WORD CLUSTERING FOR A WORD BI-GRAM MODEL

Shinsuke Mori *Masafumi Nishimura* *Nobuyasu Itoh*

Tokyo Research Laboratory, IBM Japan
1623-14, Shimotsuruma, Yamato-shi, Kanagawa-ken,
{mori,nisimura,iton}@trl.ibm.co.jp

ABSTRACT

In this paper we describe a word clustering method for class-based n-gram model. The measurement for clustering is the entropy on a corpus different from the corpus for n-gram model estimation. The search method is based on the greedy algorithm. We applied this method to a Japanese EDR corpus and English Penn Treebank corpus. The perplexities of word-based n-gram model on EDR corpus and Penn Treebank are 153.1 and 203.5 respectively. And Those of class-based n-gram model, estimated through our method, are 146.4 and 136.0 respectively. The result tells us that our clustering methods is better than the Brown's method and the Ney's method called leaving-one-out.

1. INTRODUCTION

In statistic method of natural language processing, such as speech recognition, the n-gram model [8] based on word is popular for its facility of parameter estimation and implementation. In this model, however, the number of parameters is equal to the number of vocabulary at the n-th power so that it is not possible to estimate the parameters accurately from a limited corpus which is available currently. 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, the class-base n-gram model [1] is proposed. In this model, each word belongs to a group of words, called class, and is predicted through the statistics on the class sequences, which is more reliable than that of word sequences. In addition, this method decreases the memory size for the model description.

The main problem in the class-based n-gram model is to find the optimum word-class relation for word sequence prediction, which is called word clustering. There are some attempt at English word clustering [1] [7]. The objective function of word clustering proposed in these papers is the perplexity on the very same corpus as for the probability estimation. With this criterion the optimum solution is that each word belongs to a different class. The reason is that for any pair of words merging them forces the model to lose the information. Then they give the final number of class and merge the pair of words with the minimum information loss. The main problem of this method is that the criterion used in the clustering process differs from the criterion of language model, test set perplexity, and we can not expect to obtain a class-based n-gram model with lower perplexity than the word-based n-gram model. In fact, the experimental result reported in the paper [1] shows that the obtained class-based n-gram model has higher perplexity than the word-based n-gram model.

In this paper, we propose a new method which is expected to produce a class-based n-gram model of more accurate prediction than the word-based n-gram model. The main idea is to imitate the test corpus extending the deleted interpolation technique [3], i.e. first we divide the learning corpus into k partial corpora, second we build k language models from the learning corpus reserving the i-th partial corpus for each $i \in \{1, 2, \cdots, k\}$ and finally we evaluate a word-class relation by the average perplexity of all the possible pairs of language model and reserved corpus. With this criterion, a word-class relation similar to the optimum for the test corpus is expected to be calculated without referring the test corpus.

The idea of reserving a part of the learning corpus for word clustering has already been proposed in the paper [5]. The method in this paper, called leaving-one-out, assumes the language model to be in some special form in order to reduce the computational time. As a result, the criterion of word clustering is an approximation of the idea. The result for a German corpus, reported in this paper, shows that the bi-gram model of part-of-speech (POS) given by a grammarian is better than the word-base bi-gram model and the class-based bi-gram model obtained by the proposed method. If the method was really effective, the clustering taking the POS-based bi-gram model as the initial model would produce a better model. In the paper, however, this sort of experiment is not reported. The result for a English corpus, the test set perplexity of the class-based bi-gram model obtained by the proposed method is almost as high as that of the class-based bi-gram model by the preceding method. These results tells us that the idea does not work better than it is expected to do, putting it another way, the approximations of the criterion is harmful.

Contrary to this research, we propose to extend the idea of the deleted interpolation directly and report the results on a Japanese corpus (EDR corpus [2]) and English corpus (Wall Street Journal [6]). The perplexities of word-based n-gram model on EDR corpus and Penn Treebank are 153.1 and 203.5 respectively. And Those of class-based n-gram model, estimated through our method, are 146.4 and 136.0 respectively. The result tells us that our clus-

tering methods is better than the Brown's method and the Ney's method called leaving-one-out.

2. CLASS-BASED N-GRAM MODEL

The class-based n-gram model [1] predicts a word sequence $w = w_1 w_2 \cdots w_k$ by the following formula.

$$p(w) = \prod_{i=1}^{k} p(c_i|c_{i-n+1}\, c_{i-n+2} \cdots c_{i-1}) p(w_i|c_i),$$

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. The probabilities in this formula $p(c_i|c_{i-n+1}\, c_{i-n+2} \cdots c_{i-1})$ and $p(w_i|c_i)$ are estimated from a corpus as follows:

$$p(c_i|c_{i-n+1}\, c_{i-n+2} \cdots c_{i-1}) = \frac{N(c_{i-n+1}\, c_{i-n+2} \cdots c_i)}{N(c_{i-n+1}\, c_{i-n+2} \cdots c_{i-1})}$$
$$p(w_i|c_i) = \frac{N(w_i, c_i)}{N(c_i)},$$

where $N(x)$ represents the number of the event x occurred in the learning corpus.

