CLASS-BASED VARIABLE MEMORY LENGTH MARKOV MODEL

Shinsuke MORI and Gakuto Kurata

IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma Yamatoshi Kanagawaken 242-8502 Japan

Abstract

In this paper, we present a class-based variable memory length Markov model and its learning algorithm. This is an extension of a variable memory length Markov model. Our model is based on a class-based probabilistic suffix tree, whose nodes have an automatically acquired word-class relation. We experimentally compared our new model with a word-based bi-gram model, a word-based tri-gram model, a class-based bi-gram model, and a word-based variable memory length Markov model. The results show that a class-based variable memory length Markov model outperforms the other models in perplexity and model size.

1. Introduction

In statistical methods for natural language processing, such as speech recognition, the n-gram model based on words is popular for its ease of parameter estimation and implementation. In fact, most of current speech recognizers are based on word-based n-gram models.

In a word-based n-gram model however, the number of parameters is equal to the number of vocabulary words to the n-th power so that it is not possible to estimate the parameters accurately when only a limited corpus is available. As a result, the model is less predictive than the word n-gram model estimated from a corpus of ideal size. To cope with this problem, many variations have been proposed, such as a class-based n-gram model [1] and a variable memory length Markov model [2].

In the class-based n-gram model, each word belongs to a group of words, called a class, and is predicted through statistical analysis of the class sequences, which are more reliable than that of word sequences. In addition, this method decreases the memory size for the model description. Normally the optimum word-class relation for word sequence prediction is found by automatic word clustering [1] [3] [4]. The n-gram models based on automatically acquired classes are much smaller than a word-based n-gram for the same n and that the accuracy of the class-based n-gram model is better than or comparable to the word-based n-gram model.

A variable memory length Markov model [2] is also a variant of an n-gram model. In this model, the length of each n-gram is increased selectively according to an estimate of the resulting improvement in predictive quality. For example, it may happen that in case the previous word is "I," a variable memory length Markov model does not distinguish the word before the previous word like a word-based bi-gram model, but if the previous word is "of," the same variable memory length Markov model uses the word before the previous word to help predict the next one like a word-based tri-gram model. The word three word before can also be checked out if it is considered to have some information about the next word to be predicted. Thus a variable memory length Markov model of the same size as an n-gram model is expected to have higher predictive power and it has a tendency to be smaller while achieving the same predictive power as an n-gram model.

In this paper we present a class-based variable memory length Markov model. This is an extension of a variable memory length Markov model based on a probabilistic suffix tree (PST). Our model is based on a class-based PST whose nodes have a word-class relation. This allows us to treat some similar contexts, such as "the 1st of," "the 2nd of," ..., "the 31st of," as a single context and to reduce the data-sparseness problem to build a more accurate language model. A class-based variable memory length Markov model is expected to be smaller and have more predictive power than a word-based variable memory length Markov model. An incorpolation of the notion of class into a variable memory length Markov model has already been presented [5]. In this research, first a word-based variable memory length Markov model is built and then some nodes are merged. In this paper we present a learning algorithm which executes node expansion and word clustering at the same time. Experimental results show that class-based variable memory length Markov model is as good as the word-based variable memory length Markov model and the word-based tri-gram model, and better in model size than the word-based variable memory length Markov model, class-based bi-gram model, and the word-based tri-gram model built from the same corpus.

2. Word-based Language Models

In this section, first we describe a word-based n-gram model, a class-based n-gram model, and a variable memory length Markov model.

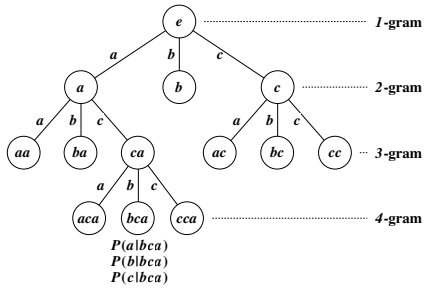


Figure 1: A word-based context tree

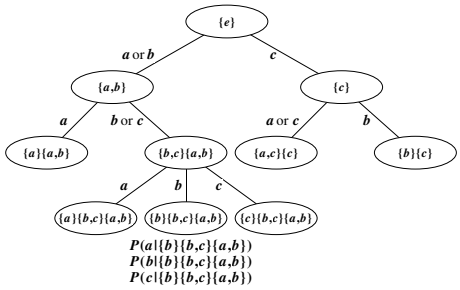


Figure 2: A class-based context tree

2.1. Word-based n-gram model

The most common stochastic language model is a word-based n-gram model. This model predicts each word in a sentence from left to right considering the last $k = n - 1$ words as the history as follows:

