Enhancing Naive Bayes with Various Smoothing Methods for Short Text Classification

Quan Yuan
School of Computer Engineering
Nanyang Technological University, Singapore
qyuan1@e.ntu.edu.sg

Gao Cong
School of Computer Engineering
Nanyang Technological University, Singapore
gaocong@ntu.edu.sg

Nadia M. Thalmann
School of Computer Engineering
Nanyang Technological University, Singapore
nadiathalmann@ntu.edu.sg

ABSTRACT

Partly due to the proliferance of microblog, short texts are becoming prominent. A huge number of short texts are generated every day, which calls for a method that can efficiently accommodate new data to incrementally adjust classification models. Naive Bayes meets such a need. We apply several smoothing models to Naive Bayes for question topic classification, as an example of short text classification, and study their performance. The experimental results on a large real question data show that the smoothing methods are able to significantly improve the question classification performance of Naive Bayes. We also study the effect of training data size, and question length on performance.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Filtering

General Terms

Algorithms, Management, Experimentation

Keywords

Short Text Classification, Smoothing, Naive Bayes

1. INTRODUCTION

Understanding the topic of a short text has important applications. For example, it can be employed for topic-based information filtering in Twitter to alleviate the problem of being overwhelmed by a huge number of messages. As another application, it can help advertisers put relevant advertisements along the tweets.

Existing work on classifying short texts focuses on enriching short texts with features extracted from external sources such as search engine results, Wikipedia and WordNet [3]. It is very time consuming to fetching and extracting features from external sources, which render them impractical to applications where short texts are generated in a fast speed. Hence, it is important that the employed classification method is able to accommodate the new training data efficiently, and to classify a new text efficiently.

The Naive Bayes (NB) method is known to be a robust, effective and efficient technique for text classification. More importantly, it is able to accommodate new in-coming training data in classification models incrementally and efficiently. The laplace smoothing is popularly used in NB for text classification. In addition, several other smoothing methods can be combined into the NB model. Unfortunately, the experimental results on normal documents show little performance improvement of other smoothing methods over laplace smoothing for NB [2]. *However, proper smoothing in language models has been shown to be essential for the performance of keyword queries* [4]. *Keyword queries are normally short. Thus we conjecture that the smoothing techniques* [4] *might enhance the performance of NB for short text classification.* We consider four smoothing methods, and incorporate them into NB classifier. To our knowledge, this is the first work that studies the problem of smoothing of NB for short text classification.

We use Yahoo! Webscope dataset that comprises 3.9M questions belonging to 1,097 categories (e.g., travel, health) from a Community-based Question Answering (CQA) service, as an example of short texts, to study question topic classification. This dataset is much larger than those used in the previous work on classifying short texts. Note that the question topic classification problem is different from the question classification task in TREC QA, which is based on the expected answer types (e.g., time, people, numerical value, etc.)

The experimental results show 1) Smoothing methods are able to greatly improve the accuracy of Naive Bayes for short text classification although they can only slightly help for normal documents as shown in [2]. Among the four smoothing methods, Absolute Discounting (AD) and Two-stage (TS) perform the best. The accuracy of NB enhanced by the best smoothing method is comparable with that of SVM, while NB is much more efficient and easier to incorporate new training data. 2) To our surprise, the classification accuracies on questions containing 2-6 words are similar, i.e., questions of 2 words are not more difficult to classify than questions of 6 words, although questions containing 1 word are the most difficult to classify. The smoothing methods consistently improve the performance on questions of different lengths. 3) The smoothing methods are able to enhance performance at different scales of training data.

2. SMOOTHING FOR NAIVE BAYES

Given a question d to be classified, Naive Bayes (NB) assumes that the features are conditionally independent and finds the class c_i that maximizes $p(c_i)p(d|c_i)$.

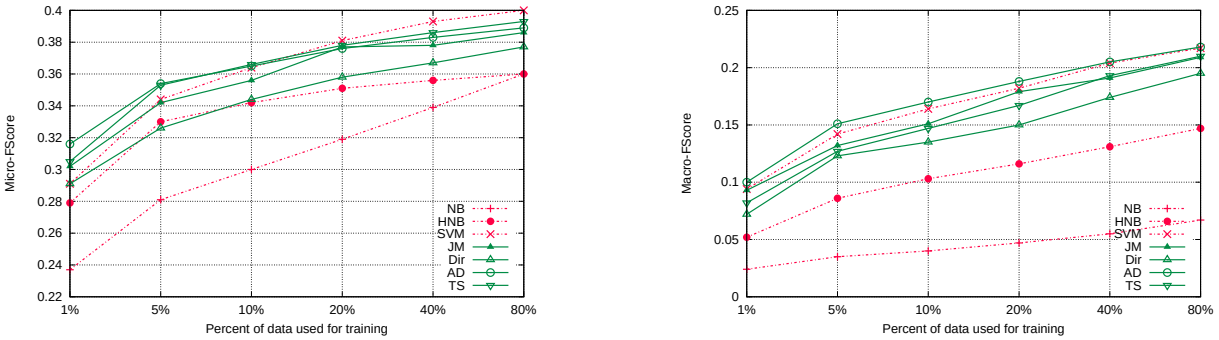
$$p(c_i) = \frac{|c_i|}{|C|}, \qquad p(d|c_i) = \prod_{k=1}^{|d|} p(w_k|c_i)$$

where $|c_i|$ is the number of questions in c_i, and $|C|$ is the total number of questions in collection. For NB, likelihood $p(w_k|c_i)$ is calculated by Laplace smoothing as follows:

$$p(w|c_i) = \frac{1 + c(w, c_i)}{|V| + \sum_{w' \in V} c(w', c_i)}$$

where $c(w, c_i)$ is the frequency of word w in category c_i, and $|V|$ is the size of vocabulary.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.



