Word N-gram Probability Estimation From A Japanese Raw Corpus

Shinsuke MORI, Daisuke TAKUMA

IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma Yamato-shi, 242-8502, Japan
forest@jp.ibm.com, ta9ma@jp.ibm.com

Abstract

Statistical language modeling plays an important role in a state-of-the-art speech recognizer. The most used language model (LM) is word n-gram model, which is based on the frequency of words and word sequences in a corpus. In various Asian languages, however, words are not delimited by whitespace, so we need to annotate sentences with word boundary information to prepare a statistically reliable large corpus. In this paper, we propose a method for building an LM directly from a raw corpus. In this method, sentences in the raw corpus are regarded as sentences annotated with stochastic word boundary information. In the experiments, we compared the predictive powers of an LM built only from a segmented coprus and an LM built from the segmented corpus and a raw corpus. The result showed that we succeeded in reducing the perplexity by 42.9% using a raw corpus by our method.

1. Introduction

All state-of-the-art speech recognizers [1] refer to an LM to choose the most likely candidate in cooperation with an acoustic model. The most important resource for stochastic language modeling is a corpus in the application domain, so basically one uses a large number of sentences and counts the frequencies of words and word sequences. For languages, such as Japanese and Chinese, in which the words are not delimited by whitespace, one first encounters a word identification problem. The best we can do is correct manually the outputs of an automatic word segmenter to prepare a segmented corpus in the application domain. The only robust and reliable word segmenter in the domain is, however, a word segmenter based on the lexical statistics on a segmented corpus in the domain.

Nowadays, it is easy to gather machine-readable sentences in various domains because of the ease of publication and access via the Web. In addition, traditional machine-readable form of medical reports or court reports are also available. When we need to develop an NLP system in many domains, there is a huge corpus without word boundary information.

In this paper, aiming at an application of a speech recognizer to a special domain, we propose a method for estimating word n-gram probabilities in a raw corpus in a realistic computation time and show that an inexpensive raw corpus in the target domain is useful as a source of word n-gram information. In the experiments, we compare the perplexities of mainly three models: 1) a word bi-gram model built from a segmented corpus, 2) a word bi-gram model built from the same segmented corpus interpolated with a word bi-gram model estimated from a raw corpus without word boundary information, 3) a word bi-gram model built from the same segmented corpus interpolated with a word bi-gram model built from the raw corpus segmented automatically by a word segmenter. As a result, the second model outperformed the others completely in perplexity (more than 40% reduction). We compare a word bi-gram model and a word uni-gram model estimated from the raw corpus. The use of word bi-gram model reduced the perplexity by 25%. We also show an experimental result about an influence of the raw corpus domain. We needed more than six times larger raw corpus in a different domain to achieve the perplexity reduction realized by a raw corpus in the same domain as the test corpus.

2. Language Model

The role of a language model (LM) is to measure the likelihood of a sequence of letters as a sentence in the language. A speech recognizer refers to an LM, as well as to an acoustic model, to choose the most likely word sequence.

2.1. Stochastic Language Model

The most famous LM is the word-based n-gram model. In this model, a sentence is regarded as a word sequence $\boldsymbol{w}_1^h$ $(= w_1 w_2 \cdots w_h)$ and words are predicted from left to right[1]:

$$M_{w,n}(\boldsymbol{w}_1^h) = \prod_{i=1}^{h+1} P(w_i | \boldsymbol{w}_{i-n+1}^{i-1}),$$

where w_i $(i \leq 0)$ and w_{h+1} is a special symbol called a `BT` (boundary token). Since it is impossible to define

[1] Throughout this paper letters in bold face denote a sequence.

