Unsupervised language model adaptation based on automatic text collection from WWW

Motoyuki Suzuki, Yasutomo Kajiura, Akinori Ito and Shozo Makino

Graduate School of Engineering, Tohoku University, Sendai, Japan

{moto, kajiura0619, aito, makino}@makino.ecei.tohoku.ac.jp

Abstract

An *n*-gram trained by a general corpus gives high performance. However, it is well known that a topic-specialized *n*-gram gives higher performance than that of the general *n*-gram. In order to make a topic specialized *n*-gram, several adaptation methods were proposed. These methods use a given corpus corresponding to the target topic, or collect documents related to the topic from a database. If there is neither the given corpus nor the topic-related documents in the database, the general *n*-gram cannot be adapted to the topic-specialized *n*-gram.

In this paper, a new unsupervised adaptation method is proposed. The method collects topic-related documents from the world wide web. Several query terms are extracted from recognized text, and collected web pages given by a search engine are used for adaptation. Experimental results showed the proposed method gave 7.2 points higher word accuracy than that given by the general *n*-gram.

Index Terms: language model adaptation, world wide web, search engine, Google, quary-based sampling

1. Introduction

N-gram is one of the most powerful language models for speech recognition. An *n*-gram trained by a general corpus which includes many kinds of topics gives high performance. However, it is well known that a topic-specialized *n*-gram gives higher performance than the general *n*-gram.

The topic-specialized *n*-gram can be trained by a text corpus that contains sentences related to that topic. It is, however, time consuming to collect a huge amount of documents related to the topic. In order to solve this problem, several adaptation methods for language models were proposed. The basic strategy of these methods is to merge topic-related documents into a general corpus. The typical method of collecting topic-related documents is to select sentences or documents related to that topic from a general corpus using some sort of statistical metric. A small amount of topic-related documents are prepared by human, and similar documents are then selected from the general corpus by calculating a statistical metric such as Kullback-Leibler divergence, perplexity given by the topic-specialized *n*-gram, and so on. If recognized text is used for selection instead of the topic-related documents, this method can be considered as unsupervised training [1].

In these methods, however, the kind of adaptable topics is limited to topics included in the general corpus. If documents related to the topic are not included in the general corpus, the general *n*-gram cannot be adapted to the topic because no topic related document is selected from the general corpus.

There have been several researches [2–6] on the training of topic-specialized *n*-gram using the World Wide Web. In these methods, topic-related web pages are retrieved using a search engine (such as Google) using several keywords (or keyphrases), and useful text data is extracted from the web pages using a text filter. This type of collecting method is called "query-based sampling" [2], and it can make a topic-specialized *n*-gram for any topic because most of the topic is included in the Web. The key point of these methods is how to select a keyword as a query term. The simplest method is to give a keyword manually [3, 4]. Other methods [2, 5] use statistical metrics (such as *ctf* (Collection Term Frequency), *df* (Document Frequency), and so on) calculated from a small amount of documents related to the topic.

The topic-specialized *n*-gram can be acquired by web-based methods. The *n*-gram, however, does not always give high performance because the collected documents may be "dirty," i.e. they may contain fragments of sentences (e.g. entries of a table), short sentences such as indices of articles, or other non-linguistic data. A text filter is applied to collected web pages in order to reduce "dirty" sentences, however, "dirty" sentences may remain in the collected text data.

In this paper, we propose a new unsupervised language model adaptation method based on "query-based sampling". This method can adapt a general *n*-gram to any topic because it is a web-based method, and the proposed method can reduce the adverse influence of "dirty" text data because the adapted *n*-gram is trained using both "dirty" text data and a "clean" general corpus.

2. Overview of the System

Figure 1 shows an overview of the proposed system. The target of the system is dictation of lecture speech, where a speaker is talking on one topic.

At first, an input speech data (which contains many utterances) is recognized using a general *n*-gram trained by a general corpus. Several keywords are selected from the recognized text, and then web pages related to the topic are retrieved using a search engine.

