Automatic Word Segmentation using Three Types of Dictionaries

Shinsuke MORI
Kyoto University
Yoshidahonmachi, Sakyo-ku, Kyoto, Japan
forest@i.kyoto-u.ac.jp

Hiroki ODA
Shinagawa, Tokyo, Japan
oda@fw.ipsj.or.jp

Abstract

In this paper we propose a new method for automatically segmenting a sentence in Japanese into a word sequence. The main advantage of our method is that the segmenter is, by using a maximum entropy framework, capable of referring to a list of compound words, i.e. word sequences without boundary information. This allows for a higher segmentation accuracy in many real situations where only some electronic dictionaries, whose entries are not consistent with the word segmentation standard, are available. Our method is also capable of exploiting a list of word sequences that are consistent with the word segmentation standard. It allows us to obtain a far greater accuracy gain with low manual annotation cost.

We prepared segmented corpora, a compound word list, and a word sequence list. Then we conducted experiments to compare automatic word segmenters referring to various types of dictionaries. The results showed that the word segmenter we proposed in this paper is capable of exploiting a list of compound words and word sequences to yield a higher accuracy under realistic situations.

1 Introduction

For languages such as Japanese and Chinese in which the words are not delimited by whitespace, word segmentation is the first task in natural language processing (NLP). Almost all NLP systems, whether their approach is empirical or not, depend on the word unit. Thus the accuracy of an automatic word segmenter is extraordinarily important for NLPs in these languages. Because of this background, there have been many attempts at building manually segmented corpora, a set of sentences whose segmentation into words is checked manually (Kurohashi and Nagao, 1998; Maekawa, 2008) and developing automatic word segmenters using empirical approaches (Nagata, 1994; Sproat and Chang, 1996; Uchimotoy et al., 2005, inter alia.).

Recently NLP is applied to a wider and wider variety of domains: automatic translation of patent disclosures, language modeling in a speech recognizer for court reports, information retrieval from medical texts, and so on. The current automatic word segmenters built from a segmented corpus in the general domain, however, are not capable of segmenting sentences accurately in those target domains. The accuracy degradation tends to be serious around words and expressions peculiar to the target domains. To make matters worse, those words and expressions contain the important information in the applications.

To avoid quality degradation in NLP applications, it is necessary to increase the accuracy of an automatic word segmenter in a certain target domain. It is ideal to prepare a corpus in the target domain whose sentences are segmented consistently with the word segmentation standard used for a general domain corpus and build an automatic word segmenter from these segmented corpora. In many real situations, however, dictionaries for human use are the only additionally available resource in the target domain. Their entries are, however, selected without regard to the word segmentation standard of a segmented corpus. Thus the entries are not necessarily consistent with the word segmentation standard. We call this type of resource a compound word list. A compound word can be manually segmented into a word sequence consistent with the standard. We call a resource containing these word sequences a word sequence list. The cost for converting a compound word list into a word sequence list is much

lower than that of preparing a segmented corpus in a target domain. In spite of this reality, no one has proposed an automatic word segmenter which can utilize a word sequence list or a compound word list. as far as the authors know.

In this paper we propose a novel method for segmenting a sentence in Japanese into a word sequence by referring to a compound word list or a word sequence list in addition to a segmented corpus and a word list. By referring to a compound word list in a certain domain, i.e. commercially available dictionaries for humans, the accuracy of an automatic word segmenter is increased without any manual labor. The capability of referring to a word sequence list allows us to obtain a far greater accuracy gain with a much lower manual annotation cost than preparing a segmented corpus in the target domain.

2 Word Segmentation Problem

In this section, first we explain the word segmentation problem in languages without any clear delimiter between words. Then we describe language resources available for building a machine, called an automatic word segmenter, which segments an input sentence into a word sequence.

2.1 Word Segmentation Problem

There are many languages in which the words are not delimited by whitespace such as Japanese and Chinese. For these languages, the first process in NLP is to annotate word boundary information to input sentences. For the readers to understand the setting let us take the following example sentence in English without word boundary information indicated by whitespace as the input of the process.