$$P(\boldsymbol{w}) = \prod_{i=1}^{m} P(w_i | w_{i-k} w_{i-k+1} \cdots w_{i-1}),$$

where $\boldsymbol{w} = w_1 w_2 \cdots w_m$ denotes the word sequence of the sentence. The probabilities $P(w_i | w_{i-k} w_{i-k+1} \cdots w_{i-1})$ in this formula are estimated from a learning corpus by the maximum likelihood estimation method.

2.2. Class-based n-gram model

In a class-based n-gram model [1], words are grouped into classes and the model predicts first the next class from the last $(n-1)$ class sequence and then predicts the next word from the predicted class as follows:

$$P(\boldsymbol{w}) = \prod_{i=1}^{m} P(c_i | c_{i-k} c_{i-k+1} \cdots c_{i-1}) P(w_i | c_i),$$

where c_i is the class which the i-th word belongs to. In this formula, it is assumed that each word belongs to a single class. A class-based n-gram model is expected to be smaller and more accurrate in prediction than a word-based n-gram model. Aiming at acquiring an appropriate word-class relation for a class-based n-gram model, some papers proposed automatic word clustering methods and reported some improvements [1] [3] [4].

2.3. Variable memory length Markov model

Another extension of an n-gram model is a variable memory length Markov model [2]. In this model, based on a probabilistic suffix tree (PST), the context length, or the value of n, varies depending on the effectiveness of the context.

A PST T, over an alphabet Σ, is a tree of degree $|\Sigma|$. Each edge in the tree is labeled by a single symbol in Σ, such that from every internal node there is exactly one edge labeled by each symbol. The nodes of the tree are labeled by pairs (s, γ_s) where s is the string associated with the path from that node to the root of the tree, and $\gamma_s : \Sigma \rightarrow [0, 1]$ is the next symbol probability function related to s. We require that for every string s labeling a node in the tree, $\sum_{\sigma \in \Sigma} \gamma_s(\sigma) = 1$.

A PST T generates strings of infinite length, but we consider the probability distributions induced on finite length prefixes of these strings. The probability that T generates a string $\boldsymbol{r} = r_1 r_2 \cdots r_m$ in Σ^m is

$$P_T(\boldsymbol{r}) = \prod_{i=1}^{m} \gamma_{s^{i-1}}(r_i),$$

where $s^0 = e$, and for $1 \leq j \leq m-1$, s^j is the string labeling the deepest node reached by taking the path corresponding to $r_i r_{i-1} \cdots r_1$ from the root of T. For example, using the PST depicted in Figure 1, the probability of generating the string "$abcab$" is $P(a|e)P(b|a)P(c|b)$ $P(a|bc)P(b|bca)$.

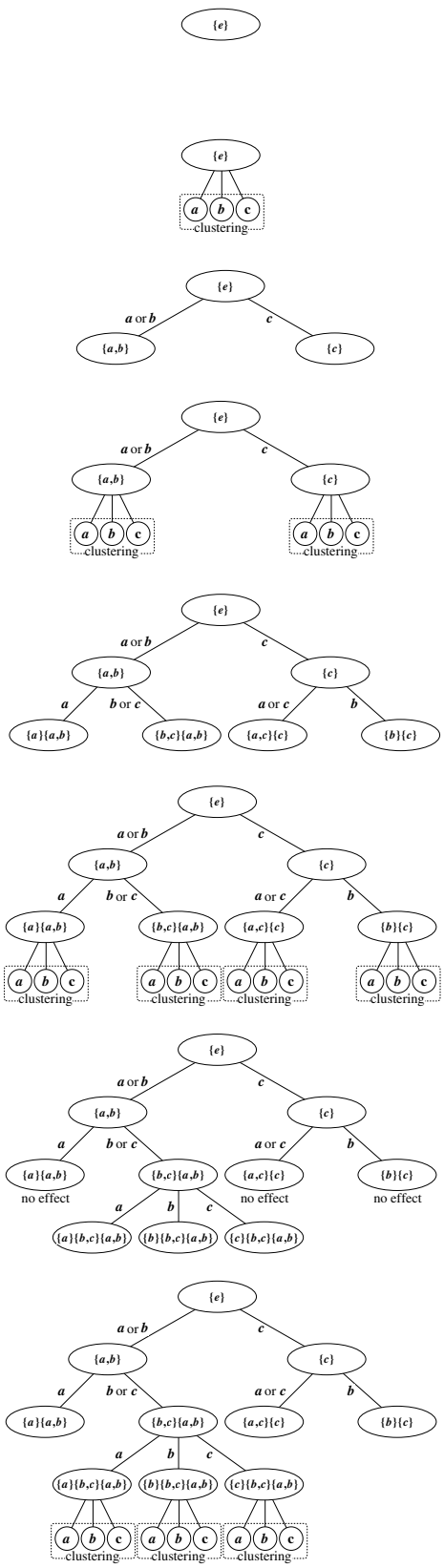
3. Class-based Variable Memory Length Markov Model