the complete vocabulary, we prepare a special token UW for unknown words and an unknown word spelling $\boldsymbol{x}_1^{h'}$ is predicted by the following character-based n-gram model after UW is predicted by $M_{w,n}$:

$$M_{x,n}(\boldsymbol{x}_1^{h'}) = \prod_{i=1}^{h'+1} P(x_i|\boldsymbol{x}_{i-n+1}^{i-1}),$$

where $x_i, i \leq 0$ and $x_{h'+1}$ is a special symbol BT. Thus when w_i is out of the vocabulary $\mathcal{W}$,

$$P(w_i|\boldsymbol{w}_{i-n+1}^{i-1}) = M_{x,n}(w_i)P(\texttt{UW}|\boldsymbol{w}_{i-n+1}^{i-1}).$$

3. Word N-gram Probability Estimation From A Raw Corpus

In this section, we propose a mathematically sound method of calculating word n-gram probabilities on a raw corpus for languages in which the words are not delimited by whitespace. This method is based on the probability that a word boundary exists at each point between characters in an unsegmented corpus.

3.1. Word Boundary Probability

Given a relatively small segmented corpus C_s, first we estimate the probability P_i that a word boundary exists between two characters x_i and x_{i+1}. Since the size of the segmented corpus may be small and the number of characters in Japanese is large (approximately 6,000), we introduce seven character classes for separator characters, Chinese characters (*Kanji*), symbols, arabic digits, *Hiragana*[2], *Katakana*, and Latin characters (including Cyrilic and Greek characters). The word boundary probability between each two characters is estimated from the segmented corpus by the maximum likelihood estimation method as follows:

$$P_i = \frac{f_s(c(x_i), \texttt{BT}, c(x_{i+1}))}{f_s(c(x_i), \texttt{BT}, c(x_{i+1})) + f_s(c(x_i), c(x_{i+1}))},$$

where $c(x)$ is the character class which x belongs to, BT is a word boundary token in the segmented corpus, and $f_s(\boldsymbol{x})$ is the frequency of the string $\boldsymbol{x}$ in the segmented corpus.

3.2. Stochastically Segmented Corpus

We regard the unsegmented raw corpus C_r (hereafter referred to as the character sequence $\boldsymbol{x}_1^{n_r}$) as a stochastically segmented corpus where each point between characters x_i and x_{i+1} is a word boundary with the probability P_i. Then the expectation of the occurrence of words in the raw corpus C_r is as follows:

$$f_r(\cdot) = 1 + \sum_{i=1}^{n_r-1} P_i.$$

[2] *Hiragana* are used to indicate grammatical function and for some certain Japanese words

3.3. Word Uni-gram Probability

A character sequence $\boldsymbol{x}_{i+1}^{i+k}$ in the raw corpus is a word $w = \boldsymbol{x}_{i+1}^{i+k}$ if and only if the following three conditions are met:

1. there is a word boundary before the leftmost character x_{i+1}, that is $X_i = 1$,
2. there is no word boundary inside the sequence, that is $X_{i+j} = 0$, where $1 \leq j \leq k-1$,
3. there is a word boundary after the rightmost character x_{i+k}, that is $X_{i+k} = 1$.

Thus, the stochastic frequency f_r for a word w in the raw corpus is defined by the summation of the stochastic frequency at each occurrence of the character sequence of the word w over all the occurrences in the raw corpus $O_1 = \{i \,|\, \boldsymbol{x}_{i+1}^{i+k} = w\}$ as follows:

$$f_r(w) = \sum_{i \in O_1} P_i \left[\prod_{j=1}^{k-1}(1 - P_{i+j})\right] P_{i+k}. \quad (1)$$

It can be shown that f_r is the expectation of the occurrence of w in the raw corpus. Then the word uni-gram probability is

$$P_r(w) = \frac{f_r(w)}{f_r(\cdot)}.$$

Due to space limitations, we omit the proof for the well-definedness of P_r.