Input: `Itisworthwhilegoingnowhere`

The word segmentation problem is defined as putting whitespaces at all points between two characters belonging to different words and puttin nothing between characters belonging to the same words according to a predefined word segmentation standard. The example sentence may be segmented as follows.

Output:

```
It is worthwhile going now here
```

Now we can use NLP based on the word unit. As you can see, however, errors in the segmentation process degrade the accuracy of subsequent NLP processes.

丨	:	There is a word boundary.
−	:	There is not a word boundary.
␣	:	There is no information.

Figure 1: Three-valued word boundary notation.

2.2 Segmented Corpus

There are many works on building automatic word segmenters based on the empirical approach (Nagata, 1994; Sproat and Chang, 1996, inter alia.). These automatic word segmenters estimate their parameters from a segmented corpus according to the word segmentation standard. Errors in the learning corpus spread to automatically segmented sentences and severely degrade the accuracy of the subsequent NLP applications. Thus the quality of the segmented corpus is very important. In addition, it is preferable that the domain of the corpus is the same as the target domain of the subsequent NLP applications. It is, however, very costly to prepare a correctly segmented corpus. The annotator must know well both linguistics and the target domain, as the written word segmentation standard is not sufficient to cover all the linguistic phenomena in the target domain. It took, for example, a skillful annotator for two weeks to prepare 5,000 segmented sentences.

2.3 Three Types of Dictionaries

Given a word segmentation standard, dictionaries used in the word segmentation task are categorized into three types. Below we explain these three types with examples denoted in the three-valued notation shown in Figure 1.

- **single word list:**
  This list contains only character sequences consistent with the word segmentation standard. That is to say, in a certain context, there are word boundaries at the left side of the leftmost character and the right side of the rightmost character and there is no word boundary inside the sequence. For example, in the three-valued notation,

  丨言−語丨 (language)

  is a single word.
- **word sequence list:**
  This list contains only word sequences consistent with the word segmentation standard. That is to say, the word boundary information matches exactly with a word sequence in a certain context. For example, in the three-valued notation,

|計-算|言-語|学| (computational linguistics)

is a word sequence.

- **compound word list:**

  This list contains character sequences that are concatenations of words without word boundary information. That is to say, there are word boundaries at the left side of the leftmost character and the right side of the rightmost character but there is no word boundary information inside the sequence. For example, in three-valued format,

  |計␣算␣言␣語␣学|

  is a compound word.

There are many commercially available dictionaries. All of those entries are selected without regard to the word segmentation standard of a segmented corpus. Entries contained in most dictionaries useful for domain adaptation are, however, technical terms and proper names, both edges of which tend to be word boundaries according to the word segmentation standard. Thus almost all available dictionaries fall into the third category. That is to say, their entries are compound words.

A compound word is converted into a word sequence consistent with the standard. This requires manual annotation of word boundary information at all points between characters inside the compound word. But the cost is much lower than that needed to prepare a segmented corpus in a target domain. Therefore it is worth devising an automatic word segmenter which can refer to a compound word list or a word sequence list.

3 Automatic Word Segmenter

In this section, we explain our new method of segmenting a sentence into a word sequence by referring to a (partially) segmented corpus and three types of dictionaries explained in the previous section.

3.1 Point-wise Maximum Entropy Method

The word segmentation can be considered as a problem to predict whether or not a word boundary exists after each character. That is, given an input sentence $\boldsymbol{x} = x_1 x_2 \cdots x_m$, the problem to be solved is to add a position-of-character (POC) tag t_i to each character x_i indicating the likelihood that the character is at the end of a word (Xue, 2003). The POC tag set consists of **B** and **N**. The tag **B** indicates that a word boundary ("|" in three-valued notation) exists after the character and **N** indicates that a word boundary does not exist after the character, that is, there is "-" after the character in three-valued notation. The POC tag of each character is predicted by a maximum entropy (ME) model (Adam L. Berger, 1996) whose parameters are estimated from sentences correctly annotated word boundary information[1]. The POC tag of a character in an input sentence is decided to be the tag that has the highest probability. That is, if the probability given by the ME model that a word boundary exists after the character is higher than the probability that a word boundary does not exist after the character as follows

