Language Model Adaptation with a Word List and a Raw Corpus

Shinsuke MORI

IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.

forest@jp.ibm.com

10.21437/Interspeech.2006-513

Abstract

In this paper, we discuss language model adaptation methods given a word list and a raw corpus. In this situation, the general method is to segment the raw corpus automatically using a word list, correct the output sentences by hand, and build a model from the segmented corpus. In this sentence-by-sentence error correction method, however, the annotator encounters grammatically complicated positions and this results in a decrease of productivity. In this paper, we propose to concentrate on correcting the positions in which the words in the list appear by taking a word as a correction unit. This method allows us to avoid these problems and go directly to capturing the statistical behavior of specific words in the application. In the experiments, we used a variety of methods for preparing a segmented corpus and compared the language models by their speech recognition accuracies. The results showed the advantages of our method.

Index Terms: language model adaptation, word list, raw corpus, stochastic segmentation, speech recognition

1. Introduction

Many language models (LMs) are based on the frequencies of words and word sequences counted in a corpus. In agglutinative languages such as Japanese an LM requires a corpus with word boundary information (a segmented corpus). A segmented corpus is, however, available only for general domains. In most applications, such as a speech recognition system for medical records or call center logs, a corpus without word boundary information (a raw corpus) and a word list in the target domain are the only available resources. In these situations, all of the sentences in the raw corpus are first segmented into words by an automatic word segmenter with the word list[1], then some of the output sentences are corrected by hand, and finally the frequencies are counted in the whole corpus.

The more the sentences are corrected by hand, the more reliable the statistics become, and the more the LM is improved. However manual error correction takes a lot of time and expense. Thus in most cases only a small set of sentences is manually checked and the rest are used as they are. In this sentence-by-sentence error correction method, however, the annotator encounters grammatically complicated positions and this results in a decrease of the annotator's productivity. In addition, it is not certain that sentence-by-sentence error correction from the beginning is the best way to allocate a limited work force [2].

In this paper, we propose taking a word as a correction unit and concentrate on correcting the positions in which words in the list appear. This method allows us to avoid the difficulties mentioned above and go directly to capturing the statistical behavior of specific words in the application field. The resulting corpus contains sentences partially annotated with word boundary information. In order to estimate word n-gram probabilities from such sentences, we extend a method for estimating word n-gram probabilities of an infinite vocabulary from a raw corpus [3] to a limited vocabulary case.

In the experiments, we used a variety of methods for preparing a segmented corpus and compared the language models from the corpora in predictive power and speech recognition accuracy. The results showed that concentrating on the error correction around the words in the list, we can build a better language model with less effort.

2. Language models and their applications

A stochastic LM M is a probability function for sequence of characters $\boldsymbol{x} \in \mathcal{X}^*$. The summation over all possible sequences of characters must not be grater than 1. This probability function is used for the likelihood in an NLP system.

2.1. Word n-gram model

The most widely known LM is an n-gram model based on words. 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 beginning to end:

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

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 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 when a `UW` is predicted by $M_{w,n}$:

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

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

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

2.2. Automatic word segmentation

Nagata [1] proposed a stochastic word segmenter based on a word n-gram model to solve the word segmentation problem. According to this method, the word segmenter divides a sentence $\boldsymbol{x}$ into a word sequence with the highest probability

$$\hat{\boldsymbol{w}} = \mathop{\mathbf{argmax}}_{\boldsymbol{w}=\boldsymbol{x}} M_{w,n}(\boldsymbol{w}).$$

3. Language model adaptation with a word list and a raw corpus

In this section, we propose a novel method for adapting an LM assuming that a word list and a raw corpus in the target domain are available.

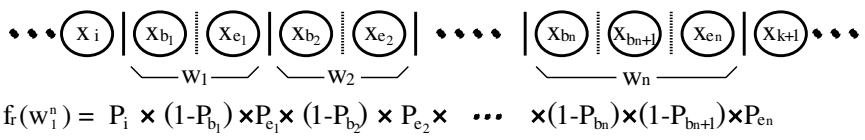

Figure 1: Word n-gram frequency in a stochastically segmented corpus (SSC).

3.1. Word n-gram probability estimation from a stochastically segmented corpus (SSC)

Generally speaking we need a corpus manually segmented into word sequences to estimate an LM. An easily available raw corpus lacks word boundary information. However there is a method for estimating an LM from a raw corpus by regarding it as a stochastically segmented corpus (SSC) [3]. Below we explain this method and extend it.