3.4. Word N-gram Probability

Similarly, the word n-gram probability in the raw corpus can be defined. Since the notation is too complecated, we show only the case of bi-gram as follows:

$$P_r(w_2|w_1) = \frac{f_r(w_1 w_2)}{f_r(w_1)}, \quad (2)$$

where

$$f_r(w_1 w_2) = \sum_{i \in O_2} \left(P_i \left[\prod_{j=1}^{k-1}(1 - P_{i+j})\right] P_{i+k} \times \left[\prod_{j=1}^{l-1}(1 - P_{i+k+j})\right] P_{i+k+l} \right) \quad (3)$$

$$O_2 = \{i \,|\, \boldsymbol{x}_{i+1}^{i+k} = w_1 \wedge \boldsymbol{x}_{i+k+1}^{i+k+l} = w_2\}.$$

Due to space limitations, we omit the proof for the well-definedness of P_r.

As we can see from Equations (1) (2) (3), word bi-gram probability in the raw corpus $P_r(w_2|w_1)$ is estimated by only consulting the preceding character and following character at each occurence of $w_1 w_2$ and w_2 as a character sequence in the raw corpus. This process is executed efficiently by using a suffix array [2].

Table 1: Segmented Corpus.

	#sentences	#words	#chars
learning	4,117	87,383	152,802
test	323	4,624	7,897

Table 2: Raw Corpus.

source	#chars	estimated #words
interview	8,800,306	5,032,637.9
newspaper (1)	8,800,418	5,032,702.0
newspaper (2)	54,415,092	31,118,401.5

The estimated numbers of words are calculated by assuming that the avarage word length is the same as in the segmented learning corpus.

3.5. Interpolation with a Word N-gram Model Built from a Segmented Corpus

Word-based n-gram probability estimated from a raw corpus may not be as accurate as an LM estimated from a corpus segmented appropriately by hand. Thus we use the following interpolation technique:

$$P(w_i|H_i) = \lambda_s P_s(w_i|H_i) + \lambda_r P_r(w_i|H_i),$$

where H_i is history before w_i, P_s is the probability estimated from a segmented corpus C_s, and P_r is the probability estimated by our method from a raw corpus C_r. λ_s and λ_r are interpolation coefficients which are estimated by the deleted interpolation method [3]. From this formulation, we can say that our model can use frequency information and context information of a word which never appears in the segmented corpus. This advantage is emphasized more when this method is used with a subword model [4] to recognize unknown words.

4. Evaluation

As an evaluation of the language model estimation from a raw corpus, we built various LMs and calculated their test set perplexities.

4.1. Corpora

Aiming at an application of a speech recognizer to an interview transcription task, we gathered interview transcriptions and segmented 4,440 sentences correctly by hand. These sentences are divided into ten parts, and the parameters of the model, including the word boundary probability, were estimated from nine of them (learning) and the model was tested on the remaining one (test). Table 1 shows the sizes of the segmented corpora. The raw corpora used in our experiments are a set of interview transcriptions and a set of articles of *Mainichi* newspaper in 1997. Table 2 shows the sizes of the raw corpora. In Table 2 "newspaper (1)" is a subset of the one year newspaper corpus "newspaper (2)" and contains approximately the same number of characters as "interview."

4.2. Criterion

The criterion we used for LMs is word-based test set perplexity PP. First we calculate the entropy H for all words in the test corpus C_t including unknown words as follows [5]:

$$H = -\log_2 \prod_{\boldsymbol{w} \in C_t} M_{w,n}(\boldsymbol{w}).$$

Then word-based test set perplexity is calculated as follows:

$$PP = 2^{H/\#\text{word}},$$

where #word stands for the number of words in the test corpus.

4.3. Models

We built six models for various comparisons.

Model A: As a baseline model we build a word bi-gram model from the segmented learning corpus.

Model B: A known method for using a raw corpus is to segment its sentences by a word segmenter. Thus we built a word segmenter based on the baseline model and segmented sentences in the raw corpus of interview transcriptions. Then we estimated word bi-gram model from those automatically segmented sentences and interpolated it with the baseline model.

Model C: We estimated word bi-gram model from the raw corpus in the same domain as the test corpus and interpolated it with the baseline model.

The following three models are variations of Model C.

