International Symposium on Chinese Spoken Language Processing (ISCSLP 2000)
Fragrant Hill Hotel, Beijing
October 13-15, 2000

Comparison of Several Smoothing Methods in Statistical Language Model

Liu Yang, Sun Jiasong, Wang Zuoying
Department of Electronic Engineering, Tsinghua University, Beijing,
liuyang@thsp.ee.tsinghua.edu.cn

Abstract

With the development of computer technology and the appearance of huge training text corpus, the performance of language model has improved a lot recently. But its intrinsic sparse data problem still exists. This paper investigates several smoothing methods in the application of Chinese continuous speech recognition. We compare the performance of different methods, particularly in the situation of pruned language model and conclude that the Kneser-Ney strategy is better for the model without pruning while its performance decreases for the pruned language model.

1. INTRODUCTION

Language Model is widely used in speech recognition, character recognition and machine translation. It helps to disambiguate by giving a prior probability for a sentence. Now statistical method (N-gram language model) is still the dominant model, although it has its intrinsic disadvantage, for instance, it can’t describe the relation between words longer than N. Statistic language model is straightforward to construct, but it needs a lot of training corpus for reliable probability estimation. The huge number of parameters in statistical N-gram language model makes its order less than 3, namely trigram language model. With the appearance of huge text corpus, there is great improvement in the performance of trigram language model. Despite of this, the sparse data problem still exists. Therefore many smoothing methods have been proposed, which solve this problem to some extent.

S. F. Chen[9] has made a detailed comparison of many state-of-the-art smoothing methods. While in Chinese speech recognition, such work has not been done, particularly for the pruned language model, which is necessary with the increase of text corpus. That is why we attempt to do this research, comparing which smoothing strategy is better in which condition.

This paper is structured as follows: In section 2, we give a brief introduction of speech recognition and data sparse problem in language model. In section 3, we describe various smoothing methods. In section 4, the results of our experiments are presented. Finally in section 5, we summarize the conclusion of our work.

2. LANGUAGE MODEL AND DATA SPARSE PROBLEM

The language model we discuss here in this paper is focused on its application in speech recognition. The model of speech recognition is as follows:

$$\hat{W} = \arg\max_{W} P(X/W)P(W)$$

For the given acoustic data X, the best hypothesis $\hat{W}$ is calculated according to the Bayesian rule. In the formula above, $P(X/W)$ is got from the acoustic model, for which HMM is widely used. If we use statistical trigram language model, $P(W)$ is formulated as:

$$P(W) = \prod_{i=0}^{K-1} P(\,W(i)/W(i-2,i-1)\,)$$

There are a lot of parameters in the language model. For a lexicon with 10 thousand entries, there are possible 10^{12} triples, among which only a small part will appear in the training corpus. This is the problem of sparse data. To address it, some smoothing methods have been proposed. In Table-1, the proportion of triples appearing only once among the whole events is given. According to Good-Turing[1], this proportion also represents the possibility of meeting a new event not occurring in the training corpus. From this table, we can see that even for a training corpus with 200 million words, there is still a large chance of meeting a new triple. So smoothing is necessary, to avoid the zero-probability problem and unreliable probability estimation for those events that don't occur or occur only several times in training corpus. The basic idea for smoothing is to make probability more uniform, by adjusting low probability upward and high probability downward. There are many strategies proposed aiming at this sparse data problem in language model. In this paper, we will give some comparison results in the application of Chinese speech recognition.

3. SMOOTHING METHODS

Firstly we briefly describe Good-Turing estimation, which is the basis for many smoothing methods. Good-Turing used the following formula to get the estimated count for an event that occurred r times:

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

n_r is the total number of different events that occur r times.

Next are some smoothing methods that we will adopt in the following experiments. We use trigram language model for each method.

