Training a language model using webdata for large vocabulary Japanese spontaneous speech recognition

Ryo Masumura, Seongjun Hahm, Akinori Ito

Graduate School of Engineering, Tohoku University
{ryo77373,branden65,aito}@spcom.ecei.tohoku.ac.jp

Abstract

This paper describes a language modeling method using large-scale spoken language data retrieved from the Web for spontaneous speech recognition. We downloaded 15 million Web pages on a comprehensive range topics. Next, spoken language-like texts were selected from the downloaded Web data using the naïve Bayes classifier, and typical linguistic phenomena such as fillers and pauses were added using simulation models. A language model trained by the generated data gave as high performance as the large-scale spontaneous speech corpus (Corpus of Spontaneous Japanese, CSJ). By combining the generated data and CSJ, we improved word accuracy.

Index Terms: Spontaneous speech recognition, language model, World Wide Web, large vocabulary continuous speech recognition, Corpus of Spontaneous Japanese

1. Introduction

Recent progress in spontaneous speech recognition relies on large-scale spontaneous speech corpora. For example, the Corpus of Spontaneous Japanese (CSJ) [1] is one of the largest spontaneous speech corpora, containing about 7 million words carefully transcribed by human transcribers. The CSJ is a powerful resource for not only training acoustic models but also language models [2], and is considered to be large enough for acoustic modeling [3]. However, manually-transcribed data is not sufficient for modeling a language model for large vocabulary continuous speech recognition. The number of distinct words in the CSJ is only around 60,000, which is too small for training a language model covering a wide variety of topics. Furthermore, amount of data itself is also too small to cover trigrams. When trigram coverage is insufficient, most trigram probabilities are calculated through back-off smoothing, which severely degrades the word accuracy. In short, to improve word accuracy, we need more transcriptions of spontaneous speech [4].

A popular approach to solve this problem is language model adaptation. This approach mixes topic-specific text data and text-independent transcription to obtain a topic-specific language model. Ideally, the topic-specific data should be in a spontaneous speech style. However, if we use a major data source such as newspapers or web pages, most of the sentences are in written style, and so mixing the written-style text data weakens the spontaneous speech characteristics of the adapted language model. Adaptation methods such as LDA or pLSA preserve the style characteristics of the adapted language model while adapting the topics of the language model to the target topic. However, topic-model-based adaptation does not solve the problem caused by back-off smoothing.

In this paper, we focus on obtaining linguistic data for spontaneous speech recognition from the Web, which is a useful source for language modeling [5]. Although most of the texts on the Web are in written style, the huge amount of data available allows us to obtain a sufficient amount of linguistic data by choosing speech-style-like data among the webdata. Misu et al. [6] reported on language model training using webdata for recognizing only query speech for a spoken QA system. A similar approach can be useful for gathering a large-scale spontaneous speech corpus from the Web.

This paper describes our attempt to gather speech-style linguistic data from the Web and generate a general language model for recognition of spontaneous speech. The language model is generated in the following three steps:

1. Data acquisition: Gather a large number of text data from the Web.
2. Data selection: Choose speech-style-like data from the gathered text data.
3. Data compensation: Insert fillers and short pauses for simulating linguistic phenomena observed in spontaneous speech.

The techniques used in these steps are not novel. The purpose of this paper is to demonstrate that a large-scale spontaneous-speech-like corpus can be created by combining existing technologies.

2. Large-scale spontaneous speech corpus generated from webdata

2.1. Retrieval of web data

As the first step, we need to retrieve a large amount of webdata from which speech-style texts are chosen. The usual method of retrieving webdata retrieval is to use keywords or phrases to retrieve candidates for downloading using a web search engine [7, 8, 9]. However, a keyword approach has the following two problems. First, it is difficult to choose keywords that characterize spontaneous speech. Speech style is often characterized by fillers or function words, most of which are stop words of a search engine. Phrase searches such as searching for “uh yes” could work, but it is also difficult to list all possible phrases that effectively characterize speech style. Second,text data gathered by keyword-based search inevitably contains biases. It is difficult to gather a large amount of text data on various topic uniformly through simple keyword searches or web crawling.

