社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－S L P－69 (30)
2007／12／20

Word and Class Dependency of N-gram Language Model

Welly NAPTALI† Masatoshi TSUCHIYA‡ and Seiichi NAKAGAWA†

† Department of Information and Computer Sciences, ‡ Information and Media Center
Toyohashi University of Technology, 1-1, Hibarigaoka, Tenpaku-cho, Aichi, 441-8580, Japan
E-mail: † {naptali,nakagawa}@slp.ics.tut.ac.jp, ‡ tsuchiya@imc.tut.ac.jp

Abstract We propose another way to calculate language model (LM) probability by making an assumption that the current word/the predicted word depends on word and class history. Experiment result on Wall Street Journal (WSJ) corpus shows that the proposed method is better than a traditional class-based n-gram LM.

Keyword N-gram, Class Language Model, Class Dependent.

1. Introduction

The speech recognition task is to find the corresponding word sequence for given an acoustic signal. Let A be an acoustic input, the corresponding word sequence W is a word sequence that has maximum posterior probability $P(W|A)$ given by the following equation:

$$\hat{W} = \arg\max_{W} \left(\log P_A(A|W) + \lambda \log P_L(W) \right), \quad (1)$$

where $P_A(A|W)$ is the probability of A given W based on an acoustic model, $P_L(W)$ is the probability of W based on an LM, and λ is a scaling factor (language weight). Both acoustic model and language model are important studies in a modern automatic speech recognition system.

The LM purpose is to assign probabilities to word sequences. Word-based n-gram is the most widely used LM. It is a simple and powerful method based on the assumption that the current word depends on only $n-1$ preceding words. In the case of trigram ($n=3$), the LM gives the following probability to a word sequence $W=w_1, w_2, \ldots, w_N$:

$$P_{TRIGRAM}(W) = \prod_{i=1}^{N} P(w_i \mid w_{i-2}, w_{i-1}). \quad (2)$$

The parameters of the LM are usually trained from a very large corpus. If the corpus is not large enough, word which occurs only few times will have unreliable probability which is known as a data sparseness problem. This is a serious problem and frequently occurs in many LMs. The problem is often solved partially using a good smoothing technique.

Another way to solve the problem is by using a class LM. A class-based n-gram LM [1,2] maps words into classes, resulting an LM with less parameters. But this makes the LM hard to recognize different histories [3], which can only be encountered by increasing the number of the context. In the other side, the increasing context will cause a parameter size larger, which is not good for application in which system resources are constrained, such as handheld computers.

A common way to improve a class-based n-gram LM is to interpolate it with a word-based n-gram LM using interpolation [4,5]. If two LMs model a different part of the language, the interpolation will leads to further improvements. Another approach is using a class-based n-gram LM to predict the unseen event while the seen event predicted by a word-based n-gram LM. This method is known as word-to-class backoff [6]. But when using both a word-based LM and a class-based LM, the size of parameters will be larger than the previous case.

There are many other class LM that give better result than the conventional class-based n-gram LM [7], but they use more complex formulation. In this paper, we propose an alternative class-based LM in a simple way, an LM that maintains its ability to recognize the different histories, and an LM that gives a fair tradeoff between the performance and its parameter size. With a simple assumption, we have a more specific class-dependent LM in comparison with the conventional class-based n-gram LM.

This paper is organized as follows. Section 2 gives a brief overview about the conventional class-based n-gram LM. In Section 3, we describe the proposed method in detail. Section 4 reports experiments carried out and the results. Then the last section is a summary.

2. Class-based *N*-gram Language Model

A class-based n-gram LM [1] was proposed to counter the data sparseness problem. Without loss of generality, let us consider a bigram case and denote C_i for class of word w_i. Given w_i, w_{i-1} and its class $C_i C_{i-1}$, the probability of current word w_i given the history w_{i-1} is calculated according to

$$P_{CLASS}(w_i \mid w_{i-1}) = P(w_i \mid C_{i-1}, w_{i-1}, C_i) P(C_i \mid C_{i-1}, w_{i-1}). \quad (3)$$

Assume that $P(w_i|C_{i-1}, w_{i-1}, C_i)$ is independent on C_{i-1}, w_{i-1}, and $P(C_i|C_{i-1}, w_{i-1})$ is independent on w_{i-1}. Then Equation (3) becomes

$$P_{CLASS}(w_i \mid w_{i-1}) = P(w_i \mid C_i) P(C_i \mid C_{i-1}), \quad (4)$$

which is known today as a class-based bigram LM.