To cope with the data sparseness problem, the interpolation technique [4] is also applicable to the class-based n-gram model. In our case, this means to mix the class-based n-gram model with other n-gram models of lower n as follows:

- word-based n-gram model

$$p'(w_i|w_{i-n+1}\, w_{i-n+2} \cdots w_{i-1}) = \sum_{j=1}^{n} \lambda_j^w p(w_i|w_{i-j+1}\, w_{i-j+2} \cdots w_{i-1}) \quad (1)$$

$$\text{where } 0 \le \lambda_j^w \le 1 \text{ and } \sum_{j=1}^{n} \lambda_j^w = 1$$

- class-based n-gram model

$$p'(c_i|c_{i-n+1}\, c_{i-n+2} \cdots c_{i-1}) = \sum_{j=0}^{k} \lambda_j^c p(c_i|c_{i-j+1}\, c_{i-j+2} \cdots c_{i-1}) \quad (2)$$

$$\text{where } 0 \le \lambda_j^c \le 1 \text{ and } \sum_{j=1}^{n} \lambda_j^c = 1$$

The value of the coefficients λ is determined using a corpus different from that for the n-gram probability estimation (held-out data). By this method, we must reserve a portion of the limited learning corpus and estimate the probabilities from a smaller corpus. As a result the estimation may be less reliable. To avoid this disadvantage, a sophisticated method, called deleted interpolation [4], has been proposed. In this method the learning corpus is divided into k partial corpora. For all possible combinations, we use $k-1$ corpora for the probability estimation and the rest for the λ value decision and let the final λ value as the average of the λ for each combination and count n-gram frequency again on the entire learning corpus.

3. WORD CLUSTERING

In this section, first we define the relation between word and class. Next, we explain the criterion function for clustering. And last, we describe the search method of word-class relation.

3.1. Relation between word and class

As described above, a class is a set of words. In addition we assume that each word belongs to only one class. Then a word-class relation is a function $f : W \mapsto C$, where W, C are the word set and the class set respectively. Putting it another way, the collection of the classes forms a partition of the set W. Now, we define the operation $move(f, w, c)$ as follows:

> move the word w to the class c and return the resulting word-class relation.

3.2. The objective function

The aim of the word clustering is to build a language model with less test set perplexity without referring the test corpus. As for estimation of interpolation coefficient, the deleted interpolation method is known as the best method. Accordingly we propose the following value, which we call the average test set perplexity, as the criterion of the word clustering

$$\overline{PP} = \left\{ \prod_{i=1}^{k} PP(M_i, \mathcal{C}_i) \right\}^{\frac{1}{k}}, \quad (3)$$

where M_i represents the n-gram model estimated from the partial corpora except for the i-th corpus and $\mathcal{C}_i$ represents the i-th corpus. In our case the corpora are constant and n-gram models depend only on the word-class relation. It follows that the average test set perplexity can be considered as a function of the word-class relation. From the definition, the less the value is, the better the language model is. Now the aim of the clustering is to find the word-class relation which minimizes the average test set perplexity defined by the formula (3).

There are some attempts at word clustering [1] [7]. Those researches propose the perplexity as the objective function. Nevertheless it is calculated on the same corpus as used for the model estimation. And they reported that the resulting class-based n-gram model was worse than the word-based n-gram model. Our method differs from those researches in that we use the cross validation technique, which is shown to be effective experimentally.

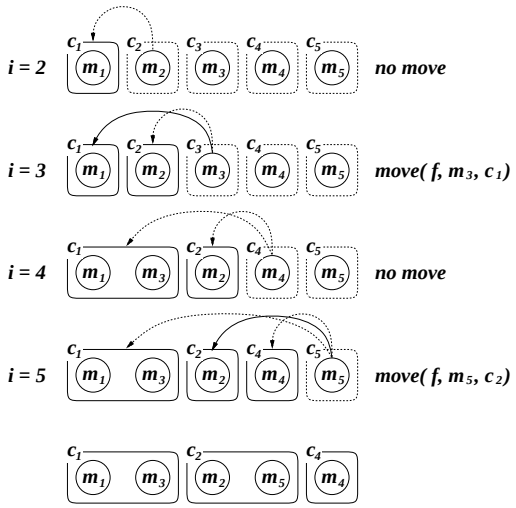

Figure 1: Clustering algorithm.

3.3. Algorithm

The solution space of the word clustering is the set of all possible word-class relations. The cardinality of the set is, however, too enormous for normal word n-gram models to calculate the average test set perplexity for all word-class relations and select the best one. So we abandoned the best solution and adopted a greedy algorithm as follows (see figure 1).