An SSC is defined as a combination of a raw corpus C_r (hereafter referred to as the character sequence $\boldsymbol{x}_1^{n_r}$) and word boundary probabilities P_i that a word boundary exists between two characters x_i and x_{i+1}. Given an SSC word n-gram frequencies are calculated as follows:

Word 0-gram frequency: This is defined as an expected number of words in the SSC:

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

Word n-gram frequency ($n \geq 1$): Let us think of a situation (see Figure 1) in which a word sequence $\boldsymbol{w}_1^n$ occurs in the SSC as a subsequence beginning at the $(i+1)$-th character and ending at the k-th character and each word w_m in the word sequence is equal to the character sequence beginning at the b_m-th character and ending at the e_m-th character ($\boldsymbol{x}_{b_m}^{e_m} = w_m,\ 1 \leq \forall m \leq n;\ e_m + 1 = b_{m+1},\ 1 \leq \forall m \leq n-1;\ b_1 = i+1;\ e_n = k$). The word n-gram frequency of a word sequence $f_r(\boldsymbol{w}_1^n)$ in the SSC is defined by the summation of the stochastic frequency at each occurrence of the character sequence of the word sequence $\boldsymbol{w}_1^n$ over all of the occurrences in the SSC:

$$f_r(\boldsymbol{w}_1^n) = \sum_{(i,\boldsymbol{e}_1^n)\in O_n} P_i \left[\prod_{m=1}^{n} \left\{ \prod_{j=b_m}^{e_m-1} (1-P_j) \right\} P_{e_m} \right],$$

where $\boldsymbol{e}_1^n = (e_1, e_2, \cdots, e_n)$ and $O_n = \{(i, e_1^n) | \boldsymbol{x}_{b_m}^{e_m} = w_m, 1 \leq m \leq n\}$.

Word n-gram probability: Similar to the word n-gram probability estimation from a decisively segmented corpus, word n-gram probabilities in the SSC are estimated by the maximum likelihood estimation method as relative values of word n-gram frequencies:

$$\begin{aligned} P_r(w) &= f_r(w)/f_r(\cdot), \\ P_r(w_n|\boldsymbol{w}_1^{n-1}) &= f_r(\boldsymbol{w}_1^n)/f_r(\boldsymbol{w}_1^{n-1}) \quad (n \geq 2). \end{aligned}$$

3.2. Word n-gram probability estimation from an SSC on a limited vocabulary

In speech recognition, words in the vocabulary must be annotated with their pronunciations. Thus in real applications, we need to estimate word n-gram probabilities on limited vocabularies including the unknown word symbol.

For a segmented corpus, all the words outside of the vocabulary $\mathcal{W}_k$ are replaced with the unknown word symbol `UW` and the word n-gram frequencies are determined of considering `UW` as a word. For an SSC, we can not follow this straight-forward strategy. However, the following characteristics enable us to calculate n-gram frequencies on the SSC[1].

$$\begin{cases} f_r(\boldsymbol{w}_u \texttt{UW} \boldsymbol{w}_v) = \displaystyle\sum_{w\in\mathcal{X}^+-\mathcal{W}_k} f_r(\boldsymbol{w}_u w \boldsymbol{w}_v) \\ \displaystyle\sum_{w\in\mathcal{X}^+} f_r(\boldsymbol{w}_u w \boldsymbol{w}_v) = \sum_{w\in\mathcal{X}^+-\mathcal{W}_k} f_r(\boldsymbol{w}_u w \boldsymbol{w}_v) + \sum_{w\in\mathcal{W}_k} f_r(\boldsymbol{w}_u w \boldsymbol{w}_v) \end{cases}$$
$$\Rightarrow f_r(\boldsymbol{w}_u \texttt{UW} \boldsymbol{w}_v) = \sum_{w\in\mathcal{X}^+} f_r(\boldsymbol{w}_u w \boldsymbol{w}_v) - \sum_{w\in\mathcal{W}_k} f_r(\boldsymbol{w}_u w \boldsymbol{w}_v), \quad (2)$$

where $\boldsymbol{w}_u, \boldsymbol{w}_v \in (\mathcal{W}_k \cup \{\texttt{UW}\})^*$ are arbitrary sequences consisting of words in the vocabulary $\mathcal{W}_k$ and `UW`.