(a) MicroFscore (b) MacroFscore

Figure 1: The effectiveness of different classifiers when varying the size of training data

With different smoothing methods, $p(w_k|c_i)$ will be computed differently. We consider the following four smoothing methods [4] used in langauge models for information retrieval. Let $c(w, c_i)$ denote the frequency of word w in category c_i, and $p(w|C)$ be the maximum likelihood estimation of word w in collection C

1) Jelinek-Mercer (JM) smoothing:

$$p_\lambda(w|c_i) = (1-\lambda)\frac{c(w,c_i)}{\sum_{w' \in V} c(w', c_i)} + \lambda p(w|C)$$

2) Dirichlet (Dir) smoothing:

$$p_\mu(w|c_i) = \frac{c(w,c_i) + \mu p(w|C)}{\sum_{w' \in V} c(w', c_i) + \mu}$$

3) Absolute Discounting (AD) smoothing:

$$p_\delta(w|c_i) = \frac{max(c(w,c_i) - \delta, 0) + \delta |c_i|_u p(w|C)}{\sum_{w' \in V} c(w', c_i)}$$

where $\delta \in [0, 1]$ and $|c_i|_u$ is the number of unique words in c_i.

4) Two-stage (TS) smoothing:

$$p_{\lambda,\mu}(w|c_i) = (1-\lambda)\frac{c(w,c_i) + \mu p(w|C)}{\sum_{w' \in V} c(w', c_i) + \mu} + \lambda p(w|C)$$

3. EXPERIMENT

Experimental Setting We extract 3,894,900 questions from Yahoo! Webscope dataset. We remove stop words and do stemming. Additionally, we delete the words that occur no more than 3 times in the dataset to reduce misspelling. After preprocessing, the average length of questions is 3.67.

Results of Naive Bayes Using Various Smoothing Methods when Varying Training Data Size We randomly select 20% questions from each category of the whole dataset as the test data. From the remaining 80% data, we generate 7 training datasets with sizes of 1%, 5%, 10%, 20%, 40%, 60%, and 80% of the whole data, respectively. There is no overlap between the training and test data.

Figure 1 shows Micro-F1 and Macro-F1 scores of different methods. For NB, we use multi-nominal NB with Laplace smoothing. We use a small independent development set to set parameters, and we have: $\lambda = 0.5$ for JM; $\mu = 0.95$ for Dir; $\delta = 0.6$ for AD; $\lambda = 0.6$ and $\mu = 100$ for TS. Details are ignored due to space limitation.

We can see: 1) all the smoothing methods greatly improve the classification performance in terms of both Macro-F1 and Micro-F1. Among the 4 smoothing methods, AD generally performs the best in terms of Macro-F1, followed by JM and TS; TS performs the best in terms of Micro-F1. For example, AD improves NB by 33.3% for Micro-F1 and 316% for Macro-F1 at 1% data; 2) Smoothing methods are able to improve the classification performance on training data of different sizes; 3) NB performs much worse than the smoothing methods in Macro-F1. This is probably because Macro-F1 treats each class equally; the Laplace smoothing of NB simply adds 1 to $c(w, c_i)$, resulting in noise, and classes containing few training data are more sensitive to the noise.

For reference, we also compare with Hierarchical Naive Bayes (HNB) [1] and SVM classifier. We use linear SVM in Liblinear toolkit to deal with the large training data. The results in Figure 1 show that NB using AD and TS smoothing, is comparable with and even better than SVM when training is small. Note that NB is much more efficient in training than SVM and can easily incorporate new training data while SVM cannot. For example, on 1% data, NB needs 289 seconds and SVM needs 84,187 seconds, while on 80% data, NB needs 5,241 seconds and SVM needs about 20 hours. Hence, NB with good smoothing is attractive for short text classification.

Table 1: Micro-F1 & Macro-F1 for test questions of different lengths

Length	NB Macro-F1	NB Micro-F1	AD Macro-F1	AD Micro-F1
1	0.0431	0.2557	0.1657	0.2877
2	0.0562	0.3110	0.2109	0.3708
3	0.0593	0.3320	0.2243	0.3935
4	0.0630	0.3417	0.2254	0.4011
5	0.0655	0.3402	0.2351	0.4037
6	0.0598	0.3338	0.2175	0.3946

Results of questions with different lengths This is to study the influence of length of test questions on the classification performance. We still use the 20% data to be the test data, and use 20% of questions as the training data. Table 1 show the Micro-F1 results of NB and AD for questions of different lengths, respectively.

As expected, the classification accuracy is the worst for 1 word questions due to the lack of sufficient information. However, to our surprise, the accuracy on questions of lengths 2–6 is similar. In addition, AD consistently improves the performance of NB for questions of different lengths.

4. FUTURE WORK

It would be interesting to apply the training models built on CQA data to other short texts, such as tweets where there exists no a large labeled data. Additionally, it would be interesting to apply smoothing models to hierarchical NB.

Acknowledgements. Quan Yuan would like to acknowledge the Ph.D. grant from the Institute for Media Innovation, Nanyang Technological University, Singapore. Gao Cong is supported in part by a grant awarded by Microsoft Research Asia and by a Singapore MOE AcRF Tier 1 Grant (RG16/10).

5. REFERENCES

[1] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *ICML*, pp.170–178, 1997.

[2] Jing Bai and Jian-Yun Nie. Using Language Models for Text Classification. In *AIRS*, 2004.

[3] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, pp.91–100, 2008.

[4] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *TOIS*, 22:179 – 214, 2004.