$$P_{ME}(\mathbf{B}|i,\boldsymbol{x}) > P_{ME}(\mathbf{N}|i,\boldsymbol{x}), \tag{1}$$

the ME model gives the tag **B** to the character x_i and **N** otherwise. By deciding the POC tags for all characters in an input sentence, we have the sentence segmented into words.

Our ME-based word segmenter is formalized as follows. First a sentence $\boldsymbol{x} = x_1 x_2 \cdots x_m$ with word boundary information (a segmented sentence) is regarded as a character sequence with a POC tag t_i between two characters x_i and x_{i+1}. Then we convert the segmented sentence into a set of examples as follows:

$$S = \{(t_i,\ f_{i,1}(\boldsymbol{x}),\ f_{i,2}(\boldsymbol{x}), \cdots) \mid \forall 1 \leq i \leq m-1\},$$

where $f_{i,j}(\boldsymbol{x})$ is a feature derived from the sentence $\boldsymbol{x}$, which we explain in the following subsections in detail.

3.2 Baseline Features to be Referred to

We propose, as possible features of the ME model, all of the character and character type n-grams ($n \leq 3$) contained by x_{i-1}^{i+2} to estimate the probability in which a word boundary exists after the character x_i. In addition we used the following modifications.

- The beginning character of the character n-grams and character type n-grams is extended by adding a flag if the character type[2] is the

[1] As in Tsuboi et al. (2008), CRF (conditional random fields) can be used. Its computational cost is, however, much more higher than our point-wise ME model since CRF model a segmented sentence as a sequence. This cost is problematic when we estimate CRF from a partially annotated corpus whose annotated points are very sparse.

[2] There are 6 character types including Latin characters, Arabic digits, symbols and *katakana* for imported words.

Table 1: Segmented corpora.

domain	usage	#sentences	#words	#chars
general	learning	27,935	626,700	878,089
newspaper	learning	1,002	29,038	43,695
general	test	3,447	77,990	109,064
newspaper	test	9,023	263,427	398,570

Table 2: Dictionaries.

type	#entries	#words	#chars
single word	145,310.0	145,310.0	430,797.0
word sequence	17,099.5	27,465.3	53,364.4
compound word	19,697.1	–	59,868.4

same as that of the previous character. The ending character of the *n*-grams is also extended by adding a flag if the character type is the same as that of the next character.

- The character *n*-grams and character type *n*-grams are annotated with the offset from the character boundary under consideration.

3.3 Using the Dictionaries

In order for our word segmenter to exploit three types of dictionaries, we propose to add two types of features referring to the entries of the dictionaries. The first ones are nine features that says if any of the character *n*-grams appear in the single word list. The second ones are to check if the following conditions are satisfied or not:

- The sequence $x_{i+1}x_{i+2}$ after the position i matches the beginning of any single word, word sequence, or compound word.
- The sequence $x_{i-1}x_i$ before the position i matches the end of any single word, word sequence, or compound word.
- The sequence $x_{i-1}x_i \,|\, x_{i+1}x_{i+2}$ matches any word sequence.
- The sequence $x_{i-1}x_i$-$x_{i+1}x_{i+2}$ matches any word or word sequence.

4 Evaluation

As an evaluation of our automatic word segmenters, we measured the word segmentation accuracies of the segmenters on a test corpus. In this section we show the results and evaluate our new method.