Uni-gram frequency of `UW`. The uni-gram frequency of `UW` on the SSC is calculated using the following relationship between word uni-gram frequencies and the word zero-gram frequency

$$f_r(\cdot) = \sum_{w\in\mathcal{X}^+} f_r(w)$$

and the following equation obtained by letting $\boldsymbol{w}_u = \boldsymbol{w}_v = \varepsilon$ (ε represents a null string) in Equation (2)

$$f_r(\texttt{UW}) = \sum_{w\in\mathcal{X}^+} f_r(w) - \sum_{w\in\mathcal{W}_k} f_r(w).$$

Thus the word zero-gram frequency is as follows:

$$f_r(\texttt{UW}) = f_r(\cdot) - \sum_{w\in\mathcal{W}_k} f_r(w).$$

Word bi-gram frequency containing `UW`. The frequency with a sequence of an arbitrary word $w_1 \in \mathcal{W}_k$ and `UW` in the SSC $f_r(w_1\texttt{UW})$ is calculated using the relationship between the word bi-gram frequency and the word uni-gram frequencies

$$f_r(w_1) = \sum_{w\in\mathcal{X}^+} f_r(w_1 w), \quad \forall w_1 \in \mathcal{W}_k \cup \{\texttt{UW}\}$$

and the following equation obtained by letting $\boldsymbol{w}_u = w_1,\ \boldsymbol{w}_v = \varepsilon$ in Equation (2)

$$f_r(w_1\texttt{UW}) = \sum_{w\in\mathcal{X}^+} f_r(w_1 w) - \sum_{w\in\mathcal{W}_k} f_r(w_1 w).$$

Thus the word bi-gram frequency $f_r(w_1\texttt{UW})$ is calculated as follows:

$$f_r(w_1\texttt{UW}) = f_r(w_1) - \sum_{w\in\mathcal{W}_k} f_r(w_1 w).$$

Similarly the frequency of a sequence of `UW` and an arbitrary word $w_2 \in \mathcal{W}_k$ in the SSC $f_r(w_2\texttt{UW})$ is calculated as follows:

$$f_r(\texttt{UW}w_2) = f_r(w_2) - \sum_{w\in\mathcal{W}_k} f_r(w w_2).$$

[1] More precisely, n-gram sequences containing multiple unknown word symbols have to be considered. We are only describing n-gram sequences containing only one unknown word symbol for simplicity in this paper.

	エコノミスト、キャ	サリン	・カミリさんなどは
	ム・デーヴィッド・	サリン	ジャーは 20 世紀アメ
○	ぐおぞましい地下鉄	サリン	事件、長い不況に追
○	始まった中川被告は	サリン	生成を認めながら「
	いるのを知りながら	サリン	流出を阻止する義務

Figure 2: An example of corpus annotation by KWIC.

In addition, the word bi-gram frequency of a two UW sequence $f_r(\texttt{UW}\,\texttt{UW})$ is calculated using

$$f_r(\cdot) = \sum_{w_1 \in \mathcal{X}^+} \sum_{w_2 \in \mathcal{X}^+} f_r(w_1 w_2)$$

as follows:

$$f_r(\texttt{UW}\,\texttt{UW}) = f_r(\cdot) - \sum_{w_1 \in \mathcal{W}_k} f_r(w_1\texttt{UW}) - \sum_{w_2 \in \mathcal{W}_k} f_r(\texttt{UW}w_2) - \sum_{(w_1 w_2) \in \mathcal{W}_k \times \mathcal{W}_k} f_r(w_1 w_2).$$

Word n-gram frequency containing UW ($n \geq 3$). Frequencies of an n word sequence containing one or more than one UWs are also calculated in the same way as the bi-gram case.

Word n-gram probability containing UW ($n \geq 1$). Same in the case of normal word n-gram probabilities, word n-gram probabilities containing UW are estimated by dividing the word n-gram frequency by the word $(n-1)$-gram frequency.

Now we can estimate word n-gram probabilities on a limited vocabulary with the unknown word symbol in an SSC

4. Usage of a raw corpus

A raw corpus in the target domain is important to capture the linguistic characteristics in the application field. There are three ways to use the raw corpus.

1. Unknown word extraction
Word candidates are extracted from the raw corpus based on a criterion such as an accessor variety [4]. The extracted words are manually checked and annotated with their pronunciations.