Collected web pages are input to a text filter. The filter removes HTML tags, and extracts natural Japanese sentences using character *n*-gram [7]. The character *n*-gram is trained by a general corpus, and the perplexity of each sentence in the web pages is calculated. Only those sentences with lower perplexity than the threshold are extracted.

After that, a mixed corpus is constructed by merging the general corpus and the extracted sentences, and the adapted *n*-gram is trained by the mixed corpus [8]. Finally, the input speech data is recognized using the adapted *n*-gram.

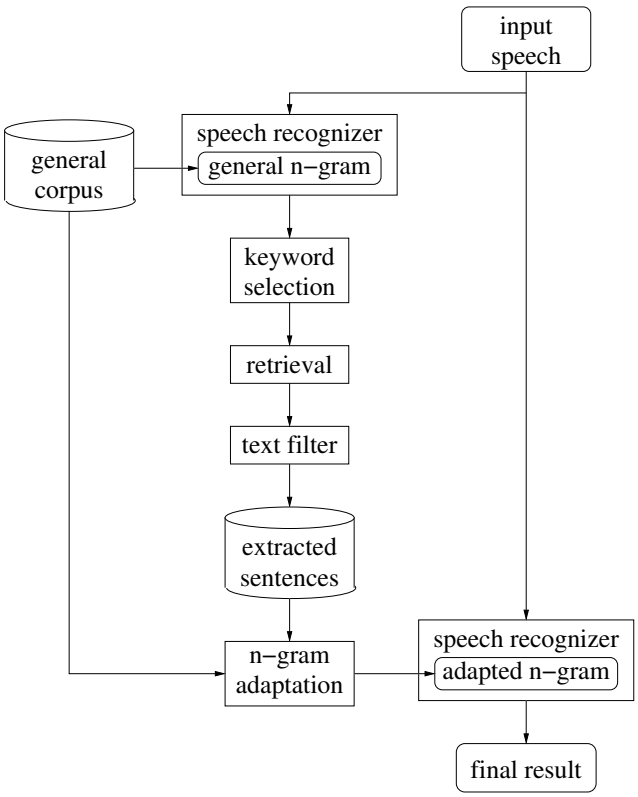

Figure 1: *Overview of the proposed system.*

2.1. Selection of query terms

We have to choose query terms that can specify the target topic. We used *nouns* as candidates of query words. *Tf·idf* is frequently used for selecting keywords in the information retrieval field. *Tf* (Term Frequency) means the frequency of a term in the document, and *idf* (Inverse Document Frequency) means the inverse frequency of documents which contain the term. In this system, *tf* is calculated by the recognized text, and *idf* is calculated by the general corpus.

The *tf* · *idf* score of the term w is calculated by Eq.(1).

$$s(w) = tf(w) \times \log \frac{N}{df(w)} \quad (1)$$

where, $tf(w)$ denotes the number of occurrences of the term w in the *recognized text*, $df(w)$ denotes the number of documents containing the term w in the *general corpus*, and N denotes the number of documents in the general corpus. Notice that the $tf(w)$ and $df(w)$ are calculated based on different document set. While $tf(w)$ is calculated from the recognized text, $df(w)$ is calculated from the general corpus.

The procedure for selecting query terms is as follows:

1. The recognized text are parsed into words through a morphemic analyzer ChaSen [9]. Let $w_1, w_2, \ldots, w_K$ be distinct noun words appear in the recognized text.
2. $s(w_1), s(w_2), \ldots, s(w_K)$ are calculated according th Eq. (1).
3. Let $w_{i_1}, w_{i_2}, \ldots, w_{i_k}$ be the noun words sorted by $s(w)$ in a discending order, i.e., $s(w_{i_1}) \geq s(w_{i_2}) \geq \cdots \geq s(w_{i_K})$. Then the top n words, $w_{i_1}, \ldots, w_{i_n}$, are used as query terms.

Table 1: Details of the evaluation data

ID	title	#words
ID 0	History of characters	1,421
ID 1	Lottery	1,286
ID 2	Pictures in USA and Europe	1,092
ID 3	Effect of charcoal	2,364
ID 4	Accounting business	1,479

2.2. Collection of text from WWW

When a query term is given to a search engine, the search engine returns a list of URLs. Based on the list of URLs, the collection of web pages is performed recursively, on a breadth-first basis. At first, all web pages in the list are collected, and then web pages linked by the top-ranked page are collected. After that, web pages linked by the second-ranked page are collected. This procedure iterates for the lower ranks until the total number of web pages reaches the pre-defined number.