4.1 Conditions of the Experiments

The test corpus in the target domain we used in the experiments consists of sentences extracted from a Japanese economic newspaper (*Nikkei* newspaper). We prepared two training corpora: one is in the general domain and the other is in the same domain as the test corpus (see Table 1). The general domain corpus is composed of the sentences in BCCWJ (Maekawa, 2008) (13,181 sentences) and example sentences in a dictionary of daily conversation (14,754 sentences). Each sentence in the corpora is segmented into words manually. We conducted 9-fold cross validation. In other words, the test corpus is divided into nine parts and we conducted nine experiments in which eight parts are used to filter compound words or word sequences and the remaining part is used for the test.

We prepared three types of dictionaries. The first one is a single word list called UniDic, whose entries are carefully checked to be consistent with the word segmentation standard of the corpora. The second one is a compound word list, whose entries were extracted from commercially available dictionaries in electronic form containing mainly technical terms and proper names. In the experiments we selected compound words appearing as character sequences in the eight partial corpora other than the one used for the test. So they are expected to be consistent with the word segmentation standard at both edges, but there is no guarantee. The third one is a word sequence. To make this list, the compound words were manually segmented into word sequences[3]. Thus it is guaranteed that they are consistent with the word segmentation standard. In the experiments we selected word sequences appearing in the eight partial corpora other than the one used for the test. Ta-

[3]After manual checking, some compound words turned out to be single words or incorrect character sequences.

Table 3: Word segmentation accuracy on the corpus in the general domain.

ID	learning resource	boundary accuracy	precision	recall	sentence accuracy
B	baseline	98.82%	97.87%	97.86%	77.22%
C	+ compound word	98.86%	97.93%	97.91%	77.62%
S	+ word sequence	98.97%	98.08%	98.16%	78.97%
W	+ single word	98.99%	98.13%	98.13%	79.12%

Table 4: Word segmentation accuracy on the corpus in the target domain.

ID	learning resource	boundary accuracy	precision	recall	sentence accuracy
B	baseline	98.07%	96.28%	96.28%	55.64%
C	+ compound word	98.20%	96.44%	96.50%	56.95%
S	+ word sequence	98.72%	97.39%	97.47%	65.54%
W	+ single word	98.43%	96.93%	96.88%	60.81%

ble 2 shows some features of the dictionaries. This table shows that the average number of characters of the compound words and that of the word sequences are 3.04 and 3.12 respectively, which are longer than the average word length in the general corpus (1.40) and that in the newspaper corpus (1.50). The word sequences are composed of 1.61 words in average.

The parameters of the word segmenters were estimated from the training corpora and dictionaries, and it was tested on the test corpus.

4.2 Evaluation Criterion

The criterion we used for automatic word segmentation is precision, recall, boundary accuracy, and sentence accuracy. Given an automatic word segmentation result (AWS) and the correct word sequence (COR), we explain how to calculate them using the following character sequences annotated with word boundary information as examples.

AWS: `It is worthwhile going now here`

COR: `It is worthwhile going nowhere`

Boundary accuracy is the number of the character boundaries whose word boundary information is correct divided by the number of the character boundaries. In the example, there are 26 characters, so the number of character boundaries is 25. And there is one character boundary whose word boundary information is incorrect. Thus the boundary accuracy is 24/25. Sentence accuracy is the ratio of the sentences whose all segmentations match with the correct word sequence completely. Precision and recall are calculated as follows. Let N_{COR} be the number of words in the correct word sequence, N_{AWS} be that of the automatic word segmentation result, and N_{LCS} be that of the longest common subsequence (LCS) in word of the correct word sequence and the automatic word segmentation result, so the precision as N_{LCS}/N_{AWS} and the recall is defined as N_{LCS}/N_{COR}. In the example case, the LCS is the underlined word sequence and $N_{LCS} = 4$. There are six words in the automatic word segmentation result ($N_{AWS} = 6$) and there are five words in the correct word sequence ($N_{COR} = 5$). Thus the precision is $N_{LCS}/N_{AWS} = 4/6$ and the recall is $N_{LCS}/N_{COR} = 4/5$.