In general, the probability of word sequence W is defined by

$$P_{CLASS}(W) = \prod_{i=1}^{N} P(w_i \mid C_i) P(C_i \mid C_{i-n+1}, \ldots, C_{i-2}, C_{i-1}). \quad (5)$$

Instead of words, a class-based n-gram LM estimates parameters for word classes. By mapping words into classes, this model significantly reduce the parameters size. As a tradeoff, the performance of this model is slightly decreasing compared to word-based n-gram LM. We call Equation (5) the baseline class-based LM.

3. Class Dependent Language Model

The traditional class LM as shown in Equation (5) is defined based on the assumption that the current word depends only on its own class, and independent to the preceding word class. These independent assumption causes information loss on the language itself. Thus, we propose a class-based LM without the independency.

We define a class dependent (CD) LM with an assumption that the current word depends on the preceding word classes,

$$P_{CD}(W) = \prod_{i=1}^{N} P(w_i \mid C_{i-n+1}, \ldots, C_{i-2}, C_{i-1}), \quad (6)$$

This formulation is similar to those of word-based n-gram LM, only by changing its words history with classes history. Because of that, other aspect of word-based n-gram LM (such as smoothing, discounting, and backoff method) can be adopted easily to be used in class dependent LM.

For simplicity, let us consider a bigram case. Unseen events are backed-off according to Katz-backoff defined by

$$P_{katz}(w_i \mid C_{i-1}) = \begin{cases} P^*(w_i \mid C_{i-1}) & : c(C_{i-1}, w_i) > 0 \\ \alpha(C_{i-1}) P_{katz}(w_i) & : otherwise \end{cases}, \quad (7)$$

where $c(.)$ means the frequency of occurrence of a particular sequence in a training data/corpus. Probability $P^*(.)$ and backoff weight $\alpha(.)$ are calculated by the following equations

Table 1 Data statistics

WSJ	#Word	OOV Rate
Training Set	36,754,891	0.0236
Test Set	336,096	0.0243

$$P^*(w_i \mid C_{i-1}) = d_{c(C_{i-1}, w_i)} \frac{c(C_{i-1}, w_i)}{c(C_{i-1})}, \quad (8)$$

$$\alpha(C_{i-1}) = \frac{1 - \sum_{w_i : c(C_{i-1}, w_i) > 0} P^*(w_i \mid C_{i-1})}{1 - \sum_{w_i : c(C_{i-1}, w_i) > 0} P^*(w_i)}. \quad (10)$$

where $d_{c(.)}$ is a discount coefficient factor for events that occurs $c(.)$ times in the training corpus.

The probability of unseen events is obtained by redistributing the leftover probability collected by smoothing method from discounting all seen events. Absolute discounting discounts all non-zero counts events by a constant m, where the discounting coefficient factor is defined by

$$d_a = \frac{a - m}{a}, \quad (9)$$

and the constant m follows the following rule

$$m = \begin{cases} \frac{c_1}{c_1 + 2c_2} & , if\ c_1 > 0\ and\ c_2 > 0 \\ 0.5 & , otherwise \end{cases} \quad (11)$$

4. Experiments

4.1. Experimental Setup

Experiments were conducted using WSJ corpus from year 1987 to year 1989 consisting of 37 million words. The data is divided into training set and test set as given by Table 1. The vocabulary is used ARPA's official "20o.nvp" (20k most common WSJ words, non-verbalized punctuation). By inserting a beginning sentence, an end sentence, and an OOV symbols, the total vocabulary size is 19,982 words and the OOV rate is about 2.4%.

The baseline, class-based n-gram LM, was build using HTK Language Model toolkit [3]. The clustering also conducted using the same toolkit based on statistically-derived class mapping. This is a hard clustering method, that is, means that one word is assigned to only one class. We also build word-based n-gram LM for comparison. Katz-backoff is used together with absolute discounting.