2. Automatically segmented corpus
Word n-gram frequencies are counted in the automatic segmentation results of the raw corpus. Although the automatic word segmenter [1] has a tendency to make mistakes around specific words in the target domain, this method is practical and widely adopted.

3. Manually segmented corpus
In the ideal situation in which all the sentences in the raw corpus in the target domain are correctly segmented, the performance of the LM is as high as possible.

The more the sentences an annotator corrects, the more the LM is improved. In practice, however, the corpus correction requires a large amount of time and expense. Thus only some of the sentences are corrected and the other automatically segmented sentences are used as they stand. We encounter here the question of whether this sentence-by-sentence correction is the best way for allocating a limited workforce.

To correct word segmentation errors means to add proper word boundary information. The smallest unit of word boundary information is the existence or absence of a word boundary between each two characters. A sentence-by-sentence correction, however, obliges the annotator to make decisions on all of the points in a sentence. In contrast we propose to consider each word as a correction unit and concentrate on correcting the positions in which the words in the list appear. Concretely speaking, as shown in Figure 2, the annotator sees a KWIC (Key Word In Context) for a word (ex. サリン) in the list with contexts and marks the occurrences if the string in question is used as a word in that context or does nothing. It is better to limiting the number of marks for each word. This limitation prevents the annotator from spending time on linguistically complicated points.

Table 1: Corpora.

usage	domain	#chars	#words	#sents
learning	conversation	14,754	187,658	254,436
learning	newspaper	20,700	625,761	917,830
test	conversation	1,639	21,105	28,655
test	newspaper	2,300	68,566	100,091

5. Evaluation

As an evaluation of our corpus correction framework, we measured the predictive powers of LMs built from segmented corpora of various correction frameworks and the character error rates of a pseudo-recognition task whose input is a phoneme-sequence, not a voice signal. In this section we show the results and evaluate our new framework.

5.1. Conditions for the Experiments

The corpus in the general domain used in our experiments is composed of example sentences in a dictionary of daily conversation. The corpus in the target domain is composed of articles extracted from newspapers (see Table 1). Both corpora were segmented into words manually. The corpus in the target domain is, however, mainly used as a raw corpus. Its word boundary information is used to simulate an ideal situation in which we have a correctly segmented corpus in the target domain. The word list in the target domain contains 21,855 words appearing in the target domain corpus.

The baseline LM is built as follows:

Base. We built a word bi-gram model form the segmented corpus in the general domain. The vocabulary contains 5,112 words. The generation probabilities of the words in the word list are increased slightly by distributing the summation of the generation probabilities of the known words [5]. The target domain corpus is not used.

The test set perplexity of the LM on the test corpus in the general domain was 64.28. The automatic word segmenter used in the experiments is based on the baseline LM and returns the word sequence with the highest probability (see Subsection 2.2). The word boundary estimation accuracy on the test corpus in the general domain was 98.26% [2].

In the experiments described below, we adopted the word boundary probability estimation method described in [3]. That is, the raw corpus is segmented by the word segmenter and P_i is set to be the word boundary estimation accuracy (α = 98.26%) for each i where the word segmenter put a word boundary and P_i is set to $1 - \alpha$ for each i where it did not put a word boundary.

[2] The word boundary estimation accuracy on the test corpus in the target domain was 89.25%.

Table 2: Predictive powers and character error rates (CER) of each model.

Model	The raw corpus usage	PP	CER
Base	–	1938	37.26%
Auto	Automatic segmentation	755.7	19.48%
Raw	Stochastic segmentation	536.5	15.30%
Rare	Partial correction	465.2	13.43%
Medium	Partial correction	364.2	11.66%
45%-done	Partial correction	427.4	13.44%
Well-done	Complete correction	353.1	11.10%

5.2. Usages of the raw corpus in the target domain

In order to compare the usages of the raw corpus in the target domain, we prepared the following six methods changing the strategy or the amount of manual error correction.

Auto. The target domain corpus is automatically segmented. This is equivalent to a stochastically segmented corpus in which $P_i = 1$ at the points where the word segmenter put a word boundary and $P_i = 0$ at the points where the word segmenter did not put a word boundary.

Raw. The target domain corpus is stochastically segmented. That is, $P_i = \alpha$ at the points where the word segmenter put a word boundary and $P_i = 1-\alpha$ at the points where the word segmenter did not put a word boundary.