The number of query terms is fixed to one for a retrieval. When two query terms are used, each query term is used for each retrieval, respectively. As a result, two URL lists corresponding to each query term are acquired. Almost all search engines can accept two or more query terms for one retrieval. If two query terms are given to a search engine, web pages which contain both query terms are listed. It is not confirmed which type of query, using two or more query terms for a retrieval, or using each query term separately for each retrieval, is better for this system.

3. Experiments

In order to confirm the effectiveness of the proposed system, several experiments were carried out. The CSJ speech corpus [10, 11] was used for training and evaluation. Transcriptions of 3,124 lectures (containing about 8.3 million words) were used as a general corpus, and 5 other lectures were used for evaluation. Details of the evaluation data are shown in Table 1. In this table, '#words' stands for the total number of words contained in the document.

The vocabulary size of the general *n*-gram was set to 39,863, and all new words in the retrieved sentences were added to the vocabulary of the adapted *n*-gram (however, the maximum number of vocabulary was set to 65,535 because of the limitation of the speech recognizer). The weighting factor in the *n*-gram adaptation method was set to the optimum value *a posteriori*.

Google Japan [12] was used as the search engine, which returns 1,000 URLs at maximum per retrieval. Julius [13] was used as the speech recognizer.

3.1. Results for single query term

In this experiment, only the query term with the highest $s(w)$ was used for retrieval. The total number of retrieved web pages was set to 10,000. Figure 2 shows the word accuracy for each evaluation data set. In this figure, the left side of each data set denotes the result given by the general *n*-gram, and the right side denotes that by the adapted *n*-gram. From this figure, the proposed method

was very effective in increasing word accuracy. The adapted *n*-gram gave higher performance than that of the general *n*-gram for all data sets, and the average word accuracy was increased from 53.9% to 60.2%.

We also investigated how many web pages should be collected. In this experiment, the query term was selected from the transcription of the evaluation data instead of the recognized text in order to investigate clearly the relationship between the number of pages and word accuracy.

Figure 3 shows word accuracy with respect to the number of retrieved web pages from 1,000 to 100,000. In this figure, the horizontal line denotes the word accuracy given by the general *n*-gram (baseline). We also checked the OOV (Out Of Vocabulary) rate. Figure 4 shows the OOV rate, and the horizontal line denotes the OOV rate given by the general *n*-gram.

Word accuracy increased with the number of web pages. The maximum gain compared with the baseline was 7.2 points at 80,000 web pages, however, a 6.3 points gain can be achieved with only 5,000 pages. The OOV rate also decreased with increase in the number of web pages. The minimum OOV rate was 0.75% at 40,000 pages, and the OOV rate at 5,000 pages was 1.19%. Note that the OOV rate at more than 40,000 pages increases because the number of vocabulary reached the maximum number (65,535 words) under these conditions.

These results show that collecting 5,000 pages increased the performance of the adapted *n*-gram. However, collecting more than 10,000 pages does not help much. As a result, about 5,000 pages were sufficient to adapt the *n*-gram to the topic.

