 88 (8)
7. Christopher D Manning, Hinrich Schutze: Foundations of Statistical Natural Language Processing. London: The MIT Press, (1999)
8. Daniel Sleator and Davy Temperley: Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October (1991)
9. Katz, Slave M: Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing, March (1987), ASSP-35 (3): 400-401