To solve this problem, we first gather web pages using an arbitrary noun as a keyword. Then the retrieval is repeated for *all* known nouns and so a vast amount of web data is downloaded. Figure 1 shows the retrieval process. By using all nouns as keywords, we can guarantee that the downloaded data con-

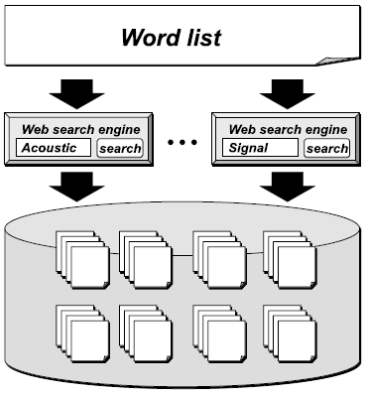

Figure 1: Retrieval of web documents using each of all nouns as a keyword

Table 1: Downloaded Web data

Search engine	Yahoo! Japan
Downloading period	Feb. 2010 to Apr. 2010
No. of keywords	288k nouns (IPAdic[12] and unidic[13])
No. of downloaded pages	15M
No. of downloaded distinct words	396k

tains texts relevant to any noun even if there are only a few pages concerning that noun on the Web.

Table 1 shows the conditions used for downloading. We used 288k nouns contained in the dictionary of a morphemic analyzer. Then we downloaded 50 web pages for each keyword, and finally downloaded about 15M pages. The pages downloaded for one keyword are combined into one text document associated with the keyword.

2.2. Filtering of the downloaded data

The downloaded pages were filtered for excluding HTML tags, Javascript codes and other parts that were not useful for language modeling. The filtering was based on a method used by Nisimura [5], which was performed in two steps. The first step was rule-based, which depends on Japanese orthography, as follows.

1. Extract lines that end with punctuation marks.
2. Exclude lines that have more than 20% alphabetical letters, digits and symbols. (Other characters in a line are either *kana* or *kanji*.)
3. Exclude lines shorter than 10 characters.

In the second step, word fragments not useful for language model training were excluded using word n-gram. We trained a word-based trigram from the CSJ, where all nouns are substituted into one class symbol to avoid task dependency. Then we formatted the extracted data into sentences, and measured the word perplexity of each sentence. Finally, those sentences having a perplexity of more than 400 were excluded from the corpus.

2.3. Selection of speech-style-like data

After retrieving the linguistic data from the Web, we chose speech-style-like sentences among the formatted data. We employed the naïve Bayes classifier for selecting the sentence. Let a sentence style $S \in \{0, 1\}$, where 1 means that the style is a speech style. We employ unigram language model θ_S for each style.

Let a document

$$D = \{(w_1, t(w_1, D)), \ldots, (w_V, t(w_V, D))\}, \quad (1)$$

where w_i is the i-th distinct word and $t(w_i, D)$ is the number of occurrences of w_i in the document D. Then the probability of generating D from θ_S is

$$P(D|\theta_S) = \prod_{(w,t)\in D} P(w|\theta_S)^t. \quad (2)$$

Then the estimation of style is

$$\hat{S} = \operatorname*{argmax}_S P(D|\theta_S). \quad (3)$$

The model θ_S is built as a multinomial distribution model. The probability is

$$P(w|\theta_S; \mu) = \frac{t(w, S) + \mu P_C(C(w)|\theta_S)}{\sum_{j=1}^{V} t(w_j, S) + \mu \sum_{j=1}^{V} P_C(C(w_j)|\theta_S)} \quad (4)$$

where $C(w)$ is the part-of-speech of word w,

$$t(w, S) = \sum_{D\in\mathcal{D}_S} t(w, D), \quad (5)$$

$\mathcal{D}_S$ is the set of documents that belong to style S, and

$$P_C(C|\theta_S) = \frac{t(C, S)}{\sum_{C'} t(C', S)} \quad (6)$$

$$t(C, S) = \sum_{D\in\mathcal{D}_S} \sum_{w\in C} t(w, D). \quad (7)$$

Here, μ is a smoothing parameter, and is estimated using leaving-one-out likelihood[14].

