Language Model Adaptation Using Word Clustering

Shinsuke MORI, Masafumi NISHIMURA, Nobuyasu ITOH

IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma Yamatoshi, 242-8502, Japan
mori@trl.ibm.com

Abstract

Building a stochastic language model (LM) for speech recognition requires a large corpus of target tasks. For some tasks no enough large corpus is available and this is an obstacle to achieving high recognition accuracy. In this paper, we propose a method for building an LM with a higher prediction power using large corpora from different tasks rather than an LM estimated from a small corpus for a specific target task. In our experiment, we used transcriptions of air university lectures and articles from *Nikkei* newspaper and compared an existing interpolation-based method and our new method. The results show that our new method reduces perplexity by 9.71%.

1. Introduction

A stochastic language model (LM) used in speech recognition systems, etc., requires a large set of sentences in the target task field. In many real applications, however, there is no sufficiently large corpus for a practical LM. This causes a decrease in recognition accuracy.

As a solution to this problem, the use of interpolation has been proposed [1, 2, 3]. In this approach, a language model built from a small set of sentences in a target task field (a task corpus) is interpolated with a language model built from a large set of sentences in general fields (a general corpus), such as newspapers, journals, and so forth.

A serious weak point of this method is that the interpolated model is not able to refer to the contexts in the task field for words belonging to the task field which do not by chance appear in the available small task corpus. In this paper, we propose a method which enables the model to predict many words in the task field appearing only in the general corpus by refering to the contexts of similar words in the task field. This is realized by a word clustering which assigns words in the general corpus to a class represented by a similar word appearing in the task corpus.

In the experimental evaluation, we compared the prediction power of our new method with that of a method based on interpolation. We used a set of transcriptions of broadcast lectures as a task field corpus and a set of newspaper articles as a general field corpus. As a result of experiments, the perplexity of the model based on our method is approxmately 10% lower than the interpolation-based approach. This shows that our method is efficient for the task adaptation problem.

2. Language Model

The task adaptation method we propose in this paper is applicable to all language models which regard a sentence as a sequence of certain units. In this paper, we explain an application of our method to a word-based n-gram model and show some experimental results.

2.1. Word n-gram Model

A word n-gram model regards a sentence as a sequence of words ($\boldsymbol{w} = w_1 w_2 \cdots w_h$) and predicts each word from the beginning to the end referring to the last $k = n-1$ words. For a simplicity, we assume that there are sufficiently large number of boundary tokens (BT) before the first word and that there is a boundary token indicating the sentence boundary. Since it is hard to enumerate all words, the model has to be able to handle unknown words. To solve this problem, the model has a special token for unknown words (UW) and each word outside of the known vocabulary set is predicted from this token by using an unknown word model, which we explain below.

The probability of a sentence $w_1 w_2 \cdots w_h$ given by a word n-gram model M_w is represented by the following formula.

$$\begin{aligned} & M_w(w_1 w_2 \cdots w_h) \\ & \quad = \prod_{i=1}^{h+1} P_w(w_i | w_{i-k} \cdots w_{i-2} w_{i-1}), \\ & P_w(w_i | w_{i-k} \cdots w_{i-2} w_{i-1}) \\ & \quad = \begin{cases} P(w_i | w_{i-k} \cdots w_{i-2} w_{i-1}) & \text{if } w_i \in \mathcal{W}_k \\ P(\mathtt{UW} | w_{i-k} \cdots w_{i-2} w_{i-1}) M_x(w_i) & \\ & \text{if } w_i \notin \mathcal{W}_k \end{cases}, \end{aligned}$$

where $\mathcal{W}_k$ represents the vocabulary set and M_x represents an unknown word model which regards an unknown word as a sequence of characters and predicts them from the beginning to the end as follows:

$$M_x(x_1 x_2 \cdots x_h) \quad = \quad \prod_{i=1}^{h+1} P_x(x_i | x_{i-k} \cdots x_{i-2} x_{i-1})$$

.