Let $(w_1, w_2, \ldots, w_n)$ as a list of the words sorted in the descending order of frequency.
foreach i $(1, 2, \cdots, n)$
  $c_i := \{w_i\}$
  $f(w_i) := c_i$
foreach i $(2, 3, \cdots, n)$
  $c := \mathbf{argmin}_{c \in \{c_1, c_2, \ldots, c_{i-1}\}} \overline{PP}(move(f, w_i, c))$
  if $(\overline{PP}(move(f, w_i, c)) < \overline{PP}(f))$ **then**
    $f := move(f, w_i, c)$

The reason why the algorithm tries to move words with higher frequency is that their move has more effect on the average test set perplexity and it may be better to move them at the early stage for faster convergence.

4. EXPERIMENTAL RESULT

We built a word-based n-gram model, a class-based n-gram model and a POS-based n-gram model from the same learning corpus and calculated the perplexity on the same test corpus. In this section, we present the results and evaluate our method.

4.1. Conditions

We used EDR corpus [2] and Wall Street Journal [6]. Table 1 is the size of the corpora. First we divide each corpus into ten parts: nine are used as learning corpus and one as test corpus. The models are built as follows.

Table 1: Corpus.

Japanese (EDR corpus)

Corpus	#sentences	#words
learning corpus	187,022	4,595,786
test corpus	20,780	509,261

English (Wall Street Journal)

Corpus	#sentences	#words
learning corpus	44,288	1,056,631
test corpus	4,920	117,135

- word bi-gram model
  1. Estimate λ_1 and λ_2 in the formula (1) using the deleted interpolation method
  2. Count frequencies of wore bi-gram and word unigram on the entire learning corpus.
- POS bi-gram model
  1. Estimate λ_1 and λ_2 in the formula (2) using the deleted interpolation method
  2. Count frequencies of POS bi-gram and POS unigram on the entire learning corpus.
- class bi-gram model
  1. Estimate λ_1 and λ_2 in the formula (1) using the deleted interpolation method
  2. Estimate word-class relation by the clustering algorithm
  3. Estimate λ_1 and λ_2 in the formula (2) using the deleted interpolation method
  4. Count frequencies of class bi-gram and class unigram on the entire learning corpus.

Unknown words are predicted by a unknown word model based on the character bi-gram model. This part is common among the above three models.

4.2. Results and Discussion

Table 2 shows the perplexities of each model. The perplexity of the class-base bi-gram model obtained by our word clustering method is lower than that of word-based bi-gram model and POS-based bi-gram model. It follows that our method is better than those proposed in [1] and [7], in which they report that the perplexity of the obtained class-based n-gram model is slightly higher that the word-based n-gram model. There is another method [5], called *leaving-one-out method*, based on the same idea as ours. Contrary to ours, the result reported in this paper is not promising: for a German corpus the obtained class-based bi-gram model has higher test set perplexity than the POS-based bi-gram model and for a English corpus the obtained class-based bi-gram model has almost the same perplexity on the test corpus as the class-based bi-gram model proposed antecedently [1]. The main reason is that the leaving-one-out method adopts some approximations in order for diminution of the computational cost, e.g. it

Table 2: An experimental result.

Japanese (EDR corpus)

language model	#states	perplexity
word-based bi-gram model	59,972	153.1
POS-based bi-gram model	31	392.4
class-based bi-gram model	5,990	146.4

English (Wall Street Journal)

language model	#states	perplexity
word-based bi-gram model	27,423	203.5
POS-based bi-gram model	91	556.1
class-based bi-gram model	4,651	136.0

uses the class-based bi-gram model which is *not* interpolated with uni-gram model during word clustering.

The number of states of the obtained class-based model is 10.0% less than that of word-based model. This means that the class-based model is also superior to the word-based model in terms of the memory space. The size of transit probability table of the class-based bi-gram model used in the experiment is 0.998% smaller than the word-based bi-gram model if the array is adopted for the implementation of the model because the size is proportional to the number of states at the second power (bi-gram model). The number of the non zero elements in the word-based bi-gram model is 724,870 and that of the class-based bi-gram model is 245,283. It follows that the reduction rate of 33.8% is achieved if hash or linked list is used for implementation.

5. CONCLUSION

We have described a new word clustering method to ameliorate a word-based n-gram model. Extending the idea of the deleted interpolation, our method have succeeded in building a class-based n-gram model with lower test set perplexity both in Japanese and in English.

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. Japan Electronic Dictionary Research Institute, Ltd. *EDR Electronic Dictionary Technical Guide*, 1993.
3. F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceeding of the Workshop on Pattern Recognition in Practice*, pp. 381–397, 1980.
4. Fredelick Jelinek, Robert L. Mercer, and Salim Roukos. Principles of lexical language modeling for speech recognition. In *Advances in Speech Signal Processing*, chapter 21, pp. 651–699. Dekker, 1991.
5. R. Kneser and H. Ney. Improved clustering techniques for class-based statistical language modelling. In *Eurospeech*, pp. 21–23, 1993.
6. Mitchell P. Marcus and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.
7. 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.
8. C. E. Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, Vol. 30, pp. 50–64, 1951.