To fairly compare the parameter size, no pruning is applied. Although the parameter size is greatly decreasing with pruning, it has often to be paid with another slight decreasing on performance of the LM. For performance comparison, the models are evaluated using *perplexity* (*PP*) as defined by the following equation

Table 2 Perplexity (Number of classes = 2k)

No	Model	Word-based *n*-gram	Class-based Baseline	Class Dependent
1	2-gram	177.15	207.01	193.44
2	3-gram	111.55	132.38	121.51
3	4-gram	101.43	121.46	111.10

Table 3 Parameters size (Number of classes = 2k)

No	Model	Word-based *n*-gram	Class-based Baseline	Class Dependent
1	2-gram	3,490,635	1,544,443	2,532,876
2	3-gram	14,019,603	11,033,279	12,149,761
3	4-gram	24,284,552	22,278,819	22,829,468

$$PP = 2^{-\frac{1}{N}\log_2 P_L(W)}. \qquad (12)$$

4.2. Results

First, let us consider class LMs with 2000 classes case. The perplexity results are given by Table 2. The bigram model of CD achieved 193.44 perplexity, which means 6.56% relative improvements against the baseline. The improvements for trigram and fourgram is 8.20% and 8.53% relative, respectively. The improvements are followed by the increasing parameters size.

Table 3 shows the parameters size for each corresponding LM. Larger parameter's size means that the CD LM is able to recognize more history than the baseline LM. Notice that trigram CD has significantly smaller parameters size than the fourgram baseline, almost half size smaller, but it has a comparable performance.

Next, we conducted the LMs for various numbers of classes. We would like to see the robustness against number of classes. The results of perplexity against number of classes is given by Fig. 1. The corresponding parameter's size is shown in Fig. 2. When using 1.5k classes, the proposed method gives 8.60%, 10.52%, and 10.94% relative improvements on perplexity for bigram, trigram, and fourgram model respectively. With 1k classes, the improvements is 11.44%, 14.08%, and 14.84% respectively. The largest improvements are achieved by the smallest number of classes 0.5k, which are 16.20%, 20.57%, and 21.81% respectively. These improvements show the robustness of the proposed method against small number of classes.

Another thing that we can analyze from the results is that the trigram model of CD gives better perplexity than the fourgram baseline, even though the parameter's size is

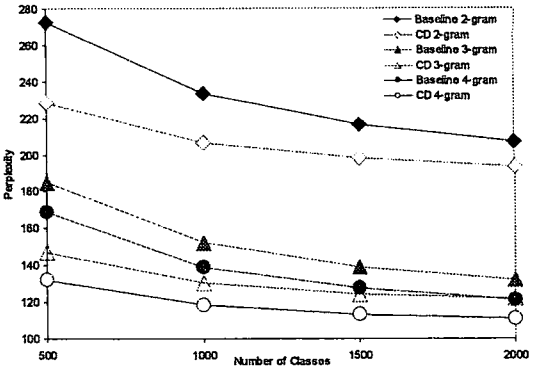

Fig. 1 Perplexity against number of classes

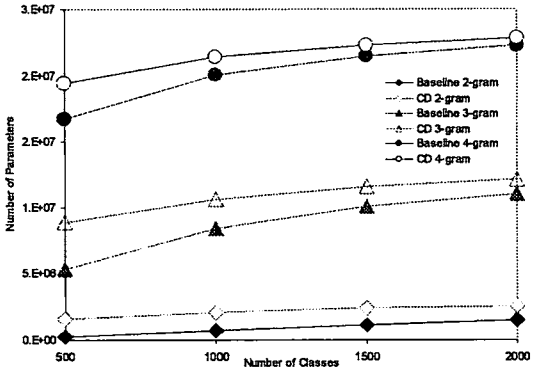

Fig. 2 Parameters size against number of classes

half smaller. Such small parameters size will be useful for low-latency or low-resource application in which system resources are limited. To further analyze the performance of LMs against the parameter's size, or to see the tradeoffs between performance and parameter's size, we plot the perplexity with its corresponding parameter's size in Fig. 3.