In this section we explain our new stochastic language model. This model is an extension of a variable memory length Markov model and includes it as a special case.

3.1. Incorporating word clustering into a probabilistic suffix tree

The notion of class is important because it reduces the size of an n-gram model and sometimes augments its predictive power. Therefore a variable memory length Markov model based on an appropriate class-word relation may be smaller and have more predictive power than the word-based counterpart.

A class-based version of the variable memory length Markov model is represented by a class-based probabilistic suffix tree (class-based PST). Each edge in the tree is labeled by a class, a *set of symbols* in Σ, such that from every internal node there is exactly one edge corresponding to each symbol representing a class. The nodes of the tree are labeled by pairs (S, γ_S) where S is the set of strings associated with the path from that node to the root of the tree, and $\gamma_S : \Sigma \rightarrow [0, 1]$ is the next symbol probability function related to S. We require that for every string set S labeling a node in the tree, $\sum_{\sigma \in \Sigma} \gamma_S(\sigma) = 1$. Thus this model is able to treat similar contexts, such as "the 1st of," "the 2nd of," ... and "the 31st of" with a single node



The prediction class-based suffix trees created during the run of the algorithm (width first version) are depicted from top to bottom.

Figure 3: An illustrative run of the learning algorithm

expand($node$)

return if (**clustering**($node$) = `false`)
foreach $child$ ($nodes$ corresponding to classes)
  create $child$
  expand($child$)

Figure 4: The recursive process of tree creation

"the {1st, 2nd, ..., 31st} of" and to reduce the difficulty of the data-sparseness problem better than a word-based variable memory length Markov model. As a result we can expect a better language model than a word-based variable memory length Markov model.

The probability that a class-based PST T generates a string $\boldsymbol{r} = r_1 r_2 \cdots r_m$ in Σ^m is

$$P_T(\boldsymbol{r}) = \prod_{i=1}^{m} \gamma_{S^{i-1}}(r_i),$$

where $S^0 = \{e\}$, and for $1 \leq j \leq m-1$, S^j is the string set labeling the deepest node reached by taking the path corresponding to $r_i r_{i-1} \cdots r_1$ starting at the root of T. For example, using the PST depicted in Figure 2, the probability of generating the string "$abcab$" is $P(a|e)$ $P(b|\{a,b\})P(c|\{a,b\}\{a\})P(a|\{b\}\{c\})P(b|\{b\}\{b,c\}\{a,b\})$ Note that the next word is predicted directly from the history, not through a class as in a class-based n-gram model.

3.2. Learning Algorithm

The creation of a class-based PST is done as follows. In the beginning the tree has only a single node (root) labeled by the set of an empty string $\{e\}$. During the execution of the algorithm, nodes are added to the tree recursively as shown in Figure 3. The algorithm consists of two processes:

clustering($node$) which calculates the best word-class relation for the preceding symbols of the contexts given as the label of $node$, and which returns the best word-class relation if it seems to be effective or `false` if it failed to find any effective word-class relation. The clustering algorithm and the criterion are the same as those proposed by the paper [4].

Table 1: EDR Corpus (Japanese).