2.2. Interpolation

In general, the parameters are estimated based on maximum likelihood estimation. This method, however, suffers from an inaccuracy of estimation when the frequency is too low. To cope with this problem, an interpolation technique is used [4]. In this method, the n-gram model is mixed with more reliable n-gram models of low n as follows:

$$\begin{aligned} & P(w_i | w_{i-k} w_{i-k+1} \cdots w_{i-1}) \\ & \quad = \sum_{j=0}^{k} \lambda_j P(w_i | w_{i-j} w_{i-j+1} \cdots w_{i-1}) \\ & \quad\quad \text{where } 0 \le \lambda_j \le 1, \ \sum_{j=0}^{k} \lambda_j = 1. \end{aligned}$$

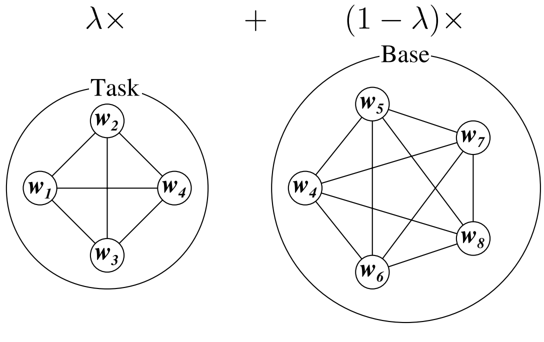


Figure 1: Interpolation-based task adaptation.

The interpolation coefficients λ are estimated by the deleted interpolation method [4].

3. Task Adaptation

In this section, we propose a task adaptation method which increases the predictive power of a language model in a task field for which a large corpus is not available. First we explain an existing interpolation-based method and how our clustering-based method works for an application to a word n-gram model.

3.1. Task Adaptation based on interpolation

The most popular task adaptation method is based on the interpolation technique. Using this method a language model estimated from a general corpus is adapted to a task field by interpolating with a language model estimated from a task corpus. For models which regard a sentence as a sequence of words and which predict words from the beginning to the end, the task adaptation model $P_a(w_i|w_1w_2\cdots w_{i-1})$ is represented as follows, where $P_t(w_i|w_1w_2\cdots w_{i-1})$ represents the model estimated from the task corpus and $P_b(w_i|w_1w_2\cdots w_{i-1})$ represents the model estimated from the general corpus (cf. Figure 1):

$$\begin{aligned}&P_a(w_i|w_1w_2\cdots w_{i-1})\\ &\quad= \lambda P_t(w_i|w_1w_2\cdots w_{i-1})\\ &\qquad+(1-\lambda)P_b(w_i|w_1w_2\cdots w_{i-1}), \quad \text{where } 0\le\lambda\le 1\end{aligned}$$

The interpolation coefficients λ are estimated by the deleted interpolation method maximizing the likelihood of the task corpus.

3.2. Task Adaptation based on word clustering

When only a small task field corpus is available, many common words (uni-gram) and many common word sequences (n-gram, $n \ge 2$) belonging to the task field will not appear in the corpus. Some of them may appear in the test corpus and cause a decrease of the predictive power of an LM. The objective of task adaptation is to reduce the number of these words and word sequences to improve the predictive power. For predicting the words belonging to the task field but not appearing in an available task field corpus the interpolation-based model cannot use the word sequence information (n-gram, $n \ge 2$) in the task field corpus.

The basic idea of our task adaptation method is that the model should predict words appearing only in the general corpus by referring to the context information for similar words in the task field corpus. This is done in the following way (see Figure 2 and Figure 3):

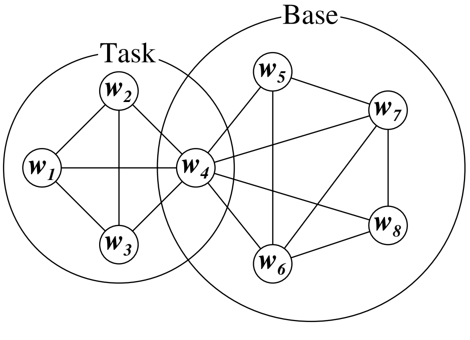


Figure 2: Model built from the simple summation corpus.

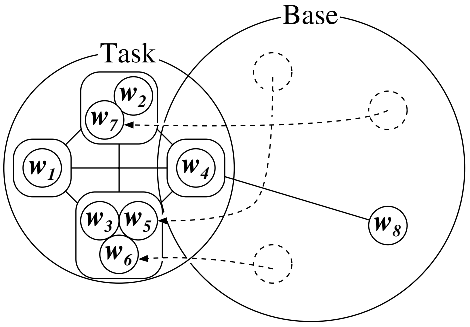


Figure 3: Clustering-based task adaptation.