4.3. Combination of word-based trigram with class-based fourgram

In this section, we tried to analyze the performance of the proposed method when it is interpolated with another LM which models a different aspect of the language. To improve the performance of class LM, the most common way is to combine it with a word-based *n*-gram LM. Here, we used linear interpolation to combine the two LMs as defined by

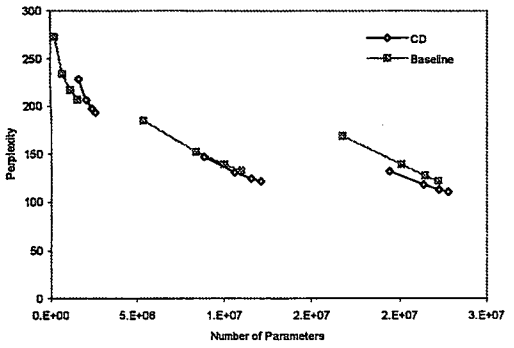

Fig. 3 Perplexity against parameters size for class LMs

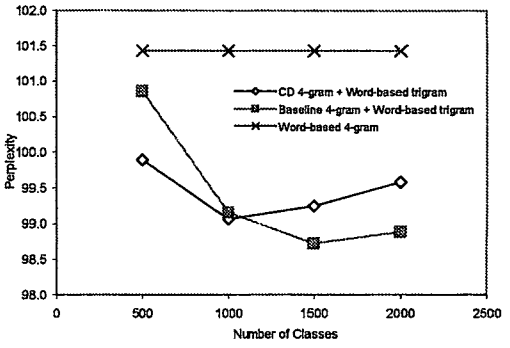

Fig. 4 Perplexity against number of classes of interpolated models with word-based trigram

$$P_L \equiv \beta P_{CLASS/CD} + (1-\beta) P_{WORD}, \qquad (13)$$

where β is a interpolation weight/constant. The weight is variated from 0.1 to 0.9, and show only the best result.

For this experiment, we choose the class fourgram to be interpolated with the word-based trigram, and the result is given by Fig. 4. Although the best perplexity is achieved by the interpolated 4-gram baseline with 1.5k classes, from the point of view number of classes, the interpolated 4-gram CD gives better perplexity in small number of classes. With 1k classes, the interpolated 4-gram CD has 0.09% relative better perplexity compared to the interpolated 4-gram baseline. Smaller number of classes, 0.5k classes, the improvement is increasing into 0.96% relative.

One of the reasons that the interpolated class-based *n*-gram LM baseline achieved better result than CD LM is because there are some information similarities between word-based *n*-gram and CD LM. In other words, CD LM contained some language information which has already modeled by word-based *n*-gram LM, while the class-based *n*-gram LM modeled a different part of the language. This fact is supported by the similarity of Equation (2) and Equation (6).

5. Summary

In this paper, we have presented a simple way to improve class-based LM. A class LM that is more specific so that the LM does not lose its ability to recognize a different history. We have showed that the proposed method, class dependent LM, gives better perplexity than the conventional class-based *n*-gram LMs. The proposed method gives better performance with a reasonable parameters size, and robust against small number of classes. The class dependent LM can be an alternative LM to be applied in a low-latency or low-resource environment.

References

[1] Brown, P.F., Pietra, V.D., De Souza, P., Lai, J.C., and Mercer, R., "Class-based n-gram models of natural language", Computational Linguistics, Vol. 18, No. 4, pp. 467-479, 1992.

[2] Samuelsson, C., Reichl, W., "A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics", IEEE Proc ICASSP99, Vol. 1, pp. 537-540, 1999.

[3] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "The HTK Book (for HTK Version 3.3)", Cambridge University Engineering Department, 2005.

[4] Broman, S., Kurimo, M., "Methods for Combining Language Models in Speech Recognition", Interspeech 2005, pp. 1317-1320. Lisbon, Portugal, September 4-8, 2005.

[5] Wada, Y., Kobayashi, N., Kobayashi, T., "Robust language modeling for a small corpus of target tasks using class-combined word statistics and selective use of a general corpus", Systems and Computers in Japan, Vol. 34, No. 12, pp. 92-102. 2003.

[6] Niesler, T.R., Woodland, P.C., "Combination of word-based and category-based language models", Proc ICSLP96, pp. 1779-1782, 1997.

[7] Nakagawa, S., "A survey on automatic speech recognition", IEICE Transactions on Information and Systems, Vol. E85-D, No. 3, pp. 465–486, March 2002.