ISCA Archive
http://www.isca-speech.org/archive

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

SEMI-CLASS-BASED N-GRAM LANGUAGE MODELING FOR CHINESE DICTATION

Min ZHANG *Engsiong CHNG* *Haizhou LI*

Lernout & Hauspie Asia Pacific, 29 International Business Park, #08-05 Acer Tower B, Singapore 609923

{ Min.Zhang, Engsiong.Chng, Haizhou.Li }@lhsl.com.sg

www.lhsl.com.sg

ABSTRACT

In this paper, we propose a novel semi-class-based n-gram language modeling. The proposed modeling estimates the n-gram probability from the observed frequencies of word-class n-tuples, constituted by the (n-1) classes of preceding (n-1) words of the utterance and the current word itself. Three kinds of language modeling, word-based, class-based and semi-class-based n-gram modeling are implemented to build bi-gram and tri-gram models for a vocabulary of 50k words over a corpus of over 200 millions Chinese words. The parameter numbers and LM perplexities among the three models have been studied and compared. Our experiments show that our proposal of using the semi-class language modeling is a good tradeoff between the number of parameters and LM perplexity.

1. INTRODUCTION

This paper discusses the semi-class-based n-gram Language Modeling (LM) used in the Lernout & Hauspie Chinese dictation products.

In recent years, the n-gram language modeling approach has gained popularity in the applications of speech recognition. In general, we use two types of n-gram LMs - word-based and class-based.

Word-based n-gram approach estimates probabilities from the observed frequencies of word n-tuples in the training corpus. This approach has the advantage of high discrimination and lower perplexity if the training corpus is sufficient. This method however suffers from the drawbacks of having huge number of parameters and training data sparseness. In contrast, the class-based n-gram estimates probabilities from the observed frequencies of class n-tuples in the training corpus. By using class-based approach, we improve the robustness of statistical LM. As a result, the perplexity of the class-based n-gram is however increased a lot as compared to the word-based approach.

In order to take advantages of both of the above two types of n-gram modeling and to reduce their disadvantages, we propose a new semi-class-based n-gram approach in this paper. The proposed modeling estimates the n-gram probability from the observed frequencies of word-class n-tuples, constituted by the (n-1) classes of preceding (n-1) words of the utterance and the current word itself. In section 2, the proposed semi-class-based modeling will be studied in detail. In section 3, we will show our experiment and give our conclusion.

2. SEMI-CLASS-BASED N-GRAM LANGUAGE MODELING

Word-based and class-based n-grams are two types of popular n-gram.

2.1 Word-based N-Gram

Word-based n-gram approach estimates probabilities from the observed frequencies of word n-tuples in the training corpus. In other words, this approach calculates probability of the current word $w(n)$ conditioned on the identity of the preceding $n-1$ words, which can be formularized as the following equation:

$$p(w_1^N) = p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid w_1 \dots w_n) \qquad （1）$$

where w_n means the n-th word in the sentence w_1^N. In general, we only take the immediate previous one or two words as the history condition, namely bi-gram and tri-gram as follows:

$$p(w_1^N) = p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid w_n) \qquad (2)$$

$$p(w_1^N) = p(w_1) * p(w_2 \mid w_1) \prod_{n=1}^{N-1} p(w_{n+1} \mid w_n w_{n-1}) \qquad (3)$$

Because this approach is word-based (instead of part-of-speech or other categories-based), this approach has the advantage of high discrimination and lower perplexity if the training corpus is sufficient. For the same reason, this approach however suffers from the drawbacks of having huge number of parameters and training data sparseness.

2.2 Class-based N-Gram

To overcome the problems of the word-based n-gram, P. F. Brown *et al* [1] proposed the class-based n-gram. To generate class-based n-gram, we first partition the whole vocabulary into a number of classes such that words within the same class share similar syntactic and semantic functionality. Instead of the word itself, the class of word will be used to calculate the n-gram probability. The class-based n-gram estimates conditional probability of the current word from the observed frequencies of class n-tuples in the training corpus, constituted by the (n-1) classes of preceding (n-1) words of the utterance and the class of the current word itself. In case of bi-gram modeling we can formulate this approach using the following equation:

$$\begin{aligned} p(w_1^N) &= p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid w_n) \\ &\approx p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid c_n) \qquad (4) \\ &\approx p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid c_{n+1}) * p(c_{n+1} \mid c_n) \end{aligned}$$

where w_n means the n-th word in the sentence w_1^N, and c_n means the class of the word w_n.

By using class-based approach, we improve the robustness of statistical LM. As a result, the perplexity of the class-based n-gram is however increased a lot as compared to the word-based approach.

2.3 Semi-class-based N-Gram

In order to take advantages of the class-based and word-based n-gram while maintaining low LM perplexity, we propose a semi-class-based n-gram approach.

Similar to class-based n-gram approach, the semi-classed n-gram approach also partitions the whole vocabulary into a number of classes. The difference is that the semi-class-based n-gram estimates probability from the observed frequencies of word-class n-tuples in the training corpus. In particular, the probability of a word is calculated using the frequency of the n-tuple, constituted by the (n-1) classes of preceding (n-1) words of the utterance and the current word itself. The following equation formulate the semi-class-based bi-gram model:

$$\begin{aligned} p(w_1^N) &= p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid w_n) \\ &\approx p(w_1) * \prod_{n=1}^{N-1} p(w_{n+1} \mid c_n) \end{aligned} \qquad (5)$$

where w_n means the n-th word in the sentence w_1^N, and c_n means the class of the word w_n.