	#sent.	#words	#chars
learning	46,755	1,149,827	1,815,326
test	20,780	509,261	802,576

Table 2: WSJ Corpus (English).

	#sent.	#words	#chars
learning	44,288	1,056,631	4,715,227
test	4,920	117,135	523,455

expand($node$) which calls the clustering process and tries to expan each $node$ recursively to create nodes corresponding to classes returned from the clustering process. Figure 4 shows the flow of the process.

4. Evaluation

We conducted experiments on EDR corpus [6] in Japanese and Wall Street Journal corpus in English. Table 1 and 2 show the size of the corpus. Each word in the corpus is annotated with a part-of-speech (POS) tag. Each learning corpus is divided into nine partial corpora for class-based PST estimation, including interpolation coefficients by deleted interpolation technique. The vocabulary of each model contains 26,792 word-POS pairs in Japanese and 27,377 word-POS pairs in English.

In order to measure the performances of the models, we calculated test set perplexity and the number of non zero parameters. We also built morphological analyzers (POS tagger) [7] for Japanese and mesured their tagging accuracies. The longest history in the class-based variable memory length Markov model consists of five words (6-gram) in both languages.

Table 3 and 4 show the results of the experiments[1]. According to these results, in Japanese the class-based variable memory length Markov model is the best model of all those tested here in perplexity, morphological analysis accuracy, and in model size (number of non zero paremeters). In English, the class-based variable memory length Markov model is as good as the word-based variable memory length Markov model and the word-based tri-gram model in peplexity. In both languages the class-based variable memory length Markov model is as large as the word-based bi-gram model, but is much better than in perplexity. Comparing the class-based and word-based variable memory length Markov model, it can be said that the notion of class improves the word-based variable memory length Markov model in model size. It follows that the model presented in this paper is very useful, especially when we need a small language model without any decrease of accuracy.

[1] Since word clustering for tri-gram model is computationally expensive, we didn't build a class-based tri-gram model.

Table 3: Result on EDR corpus.

language model	class v-gram	word v-gram	class 2-gram	word 2-gram	word 3-gram
perplexity	89.4	91.0	96.7	102.9	89.6
tagger precision	92.2%	91.8%	92.2%	91.8%	91.8%
tagger recall	92.6%	92.5%	92.3%	92.4%	92.6%
#all parameters	139M	2.19G	52.1M	719M	19.3T
#non zero param.	254K	412K	79.4K	257K	606K

$K = 1000$, $M = 1000^2$, $G = 1000^3$, $T = 1000^4$

Table 4: Result on WSJ corpus.

language model	class v-gram	word v-gram	class 2-gram	word 2-gram	word 3-gram
perplexity	209	202	241	252	208
#all parameters	577M	1.88T	212M	752M	20.6T
#non zero param.	423K	563K	182K	349K	727K

$K = 1000$, $M = 1000^2$, $G = 1000^3$, $T = 1000^4$

5. Conclusion

In this paper we have presented a class-based variable memory length Markov model and a stochastic language model based on it. As a result of the experiments. The reductions in size are considerable, but the reductions in perplexity are rather moderate like other improvement for n-gram models. It follows that the model presented in this paper is very useful, especially when we need a small language model without any decrease of accuracy.

6. References

[1] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[2] Dana Ron, Yoram Singer, and Naftali Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Machine Learning*, vol. 25, pp. 117–149, 1996.

[3] Hermann Ney, Ute Essen, and Reinhard Kneser, "On structuring probabilistic dependences in stochastic language modeling," *Computer Speech and Language*, vol. 8, pp. 1–38, 1994.

[4] Shinsuke Mori, Masafumi Nishimura, and Nobuyasu Itoh, "Word clustering for a word bi-gram model," in *ICSLP*, 1998.

[5] Manhung Siu and Mari Ostendorf, "Variable n-gram language modeling and extensions for conversational speech," in *Proc. of the EuroSpeech1997*, 1997, pp. 2739–2742.

[6] Japan Electronic Dictionary Research Institute, Ltd., *EDR Electronic Dictionary Technical Guide*, 1993.

[7] Masaaki Nagata, "A stochastic Japanese morphological analyzer using a forward-DP backward-A* n-best search algorithm," in *Proc. of the COLING94*, 1994, pp. 201–207.