4.3 Evaluation

In order to clarify the difference in the dictionaries to be used, we built four automatic word segmenters compared their accuracies. The first one ***B*** refers to no dictionary, which is the baseline. The others refers to one of the three types of dictionaries as follows:

***C*:** baseline with the compound word list.

***S*:** baseline with the word sequence list.

***W*:** baseline with the single word list.

Table 3 and 4 show the accuracies of the segmenters in the general and in the target domain respectively. The accuracy of the baseline segmenter ***B*** in the general domain is sufficiently high. In the target domain, however, we observe a severe decrease in accuracy. By referring to the compound word list ***C***, the accuracy increases in the target domain. Considering the fact that the compound word list was automatically extracted from some

machine readable dictionaries and no manual annotation is required to prepare it, we can say that our proposal to use a compound word list is efficient.

The accuracies of $\boldsymbol{C}$ in the both domain are lower than that of $\boldsymbol{W}$, which refers to the single word list supplied along with BCCWJ corpus. The single word list is, however, manually prepared by the linguists who are very familiar with the word segmentation standard. The total number of characters of the words in the list is 430,797 (see Table 2), which corresponds to approximately 9,879 sentences in the target domain (see Table 1). Thus the improvement resulting from using the single word list for general usage is very costly.

The accuracy of the word segmenter referring to the word sequence list $\boldsymbol{S}$ in the target domain is higher than the segmenter referring to the compound word list $\boldsymbol{C}$. Thus it can be said that far more improvement is achieved by annotating the compound word with word boundary information appearing in sentences in the target domain.

The accuracy in the target domain of $\boldsymbol{S}$ is much higher than that of $\boldsymbol{W}$. The total number of characters of the compound words for manual annotation is 59,868 (see Table 2) which corresponds to approximately 1,373 sentences of newspaper articles. The annotation cost is far less than that required to prepare the single word list. In addition, since the compound words are mainly technical terms and proper names, annotators only need to know the word segmentation standard related to nouns as well as the domain knowledge. This fact makes it easier to find annotators.

From above observations, we can conclude that the best strategy to have an automatic word segmenter in a target domain under realistic situations is 1) to gather dictionary entries appearing in raw texts in the target domain, form a compound word list, and use our automatic word segmenter referring to them, 2) to annotate the compound words with word boundary information if possible.

5 Conclusion

In this paper we proposed a new method for segmenting a sentence in Japanese into a word sequence automatically. The main advantage of our method is that the segmenter is capable of referring to a list of compound words, i.e. word sequences without boundary information in them. With these characteristics, we can enjoy a higher segmentation accuracy in many real situations where only some ordinary dictionaries, whose entries are not consistent with the word segmentation standard, are available. Our method is also capable of exploiting a list of word sequences that are consistent with the word segmentation standard. It allows us to obtain a far greater accuracy gain with low manual annotation cost.

We conducted experiments to compare automatic word segmenters referring to various types of dictionaries. The results showed that the word segmenter we proposed in this paper is capable of exploit a list of compound words and word sequences to yield a higher accuracy under realistic situations.

Our word segmenter is general enough that it is applicable to other languages which require an automatic word segmenter, such as Chinese, etc., as the first step of natural language processing.

References

Vincent J. Della Pietra Adam L. Berger, Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1).

Sadao Kurohashi and Makoto Nagao. 1998. Building a japanese parsed corpus while improving the parsing system. pages 719–724.

Kikuo Maekawa. 2008. Balanced corpus of contemporary written japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*, pages 101–102.

Masaaki Nagata. 1994. A stochastic japanese morphological analyzer using a forward-dp backward-a* n-best search algorithm. In *Proc. of the COLING94*, pages 201–207.

Richard Sproat and Chilin Shih William Gale Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for chinese. *Computational Linguistics*, 22(3):377–404.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proc. of the COLING08*.

Kiyotaka Uchimotoy, Satoshi Sekine, and Hitoshi Isahara. 2005. The unknown word problem: a morphological analysis of japanese using maximum entropy aided by a dictionary. In *EMNLP*, pages 265–272.

N. Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese*, 8(1):29–48.