Model D: The word uni-gram version of Model C

Model E: With a raw corpus of the same size but in a different domain

Model F: With a raw corpus in a different domain resulting a similar performance as Model C

4.4. Results and Discussion

Table 3 shows the perplexity of the models. From a comparison among Model A, B, and C, it can be concluded that regarding a raw corpus as a stochastically segmented corpus is a good way of using a raw corpus. Though the accuracy of the word segmenter used for Model B is high

Table 3: Test Set Perplexity.

	raw corpus	method	model	PP
A	no	–	–	140.86
B	interview	auto. seg.	bi-gram	141.71
C	interview	stoch. seg.	bi-gram	80.49
D	interview	stoch. seg.	uni-gram	107.05
E	newspaper (1)	stoch. seg.	bi-gram	95.69
F	newspaper (2)	stoch. seg.	bi-gram	80.69

All models are interpolated with a word bi-gram model built from the segemented corpus.

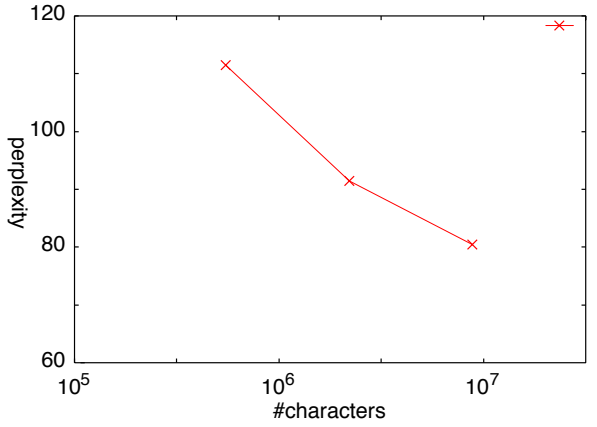

Figure 1: Relation between raw corpus size and perplexity.

(96.37% on test corpus), Model B did not perform better than Model A. The reason may be as follows. Since the word segmenter tends to make mistakes around unknown words, Model B fails to capture word n-gram information containing unknown words.

From a comparison between Model C and D, one can say that word bi-gram information extracted from a raw coprus improves the baseline LM significantly. According to the result of Model E, the raw corpus in a different domain from the target corpus is much less effective than the raw corpus in the target domain. A comparison with the result of Model F tells us that about 6.2 times larger newspaper corpus was needed to yield the same reduction in perplexity as the raw corpus in the target domain.

In order to clarify the influence of the size of the raw corpus, we calculated the perplexity of Model C changing the size of the raw corpus (1/1, 1/4, 1/16). Figure 1 shows the result. In this graph the perplexity is still strongly decreasing even at the rightmost point. Thus it is worth gathering raw sentences to prepare a far larger corpus.

In many domains, a huge raw corpus is available almost freely. Thus it is a good strategy to gather as many sentences as possible in the target domain and use a word n-gram model estimated from them by our method when one want to have a good speech recognizer in a new domain in many Asian languages as well.

5. Conclusion

In this paper, aiming at an application of a speech recognizer to a special domain, we proposed a method for estimating word n-gram probability from a raw corpus in languages, such as Japanese and Chinese, in which the words are not delimited by whitespace. We built various language models and checked out the efficiency of the use of a raw corpus. From the experimental results we conclude that it is a good strategy for language modeling in Japanese to gather as many sentences as possible in the target domain and use a word n-gram model estimated from them using the method we proposed in this paper.

6. References

[1] F. Jelinek. Self-organized language modeling for speech recognition. Technical report, IBM T. J. Watson Research Center, 1985.

[2] Udi Manber and Gene Myers. Suffix arrays: A new method for on-line string searches. *SIAM J. Comput*, 22(5):935–948, 1993.

[3] 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.

[4] Yoshihiko Ogawa, Hirofumi Yamamoto, Yoshinori Sagisaka, and Genichiro Kikui. Word class modeling for speech recognition with out-of-task words using a hierarchical language model. In *Proceedings of the Eighth European Conference on Speech Communication and Technology*, volume 1, pages 221–224, 2003.

[5] Peter F. Brown, Stephen A. Della Pietra, and Robert L. Mercer. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40, 1992.