1. Build a word bi-gram model from the summation of the task field corpus and the general field corpus (Figure 2)
2. Regard the resulting model as a class bi-gram model (one word corresponds to one class)
3. Cluster the words in the general corpus into a similar word appearing in the task field corpus (if one exists)
4. When no similar word is found, the word in the general corpus represents a class by itself
5. Count class uni-grams and bi-grams using the word-class map acquired by the above clustering algorithm
6. Perform an interpolation among the class uni-gram and the class bi-gram in the task corpus and the class uni-gram and the class bi-gram in the general corpus
7. Calculate the word probabilities from each class (see below)

The model built by the above method is a class bi-gram model. Therefore the model is represented as a product of a probability of class prediction from class sequence and a probability of word prediction from the predicted class.

$$P(\boldsymbol{w}) = \prod_{i=1}^{n} P(w_i|c_i)P(c_i|c_{i-1})$$

Below, we explain how to calculate the two conditional probabilities in the above formula.

As the following formula shows, the probability of word prediction from the predicted class is divided into two cases depending on the number of words belonging to the class (one or more than one). When the class contains more than one word, the prediction is divided into two more cases: 1) predicting words in the task corpus and 2) predicting words in the general corpus. In the following formula, $\mathcal{W}_t$ denotes the vocabulary from the task corpus, $\mathcal{C}_t$ the class set and α the probability that the words in the task corpus are generated from the class.

$$P(w_i|c_i) = \begin{cases} \alpha & \text{if } c_i \in \mathcal{C}_t, w_i \in \mathcal{W}_t \\ (1-\alpha)\frac{f_b(w_i)}{\sum_{w \in c_i} f_b(w)} & \text{if } c_i \in \mathcal{C}_t, w_i \notin \mathcal{W}_t \\ 1 & \text{otherwise.} \end{cases}$$

The value of $1 - \alpha$ in the formula is the summation of the probabilities that the words in the general corpus are generated from the class. And the summation is distributed to the words in proportion with the frequency of the words in the general corpus. The value of α is determined as follows:

$$\alpha = \frac{\sum_{c_i \in \mathcal{Y}} f_t(y(c_i))}{\sum_{c_i \in \mathcal{Y}} \sum_{w \in c_i} f_t(w)},$$

where $\mathcal{Y} \subseteq \mathcal{C}_t$ represents the set of classes containing words in the general corpus, $y(c)$ the word in the task corpus of the class c, and f_t the frequency in the task corpus [1].

The probability of class prediction from the class sequence is defined as the interpolation of the class uni-gram model, the class bi-gram model in the task corpus, the class uni-gram model, and the class bi-gram model in the general corpus as follows:

$$P(c_i|c_{i-1}) = \begin{cases} \lambda_1 P_t(c_i) + \lambda_2 P_s(c_i) + \lambda_3 P_t(c_i|c_{i-1}) + \lambda_4 P_s(c_i|c_{i-1}) \\ \hfill \text{if } f_t(c_{i-1}) > 0 \\ \lambda_5 P_t(c_i) + \lambda_6 P_s(c_i) + \lambda_7 P_s(c_i|c_{i-1}) \;\; \text{otherwise} \end{cases}$$

In this formula P_t is the probability estimated from the task corpus and P_s is the probability estimated from the general corpus and the task corpus.

3.3. Word Clustering For Task Adaptation

As already described, the central idea of our method is that the model can refer to the context information for similar words in the task corpus as the context information for words in the general corpus. To implement this idea, we use a word clustering in which the words in the task corpus are cluster centers and the words appearing only in the general corpus are merged into a cluster center word having similar behavior. The best class is searched for by using an existing word clustering method [5, 6, 7, 8].

In the experiments, we used a bottom up clustering method with the cross entropy as the similarity measure [7]. In this method the algorithm calculates the effects of all possible merges of a word into classes in the descending order of frequency of the target words. The effectiveness is measured by the cross entropy on the summation of the general corpus and the task corpus. The word is merged into the class causing the largest decrease of the cross entropy. If all of the possible merges increase the cross entropy, the word is not merged and represents a class.

Table 1: Corpus.

	#sentences	#words	#chars
learning	7,677	218,628	336,726
test	853	24,268	37,552

4. Evaluation

We conducted experiments in order to compare the language model adaptation method explained in section 3 and some other existing methods. In this section, we describe the conditions and the results of the experiments and evaluate our new method.

4.1. Conditions

We used a set of transcriptions of broadcast lectures in Japanese as a task field corpus and a set of newspaper articles (*Nikkei* Economic Newspaper) as a general field corpus. The task field corpus was divided into 10 parts, one for testing and nine for parameter estimation. The general field corpus was divided into nine parts for parameter estimation. Nine parts of each training corpus are added to make nine simple summation corpora. We call the set of these corpora a simple summation corpus. The vocabulary of the models in the experiments is the set of words appearing in more than one part.