Comparing equation (4) to equation (5), we can see that the expression $p(w_{n+1} \mid c_{n+1}) * p(c_{n+1} \mid c_n)$ in equation (4) is used as an estimate of the expression $p(w_{n+1} \mid c_n)$ in equation (5). In the semi-class-based n-gram, we explicitly estimate $p(w_{n+1} \mid c_n)$ from the training corpus as LM parameters instead of computing it on the fly using the expression $p(w_{n+1} \mid c_{n+1}) * p(c_{n+1} \mid c_n)$. Using equation (5), we can achieve good perplexity results with no significant increase in the number of parameters.

In L&H Chinese dictation products, two kinds of language models, class-based bi-gram and semi-class-based tri-gram are used to make a good tradeoff between the recognition accuracy, speed, memory usage and the large number of parameters used in language modeling. Firstly, class-based bi-gram is employed to work together with acoustic modeling to decode the N-best recognition sentence results from lattice. The semi-class-based tri-gram is then applied on the selected top-N sentences to get the final result.

3. EXPERIMENT

Three kinds of language modeling, word-based, class-based and semi-class-based n-gram modeling are implemented to build bi-gram and tri-gram models for study. We only use class-based bi-gram and semi-class-based tri-gram in our products.

The training corpus includes over 200 millions Chinese words, we use our Chinese segmentation tool to do segmentation and text normalization [5]. All of the ASCII characters in our corpus is mapped into our pre-defined 12 classes. For example:

Input Chinese: 月薪800多元

(The monthly salary is over 800$.)

Segmentation: 月薪 <DIGITS> 多 元 _S_

In the example above, the digit “800” is mapped into the class “<DIGITS>” after segmentations.

The number of lexicon entry is 50k, and we use the maximum mutual information (MMI)-based clustering algorithm to partition the whole vocabulary into 2000 classes [1].

Training data sparseness is an open problem for LM training. After comparing several smoothing methods, we use the bounded model of absolute discounting algorithm to get discount from the low-frequency gram and predict the probability of the unseen gram [2].

Experiments are carried out on 400MHz Pentium III, 1GB memory computer.

Our LM training includes the following three steps:

1. Corpus segmentation and text normalization:

   It took 1 day to process 800M bytes Chinese corpus [5].

2. n-gram word pair counting:

   The purpose of this job is to get the distinct seen n-gram word pairs and their counts from the training corpus. It took 1 hour to count the b-gram and tri-gram word pairs from the 800M bytes training corpus.

3. Word class training:

   It took 3 days to assign words to classes using the MMI-based clustering algorithm based on the word pair counting file generated in the step 2 [1]. The lexicon size is 50K.

4. LM training:

   It took about 13 hours and 26 hours to train the semi-class-based and word-based tri-gram, respectively. It only took a few minutes to train other kinds of modeling.

The parameter numbers and LM perplexities among the three models have been studied and compared. As we can see from equation (4), the number of parameters in the class-based bi-gram is not large, only 4M[1], it is very possible to load this kind of LM into memory during search processing, while a semi-class bi-gram modeling in equation (5) has about 10M parameters alone without taking data addressing overhead into accounts. As such, this class-based bi-gram language model is first used with the acoustic model to decode the N-best sentence results from lattice in the first pass.

Table-1 illustrates the parameter numbers of the three kinds of tri-gram. Table-2 lists the perplexities of the six different kinds of LM. The close test data is our 800MB training corpus, and the open test data is 861KB.

Table 1. Parameter numbers

	Parameter numbers of different tri-gram models
Class	$8.00 * 10^9$
Semi-class	$1.00 * 10^{11}$
Word	$1.25 * 10^{12}$

[1] 4M = 2000*2000, 2000 is the total class number.

Table 2. Perplexities of different models

	Bigram open (*close*) test	**Trigram open (*close*) test**
Class	649.44 (*484.03*)	146.63 (*115.25*)
Semi-class	479.84 (*367.08*)	104.23 (*62.86*)
Word	332.33 (*264.49*)	83.75 (*47.28*)

The number of parameters with class-based LM is small, which is good for first pass search whereby memory is of first concern.

Our LM training and testing results also show that the perplexity of the semi-class tri-gram is reduced significantly compared with that of the class-based model and the number of parameters of the semi-class tri-gram is also reduced significantly compared with that of the word-based model. So semi-class-based tri-gram is applied on the selected top-N sentences that are generated from the first pass to get the final recognition result in the second pass.

In addition, from the Table-2, we can see that the perplexity of the class-based tri-gram is reduced 56% as compared with that of the word-based bi-gram. So we can say that the order of n-gram has higher influence on the perplexity than other factors, such as class, semi-class, word and so on.

In conclusion, we show that our proposal of using the semi-class model is a good tradeoff between the number of parameters and LM perplexity.

REFERENCES

[1] P. F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai and R.L. Mercer. “Class-based n-gram models of natural language”. *Computational Linguistics*, 18(4): 467-480, December 1992

[2] H. Ney, S.Martin and F.Wessel, “Statistical Language Modeling Using Leaving-One-Out”, chapter 6 in the book <<Corpus-based methods in language and speech processing>> edited by Steve Young and Gerrit Bloothooft. Kluwer Academic Publishers, 1997

[3] Frederick Jelinek, “Statistical Methods for Speech Recognition”, The MIT Press, Cambridge, Massachusetts, London, England, 1997

[4] Thomas Niesler, “Category-based Statistical Language Models”, Ph.D thesis, University of Cambridge, 1997

[5] Li Haizhou, Bai Shuanhu and Lin Zhiwei, “Chinese Sentence Tokenization Using Viterbi Decoder”, 1998 International Symposium on Chinese Spoken Language Processing Symposium Proceeding, 151-154, Singapore 1998