As the style of a document is independent of its topic, we excluded all nouns from documents for style discrimination. We used the CSJ for training the speaking style model and two years of newspaper articles (Mainichi Shimbun) for training the writing style model. Then the style of a document was estimated using Eq. (3).

2.4. Compensation of fillers and short pauses

Even if a document is classified as "speech style," most of such documents do not have the disfluencies observed in real spontaneous speech, such as fillers and short pauses. Therefore, we need to simulate these phenomena and insert them in the corpus. For generating disfluencies, we used models for predicting fillers [10] and short pauses [11].

Insertion of fillers is performed in two steps. First, given a sentence, positions for filler insertion are predicted. Next, a filler is selected among candidate fillers for each filler insertion position. While Ohta et al. used the conditional random field (CRF) for predicting filler insertion positions, we used the trigram model for simplicity. The probability that a filler is inserted after a word pair w_{i-1} and w_i is given by

Table 2: Comparison of data size

Data	Corpus size (Mwords)
CSJ	7.653
Filtered (all)	4,844
Filtered (written)	4,270
Filtered (spoken)	573.7
Filler	595.7
Short pause	611.3

$P_I(F|w_{i-1}w_i)$, where $F \in \{0, 1\}$, 1 means that a filler is inserted after w_i. w_k is basically a word, and all nouns are substituted by one symbol to avoid task dependency of the filler prediction. After a filler is predicted, a filler word is selected using a filler selection model, $P_F(f|w_{i-1}w_i)$, $f \in \{f_1, \ldots, f_{N_f}\}$, where f_k is a filler word. According to these probabilities, filler words are randomly inserted into the original sentences.

Insertion of short pauses was performed in the same way as the above-mentioned filler prediction. The filler model and short pause model were trained using the CSJ.

3. Experiment

3.1. Experimental conditions

We employed 2536 lectures taken from the CSJ for training. We trained five models from this training set: language models for text formatting, style classification, filler insertion, short pause insertion and a language model for usual speech recognition. We also used newspaper articles (Mainichi Shimbun) for two years (2000 and 2001, about 200k articles) for training the style classifier. Forty lectures from the CSJ that are not included in the training set were used for evaluation.

We used Julius 4.1.5 as a decoder and triphone models distributed with the CSJ as acoustic models.

3.2. Collected webdata and generated Web corpus

First, we compared the amount of collected data and CSJ. The comparison is shown in Table 2. In this table, "Filtered (all)" denotes all of the downloaded and filtered text data, "Filtered (written)" is the data classified as writing style, and "Filtered (spoken)" is that classified as speech style. These results show that almost one-eighth of the downloaded texts are classified as speech style. The collected speech style data were eighty times as large as the CSJ. The rows "Filler" and "Short pause" denote the data after filler insertion and short pause insertion (after filler insertion), respectively. In the "Short pause" corpus, 2.6% of the words were short pauses and 3.6% were filler words.

In the later experiments, "Filtered (spoken)", "Filler" and "Short pause" data were used.

Next, we investigated trigram coverage and out-of-vocabulary (OOV) rate. We prepared two vocabulary sets: a 40k set and a 300k set. As the number of distinct words appearing in the CSJ is 41696, the vocabulary size of the 40k set is comparable with that of the CSJ. The Web corpus is much larger than the CSJ, we also tested larger vocabulary size (300k).

Figure 2 shows the number of distinct trigrams and trigram coverage over the test set. The number of distinct trigrams in the Filtered (40k) corpus was twenty-five times as large as that of the CSJ. Trigram coverage of the Filtered corpus was 11.4 points higher than that of the CSJ. Inserting fillers and short pauses increased both the number of trigrams and the trigram coverage. The Filtered (300k) corpus had lower trigram coverage because of larger vocabulary size. If we increase the vocabulary size, rare trigrams occur more frequently in the test data, which are regarded as sequence of unknown words when a small vocabulary is used. On calculating the trigram coverage, an unknown word (UNK) is treated just like an ordinary word, and this is why the coverage become higher when vocabulary size is small. However, the coverage was still higher than that of the CSJ.