A. Jelinek-Mercer Interpolation [2] [8]:

$$p(w_3/w_1,w_2)=\lambda_3 f(w_3/w_1,w_2)+\lambda_2 f(w_3/w_2)+\lambda_1 f(w_3)$$

$f(\bullet)$ is the probability estimation using Maximum Likelihood method. The interpolation factors can be calculated by Expectation Maximization method or based on experiments. In theory these factors should change according to the current history. To implement simply, we got the interpolation weight based on the experiments and kept them fixed.

B. Katz Backoff method [3]:

The basic idea of backoff method is to adjust the probability of those seen events downward and allocate this probability to the unseen events. The allocation is based on the probability distribution of lower order model. Katz-backoff method uses the strategy of Good-Turing estimation. It is formulated as follows:

$$p(w(i)/w(i-2,i-1)) = \begin{cases} d_r * \dfrac{N(w(i-2,i))}{N(w(i-2,i-1))} \\ N(w(i-2,i)) = r > 0 \\ \beta(w(i-2,i-1))\,p(w(i)/w(i-1)) \\ N(w(i-2,i)=0 \end{cases}$$

In the formula above,

$$d_r = \frac{\dfrac{(r+1)\,n_{r+1}}{r\,n_r} - \dfrac{(k+1)\,n_{k+1}}{n_1}}{1-\dfrac{(k+1)\,n_{k+1}}{n_1}}$$

$$\beta(w(i-2,i-1)) = \frac{1-\sum_{w(i):C(w(i-2,i))>0} p\,(w(i)/w(i-2,i-1))}{1-\sum_{w(i):C(w(i-2,i))=0} p\,(w(i)/w(i-1))}$$

C. Witten-Bell backoff method[6][7]:

This method was first applied in text

compression [6]. It uses the number of unique words that follow the current history. We use the backoff way for this model as [7]:

$$\hat{p}_{wb}(w_i / w(i-2,i-1)) = \frac{c(w(i-2,i))}{c(w(i-2,i-1)) + N_{1+}(w(i-2,i-1),\bullet)}$$

The backoff factor is:

$$\beta = \frac{\hat{p}_{0/w(i-2,i-1)}}{1-\sum_{w\in W(x)} \hat{p}_{w/w(i-1)}}$$

$$\hat{p}_{0/w(i-2,i-1)} = \frac{N_{1+}(w(i-2,i-1),\bullet)}{c(w(i-2,i-1))+N_{1+}(w(i-2,i-1),\bullet)}$$

$$N_{1+}(w(i-2,i-1),\bullet) = |\, w_i : c(w_{i-2}, w_i) > 0\}|$$

represents the number of different words following the word w.

Usually this method has almost the same performance as Katz backoff. But in dealing with abnormal data, that is when Good-Turing assumption (the number of events occurring once is more than that occurring twice, the number of events occurring twice is more than that occurring three times, and so on) is not right, it is more robust than Katz backoff.

D. Kneser-Ney smoothing method [4] [5]:

Kneser-Ney proposed a newer way to estimate probability. It assumed that the probability for unigram should not only consider the count of that word itself, but also how many different words that this word could follow. Here we adopt the interpolated version suggested by Chen-Goodman [9]:

$$p_{KN}(w_i / w_{i-n+1}^{i-1}) = \frac{\max\{c(w_{i-n+1}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}$$

$$+ \frac{D}{\sum_{w_i} c(w_{i-n+1}^{i})} N_{1+}(w_{i-n+1}^{i}\bullet) p_{KN}(w_i / w_{i-n+2}^{i-1})$$

Unigram probability: $p_{KN}(w_i) = \frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)}$

$N_{1+}(\bullet w_i) = |\{w_{i-1} : c(w_{i-1} w_i) > 0\}|$ represents how many unique words that the word can follow in the training corpus.

$$N_{1+}(\bullet\bullet) = \sum_{w_i} N_{1+}(\bullet w_i)$$

4. RESULTS OF EXPERIMENT

We implemented the smoothing methods mentioned above using different training size and applied these smoothing models in our speech recognition system. Their performance in our recognition system is listed in the tables below. Considering Perplexity is not always correlated well with recognition result, here we compare the recognition right rate of different smoothing methods. The four methods are represented respectively as: interpolation, Katz, W-B, K-N.