4.2. Detail of the models

In the experiments we compared the predictive powers of the bi-gram models estimated by the following task adaptation methods.

- simple frequency summation

  A word-based bi-gram model estimated from the simple summation corpus (see Figure 2)

- interpolation-based task adaptation (an existing method, see Figure 1)

  An interpolation of a word-based bi-gram model estimated from the task field corpus and a word-based bi-gram model estimated from the general field corpus.

- clustering-based task adaptation (our method, see Figure 3)

  A class-based bi-gram model built from a word-based bi-gram estimated from the simple summation corpus by moving the words in the general corpus [2] to appropriate classes represented by a word in the task field corpus.

The task adaptation models have the same unknown word model as the simple summation model. Since the models have the same vocabulary, the contributions of the unknown word models to the entropy is constant.

4.3. Evaluation

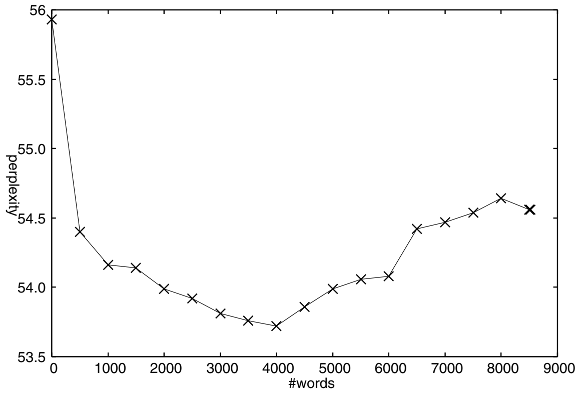
In order to evaluate the predictive power of the models, we calculated character-based entropy and word-based perplexity of the test corpus extracted from task field corpus (Table 2). As a result,

[1] $f_t(w) \geq 1$ for some words in the general corpus.

[2] words in the vocabulary appearing in less than two parts of the nine task field corpora

Table 2: Predictive powers of the models (character-based entropy).

adaptation method	word prediction	character prediction of unknown words	summation	word-based perplexity
simple summation	3.865	0.511	4.376	87.11
interpolation	3.492	0.511	4.003	59.52
word clustering	3.391	0.511	3.902	53.72



The perplexity (55.93) of zero word clustering is the result of the interpolation of the uni-gram models and the bi-gram models of the both corpora at the same time.

Figure 4: relationship between the number of clustering words and predictive power.

the perplexity of the model estimated by the interpolation-based task adaptation method is lower than that of the model estimated from the simple summation corpus. This result confirms the known fact that the interpolation-based task adaptation method improves on the predictive power of the simple model. The perplexity of the model estimated by the clustering-based task adaptation method is 9.74% lower than that of the model estimated by the interpolation-based task adaptation method. This shows that our method is superior to the interpolation-based method.

In the above experiments, we clustered 3,779 words appearing more than four times in the descending order of their frequency. We calculated the perplexities, changing the number of words to be clustered (Figure 4). The result shows that the effect of the clustering weakens along with the decrease of frequency, and the clustering has a negative effect around the 4,000th word. This result concurs with the general consensus that the clustering of low frequency words is inaccurate. Another reason is that we have to use the cross entropy on the general corpus as the clustering criterion, instead of that on the task corpus.

5. Conclusion

In this paper we have presented a new method for adapting a language model to a task for which a large corpus is not available. With our method a language model is able to refer to the context information of words appearing only in a general corpus. This is done by clustering general corpus words and merging them into task corpus words. The experimental results show that our method is superior to the interpolation-based method.

6. References

[1] Shoichi Matsunaga, Tomokazu Yamada, and Kiyohiro Shikano. Task adaptation in stochastic language models for continuous speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 165–168, 1992.

[2] Reinhard Kneser and Volker Steinbiss. On the dynamic adaptation of stochastic language models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 586–589, 1993.

[3] P. R. Clarkson and A. J. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 799–802, 1997.

[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, pages 651–699. Dekker, 1991.

[5] 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*, 18(4):467–479, 1992.

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

[7] Shinsuke Mori, Masafumi Nishimura, and Nobuyasu Itoh. Word clustering for a word bi-gram model. In *International Conference on Speech and Language Processing*, 1998.

[8] Jianfeng Gao, Joshua Goodman, Guihong Cao, and Hang Li. Exploring asymmetric clustering for statistical language modeling. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, 2002.