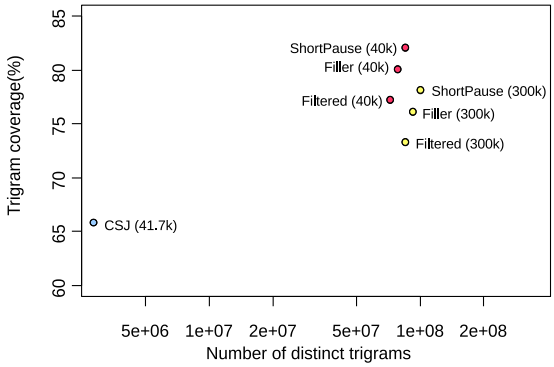

Figure 2: Number of distinct trigrams and coverage

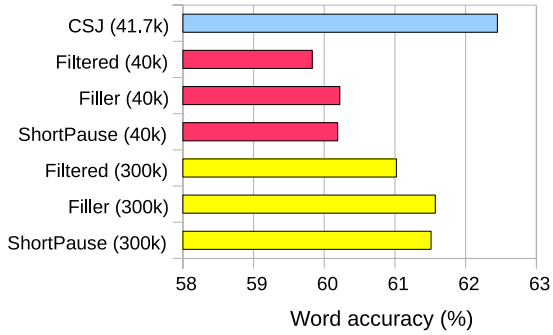

Figure 3: Word accuracy using downloaded corpora

Next, we investigated OOV rates when using language models trained from the CSJ with a 41.7k vocabulary and that from the Web corpus with 40k and 300k vocabularies. Note that the "Filtered," "Filler" and "ShortPause" corpora had identical OOV rates because the only differences of vocabularies of these corpora were filler words and a short pause symbol.

The OOV rate of the 40k vocabulary from the Web corpus was 1.15%, which was slightly lower than that of the CSJ (1.54%), although the CSJ and the Web 40k corpus had almost the same vocabulary size. When we employed the 300k vocabulary, the OOV rate was almost negligible (0.03%).

3.3. Speech recognition results

We performed speech recognition using trigram language models trained from the above-mentioned corpora and compared the recognition performance. Figure 3 shows the word accuracy results for LMs from the CSJ and the three Web-based corpora. These results confirm that insertion of fillers improved the word accuracy, but insertion of short pauses yielded no improvement. The accuracy of the "Filler" model of the 300k vocabulary was only 0.88 point behind that of the CSJ model.

We can improve the coverage of a trigram model by simply adding "Filter," "Filler" and "ShortPause" corpora because

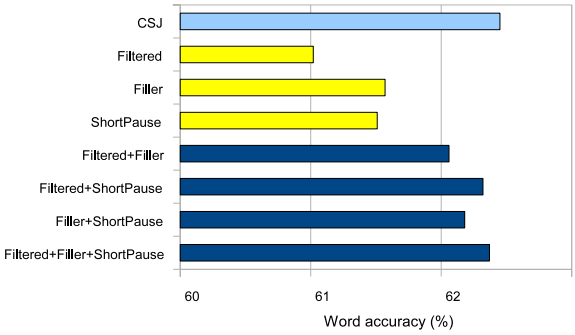

Figure 4: Word accuracy using combined corpora

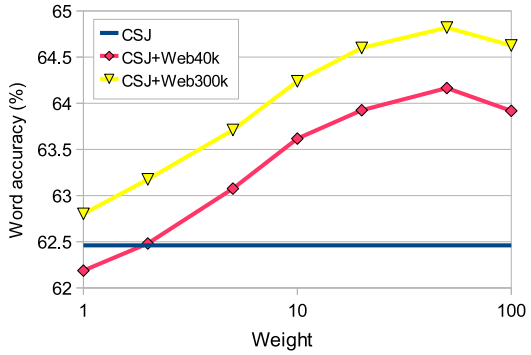

Figure 5: Result of combination of CSJ and Web corpus

we can train word sequence with and without disfluencies. In fact, by combining these three corpora, the trigram coverage increased to 79.49% when the 300k vocabulary was employed. Figure 4 shows the word accuracy results for combined corpora with the 300k vocabulary; we achieved a word accuracy of 62.37%, which was comparable with that by the CSJ (62.45%).