We used the simple pruning method, that is, deleted those events whose count is less than the pruning threshold while keeping the factors for backoff or interpolation unchanged. From Table-2, we can see that for the condition of unpruned model Kneser-Ney usually performs better, Katz and Witten-Bell take second order, interpolation the worst. In Table-3, we give the recognition result for interpolation and Kneser-Ney methods using different size of training corpus, which shows that the smaller the training corpus, the worse the interpolation method. While with pruning, Kenser-Ney doesn’t perform well. The reason for this is that, Kneser-Ney method is sensitive to the number of unique words that a word can follow and that follow the word. Pruning change these parameters a lot, thus causing the interpolation or backoff factors not right and decreasing the recognition performance. In our experiments, interpolation model performs worst, that is because we use fixed interpolation factors and these fixed factors are not optimal. So from our experiments, we can’t say that Jelinek-Mercer interpolation model has the worst performance.

5. CONCLUSION

Smoothing is an important technique for statistical language model. In this paper we implemented several smoothing methods dominant in language model and compared their performance when used in our speech recognition system. This is the first attempt of such research in Chinese speech recognition. We made some conclusion about how factors, such as training corpus and pruning, affect the performance of these different methods. We are going to do some more systematic work on this topic.

6. REFERENCE

[1] I.J.Good, *The population frequencies of Species and the Estimation of Population Parameters,* Biomatrika, 40:237-264 1953

[2] Bahl.L, Jelinek.p.F, Mercer,R.L, *A Maximum Likelihood Approach to Continuous Speech Recognition,* IEEE Transactions on PAMI, v5, no2, pp179-190, March, 1983

[3] Slave M. Katz, *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer,* IEEE Transaction on ASSP-35 no.3 pp.400-401, 1987

[4] H.Ney, U.Essen, R.Kneser, *On the Estimation of 'small' Probability by Leaving-One-Out* IEEE Transactions on PAMI No 12, pp.1202-1212, 1995

[5] Reinhard Kneser, Hermann Ney, *Improved Backing-off for M-gram Language Modeling.* In proceedings of ICASSP, v1, pp181-184, 1995

[6] T.C.Bell, J.G.Cleary, I.H.Witten, Text Compression, Prentice Hall, 1900

[7] P.Placeway, R.Schwartz,P.Fung,L.Nguyen, *The Estimation of Powerful Language Models From Small and Large Corpora,* In proceedings of ICASSP 1993, v2, pp33-36

[8] F.Jelinek, *Self-organized Language Modeling for Speech Recognition*, in Readings in Speech Recognition, pp450-504, 1990

[9] Stanley.F.Chen, J.Goodman, *An Empirical Study of Smoothing Techniques for Language Modeling,* Computer Speech and Language, pp.359-394, 1999

Table 1: proportion of triples occurring once to the whole events using different size of training corpus

Training corpus	20million	40million	200million
Percent	0.3487	0.2905	0.1885

Table 2: recognition result (%) for four smoothing methods using different size of training corpus

Training corpus	Interpolation	Katz	W-B	K-N
5million（without pruning）	81.88	82.53	82.40	82.66
40milliion（without pruning）	84.52	84.84	85.07	84.89
200million （without pruning）	91.13	91.24	91.33	91.30
200million （pruning threshold 10）	85.29	85.71	85.50	85.14

Table 3: recognition result (%) for interpolation and Kneser-Ney using different size of training corpus

Training corpus	50 thousand	50thousand	20million	40million	200million
Interpolation	62.93	74.03	82.6	84.52	90.1
Kneser-Ney	64.33	75.55	83.39	85.07	90.39