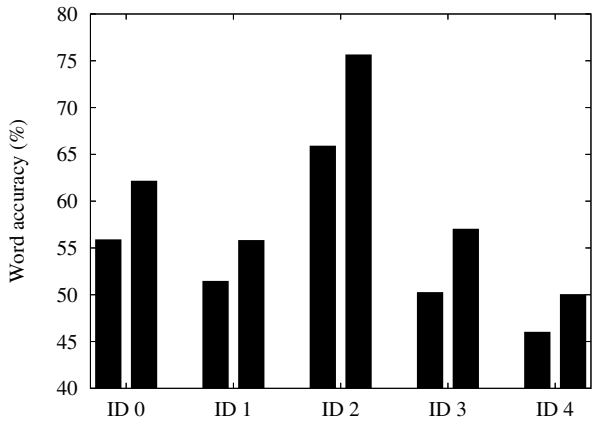

Figure 2: *Word accuracy using a single query term.*

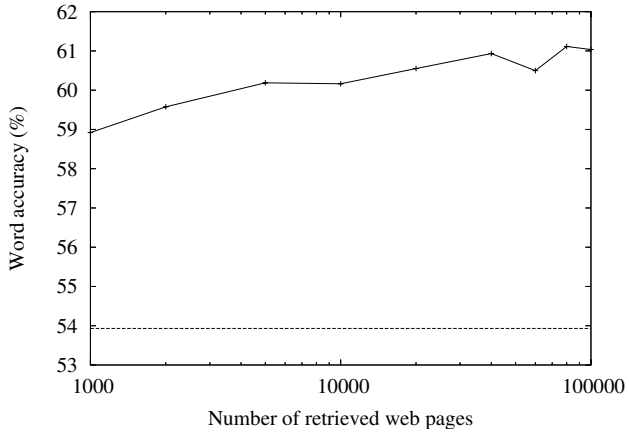

Figure 3: *Relationship between word accuracy and number of pages.*

Table 2: *Relationship between word accuracy and number of query terms.*

#query	transcription	recognized
1	60.6%	60.3%
2	60.4%	60.0%
5	61.1%	59.9%
10	61.7%	60.4%
baseline	53.9%	

3.2. Results for multiple query terms

The previous experiment suggested that the number of retrieved pages should be set to 5,000. If much memory and computational power are available, two or more query terms can be used for collecting additional pages. In this section, we investigate the effectiveness of increasing query terms.

The total number of retrieved web pages was fixed to 20,000, and the number of query terms was increased from one to ten. For example, when the number of query terms was set to five, the number of retrieved pages became 4,000 for each query term.

Table 2 shows the relationship between word accuracy and number of query terms. In this table, "transcription" denotes that query terms were selected from the transcription of the evaluation data, and "recognized" denotes that query terms were selected from the recognized text. "baseline" means the word accuracy given by the general *n*-gram.

In the "transcription" condition, increasing the query terms improved word accuracy, showing that the collecting web pages using several query terms is effective in adaptation of the general *n*-gram.

On the other hand, in the "recognized" condition, word accu-

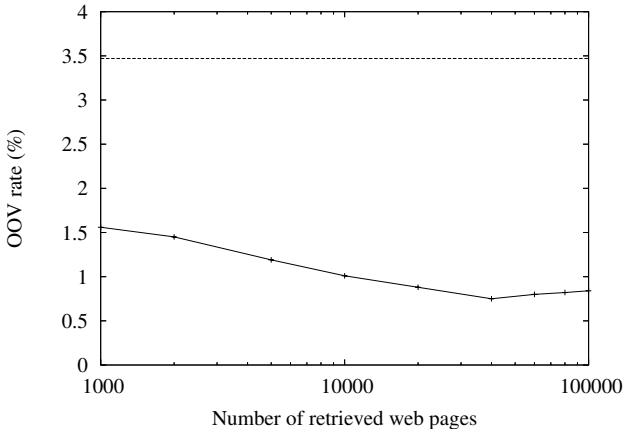

Figure 4: *Relationship between OOV rate and number of pages.*

Table 3: *Rank number of query terms in the "transcription" condition.*

	1	2	3	4	5	6	7	8	9	10
ID 0	2	—	—	7	9	—	38	3	15	8
ID 1	1	2	5	9	19	20	—	25	26	27
ID 2	1	6	3	12	16	21	—	13	5	—
ID 3	3	—	8	9	—	—	—	13	14	27
ID 4	9	10	16	—	—	6	1	14	17	8

Table 4: *Selected query terms of document ID 0.*

rank	transcription	recognized
1	type face	character
2	character	household
3	Gothic type	leaving hospital
4	style	writing brush
5	horizontal line	printing
6	type	good book
7	writing brush	Ming-style type
8	*Anno Domini*	Gothic type
9	printing	familiarity
10	Mr.	*Anno Domini*

racy was not increased even if the number of query terms was increased. The reason is that there were misrecognized words in the query terms. Table 3 shows the relationship between rank numbers in both the "transcription" and "recognized" conditions. In this table, the top row indicates the rank number in the "recognized" condition, and other rows indicate the rank number of the word in the "transcription" condition. For example, the first-ranked word in the "recognized" condition for document ID 0, which is the word "character", was ranked second in the "transcription" condition. "—" denotes the word is not listed in the "transcription" condition.