3.4. Combination of CSJ and Web corpora

Finally, we combined the Web corpus and the CSJ to improve the language model further. The two corpora were combined at the n-gram count level [15], with the n-gram count of the CSJ multiplied by a weighting factor. Figure 5 shows the word accuracy results. The best result was obtained when we multiplied the n-gram count of the CSJ by 50, and the best word accuracy for the 40k vocabulary model was 64.16% and that for the 300k vocabulary model was 64.82%.

4. Conclusions

In this paper, we trained a language model for spontaneous speech recognition from linguistic data downloaded from the Web. Although we used existing methods for selecting the spontaneous-speech-like data and simulating disfluencies, we believe that this work is the first attempt to gather a large-scale corpus from the Web for general-purpose spontaneous speech recognition. The results showed that good word accuracy was acheved by using the language model trained using the gathered corpus, which was comparable with the results by the CSJ, one of the largest spontaneous speech corpora.

As a future work, the performance of the language model trained by the web corpus should be evaluated using test data other than the CSJ, since the results presented in this paper might depend on the CSJ.

5. References

[1] K. Maekawa, H. Koiso, S. Furui and H. Isahara, "Spontaneous speech corpus of Japanese," Proc. LREC, pp. 947–952, 2000.

[2] M. Nakamura, S. Furui and K. Iwano, "Acoustic and linguistic characterization of spontaneous speech," Proc. Symp. on Large-Scale Knowledge Resources (LKR2007), pp. 163–168, 2007.

[3] S. Furui, M. Nakamura, T. Ichiba and K. Iwano, "Why is the recognition of spontaneous speech so hard?," Proc. 8th Int. Conf. on Text, Speech and Dialogue (TSD2005), pp. 9–22, 2005.

[4] S. F. Chen and J. Goodman, "An empirical study of smoothing technique for language modeling," Computer Speech and Language, vol. 13, pp. 359–394, 1999.

[5] R. Nisimura, K. Komatsu, Y. Kuroda, K. Nagatomo, A. Lee, H. Saruwatari and K. Shikano, "Automatic n-gram language model creation from Web resources," Proc. Eurospeech, pp. 2127–2130, 2001.

[6] T. Misu and T. Kawahara, "A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts," Proc. Interspeech, pp. 9–12, 2006.

[7] I. Bulyko, M. Ostendorf and A. Stolke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixture," Proc. HLT/NAACL, vol. 2, pp. 7–9, 2003.

[8] T. Ng, M.Ostendorf, M. Y. Hwang, M. Siu, I. Bulyko and X. Lei, "Web-data augmented language model for Mandarin speech recognition," Proc. ICASSP, vol. 1, pp. 589–592, 2005.

[9] A. Ito, Y. Kajiura, M. Suzuki and S. Makino, "Automatic Query Generation and Query Relevance Measurement for Unsupervised Language Model Adaptation of Speech Recognition," EURASIP J. Audio, Speech and Music Processing, Article ID 140575, 12 pages, doi:10.1155/2009/140575, 2009.

[10] K. Ohta, M. Tsuchiya and S. Nakagawa, "Evaluating spoken language model based on filler prediction method in speech recognition," Proc. Interspeech, pp. 1558–1561, 2008.

[11] K. Ohta, M. Tsuchiya and S. Nakagawa, "Effective use of pause information in language modeling for speech recognition," Proc. Interspeech, pp. 2691-2694, 2009.

[12] M. Asahara and Y. Matsumoto, "IPADIC User Manual," Nara Institute of Science and Technology, Japan, 2002.

[13] Y. Den, J. Nakamura, T. Ogiso and H. Ogura, "A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation," Proc. LREC, 2008.

[14] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," ACM TOIS, vol. 22, no.2, pp. 179–214, 2004.

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