Well-done. The target domain corpus is manually segmented and used as a decisively segmented corpus as in **Auto**.

45%-done. The target domain corpus is manually segmented from the beginning to the 281,398-th word (45.00%) and the rest is automatically segmented. The whole corpus is used as a decisively segmented corpus as in **Auto**.

Medium. First, word boundary probabilities are set in the same way as **Raw**. Then at each point where a word in the word list appears, the word boundary probabilities before the word and after the word are set to be 1 and the word boundary probabilities inside of the character sequence of the word are set to be 0. This corresponds to the annotation strategy in which a corpus annotator checks the KWIC of the strings in the word list and marks the occurrences of the strings as the word. The number of checked words was 138,483 (22.13%).

Rare. First, word boundary probabilities are set in the same way as **Raw**. Then at two points for each word where the word in the word list appears, the word boundary probabilities before the word and after the word are set to be 1 and the word boundary probabilities inside of the character sequence of the word are set to be 0. This corresponds to the annotation strategy in which a corpus annotator checks the KWIC of the strings in the word list and marks two occurrences of a string as a word. The number of marked words was 32,643 (5.22%).

We estimated word uni-gram probabilities and word bi-gram probabilities from the target domain corpus with word boundary information obtained by these methods on the vocabulary containing the **Base** model vocabulary words and the words in the word list. Then we interpolated the above LMs with the **Base** model to obtain final LMs.

5.3. Evaluation

Table 2 shows the predictive powers and the character error rates (CER) of the models. The comparison of **Base** with **Auto** and **Raw**, which do not require an error correction work, shows that it is good to gather as many sentences as possible in the target domain. The better usage is **Raw** in which we do not completely rely on the automatic segmentation results.

The predictive powers and the CER of **Raw** are much worse than those of **Well-done**, which requires manual annotation. It follows that it is worthwhile to annotate the raw corpus with word boundary information.

In contrast to sentences that are segmented manually one by one normally, in **Rare** and **Medium** the annotator only checks parts of the sentences. In **Rare**, in which the annotator checks only two occurrences, at least for each word in the word list, only 5.22% of the word occurrences are checked, but the CER is almost the same as that of **45%-done**. The CER of **Medium**, in which the annotator checks all of the occurrences of the strings of the listed words, is almost the same as that of **Well-done**, in which all of the sentences are segmented manually. This result tells us that by using our method we can achieve a comparable accuracy by checking only about 22.13% of the words.

Although the cost to correct boundary information of a word is not necessarily equal in sentence-by-sentence correction and KWIC-based correction, the number of words to be checked in **Rare** is nine times fewer than in **45%-done**. Thus we can say that the total cost in **Rare** is less than that in **45%-done**. In addition, the sentence-by-sentence error correction includes grammatically complicated points to correct properly, even for grammar experts, such as function word sequences, etc. In contrast, the KWIC-based error correction allows an annotator to avoid this kind of difficulty and to concentrate on correcting the positions in which specific words in the application field appear. From the above observations, we conclude that the KWIC-based error correction strategy combined with the notion of an SSC enables us to develop an LM suitable for target applications with a lower cost and in a shorter shorter time.

6. Conclusion

In this paper we compared usages of the raw corpus in the target domain for an LM adaptation assuming that two resources in the target domain are available: a raw corpus and a word list. The comparisons of the predictive powers and the recognition accuracies showed that we can effectively improve the performance of an LM in the taget domain by concentrating the costly error correction effort on the points where the words in the given list occur.

7. References

[1] Masaaki Nagata, "A stochastic Japanese morphological analyzer using a forward-DP backward-A* n-best search algorithm," in *Proc. of the COLING94*, 1994, pp. 201–207.

[2] Dilek Hakkani-Tür, Gokhan Tur, Mazin Rahim, and Giuseppe Riccardi, "Unsupervised and active learning in automatic speech recognition for call classification," in *Proc. of the ICASSP2004*, 2004.

[3] Shinsuke Mori and Daisuke Takuma, "Word n-gram probability estimation from a Japanese raw corpus," in *Proc. of the ICSLP2004*, 2004.

[4] Frank Smadja, "Retrieving collocations from text: Xtract," *Computational Linguistics*, vol. 19, no. 2, pp. 143–177, 1993.

[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*, vol. 18, no. 1, 1992.