From this table, the first-ranked word in the "recognized" condition was adequate for adaptation. However, other words were not necessarily adequate. Because there were many misrecognized words in the recognized text, the word accuracy did not improved in the "recognized" condition.

Table 4 shows the selected query terms of document ID 0 (history of characters) in both conditions. While there are appropriate words in the "transcription" condition, the words ranked second and third in the "recognized" condition are not appropriate for this topic. These two words were the result of misrecognition. The word "household" is written "*shotai*" in Japanese, which is a homonym of the word "type face" in Japanese. It is therefore necessary to select the correct words from the recognized text.

4. Conclusions

In this paper, a new unsupervised adaptation method is proposed. The method collects topic-related documents from the web. At first, input speech data are recognized using the general n-gram, and several query terms are extracted from the recognized text using the $tf \cdot idf$ metric. Each query word is given to a search engine, and web pages related to the topic are collected. Finally, the general n-gram is adapted to the target topic using collected and general corpora, and the input speech data is recognized using the adapted n-gram.

From the experiments with a single query term, the proposed method gave 7.2 points higher word accuracy than that given by the general n-gram, and 5,000 pages was sufficient to adapt the n-gram to the topic.

We also carried out the experiments with multiple query terms. It was effective to increase the number of query terms when query terms were extracted from the transcription of the evaluation data. However, it was not effective when query terms were extracted from the recognized text because the recognized text contained many misrecognized words.

5. References

[1] B. Bigi, Y. Huang, and R. D. Mori, "Vocabulary and language model adaptation using information retrieval," in *Proc. ICSLP*, 2004, pp. 602–605.

[2] J. Callan, M. Connell, and A. Du, "Automatic discovery of language models for text databases," in *Proc. ACM SIGMOD International Conference on Management of Data*, 1999, pp. 479–490.

[3] A. Sethy, P. G. Georgiou, and S. Narayanan, "Building topic specific language models from webdata using competitive models," in *Proc. INTERSPEECH*, 2005, pp. 1293–1296.

[4] Y. Ariki, T. Shigemori, T. Kaneko, J. Ogata, and M. Fujimoto, "Live speech recognition in sports games by adaptation of acoustic model and language model," in *Proc. EUROSPEECH*, 2003, pp. 1453–1456.

[5] G. A. Monroe, J. C. French, and A. L. Powell, "Obtaining language models of web collections using query-based sampling techniques," in *Proc. HICSS*, 2002.

[6] A. Berger and R. Miller, "Just-in-time language modelling," in *Proc. ICASSP*, 1998, vol. II, pp. 705–708.

[7] R. Nishimura, K. Komatsu, Y. Kuroda, K. Nagatomo, A. Lee, H. Saruwatari, and K. Shikano, "Automatic n-gram language model creation from web resources," in *Proc. EUROSPEECH*, 2001, pp. 2127–2130.

[8] A. Ito, H. Saitoh, M. Katoh, and M. Kohda, "N-gram language model adaptation using small corpus for spoken dialog recognition," in *Proc. EUROSPEECH*, 1997, pp. 2735–2738.

[9] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, O. Imaichi, and T. Imamura, "Japanese morphological analysis system ChaSen manual," NAIST Technical Report NAIST-IS-TR97007, Nara Institute of Science and Technology, 1997.

[10] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," in *Proc. Second International Conference on Language Resources and Evaluation (LREC)*, 2000, pp. 947–952.

[11] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR)*, 2003.

[12] "Google Japan," http://www.google.co.jp/.

[13] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine," in *Proc. EUROSPEECH*, 2001, pp. 1691–1694.