Segmenting a Sentence into Morphemes Using Statistic Information between Words

Shiho Nobesawa
Junya Tsutsumi, Tomoaki Nitta, Kotaro Ono, Sun Da Jiang, Masakazu Nakanishi

Nakanishi Laboratory
Faculty of Science and Technology, Keio University

Abstract

This paper is on dividing non-separated language sentences (whose words are not separated from each other with a space or other separaters) into morphemes using statistical information, not grammatical information which is often used in NLP. In this paper we describe our method and experimental result on Japanese and Chinese sentences. As will be seen in the body of this paper, the result shows that this system is efficient for most of the sentences.

1 Introduction and Motivation

An English sentence has several words and those words are separated with a space. It is easy to divide an English sentence into words. However a Japanese sentence needs parsing if you want to pick up the words in the sentence. This paper is on dividing non-separated language sentences into words(*morphemes*) without using any grammatical information. Instead, this system uses the statistic information between morphemes to select best ways of segmenting sentences in non-separated languages.

Thinking about segmenting a sentence into pieces, it is not very hard to divide a sentence using a certain dictionary for that. The problem is how to decide which 'segmentation' the best answer is. For example, there must be several ways of segmenting a Japanese sentence written in Hiragana(Japanese alphabet). Maybe a lot more than 'several'. So, to make the segmenting system useful, we have to consider how to pick up the right segmented sentences from all the possible seems-like-segmented sentences.

This system is to use statistical information between morphemes to see how 'sentence-like'(how 'likely' to happen as a sentence) the segmented string is. To get the statistical association between words, mutual information(MI) comes to be one of the most interesting method. In this paper MI is used to calculate the relationship between words found in the given sentence. A corpus of sentences is used to gain the MI.

To implement this method, we implemented a system MSS(Morphological Segmentation using Statistical information). What MSS does is to find the best way of segmenting a non-separated language sentence into morphemes without depending on grammatical information. We can apply this system to many languages.

2 Morphological Analysis

2.1 What a Morphological Analysis Is

A morpheme is the smallest unit of a string of characters which has a certain linguistic meaning itself. It includes both content words and function words. In this paper the definition of a morpheme is a string of characters which is looked up in the dictionary.

Morphological analysis is to:

1) recognize the smallest units making up the given sentence

   if the sentence is of a non-separated language, divide the sentence into morphemes (automatic segmentation), and

2) check the morphemes whether they are the right units to make up the sentence.

2.2 Segmenting Methods

We have some ways to segment a non-separated sentence into meaningful morphemes. These three methods explained below are the most popular ones to segment Japanese sentences.

- **The longest-segment method:**
  Read the given sentence from left to right and cut it with longest possible segment. For example, if we get 'isheold' first we look for segments which uses the first few letters in it,'i' and 'is'. It is obvious that 'is' is longer than 'i', so the system takes 'is' as the segment. Then it tries the same method to find the segments in 'heold' and finds 'he' and 'old'.

- **The least-*bunsetsu* segmenting method:**
  Get all the possible segmentations of the input sentence and choose the segmentation(s) which has least *bunsetsu* in it. This method is to segment Japanese sentences, which have content words and function words together in one *bunsetsu* most of the time. This method helps not to cut a sentence into too small meaningless pieces.

- **Letter-type segmenting method:**
  In Japanese language we have three kinds of letters called Hiragana, Katakana and Kanji. This

method divides a Japanese sentence into meaningful segments checking the type of letters.

2.3 The Necessity of Morphological Analysis

When we translate an English sentence into another language, the easiest way is to change the words in the sentence into the corresponded words in the target language. It is not a very hard job. All we have to do is to look up the words in the dictionary. However when it comes to a non-separated language, it is not as simple. An non-separated language does not show the segments included in a sentence. For example, a Japanese sentence does not have any space between words. A Japanese-speaking person can divide a Japanese sentence into words very easily, however, without any knowledge in Japanese it is impossible. When we want a machine to translate an non-separated language into another language, first we need to segment the given sentence into words.

Japanese is not the only language which needs the morphological segmentation. For example, Chinese and Korean are non-separated too. We can apply this MSS system to those languages too, with very simple preparation. We do not have to change the system, just prepare the corpus for the purpose.

2.4 Problems of Morphological Analysis

The biggest problems through the segmentation of an non-separated language sentence are the ambiguity and unknown words.

For example,

にわにはにわとりがいる。
niwanihaniwatorigairu.

庭	には	鶏	が	いる	。
niwa	niha	niwatori	ga	iru	.

A cock is in the yard.

庭	には	二羽	鳥	が	いる	。
niwa	niha	niwa	tori	ga	iru	.

Two birds are in the yard.

庭	に	埴輪	取り	が	居る	。
niwa	ni	haniwa	tori	ga	iru	.

A clay-figure robber is in the yard.

Those sentences are all made of same strings but the included morphemes are different. With different segments a sentence can have several meanings. Japanese has three types of letters: Hiragana, Katakana and Kanji. Hiragana and Katakana are both phonetic symbols, and each Kanji letters has its own meanings. We can put several Kanji letters to one Hiragana word. This makes morphological analysis of Japanese sentence very difficult. A Japanese sentence can have more than one morphological segmentation and it is not easy to figure out which one makes sense. Even two or more segmentation can be 'correct' for one sentence.

To get the right segmentation of a sentence one may need not only morphological analysis but also semantic analysis or grammatical parsing. In this paper no grammatical information is used and MI between morphemes becomes the key to solve this problem.

To deal with unknown words is a big problem in natural language processing(NLP) too. To recognize unknown segments in the sentences, we have to discuss the likelihood of the unknown segment being a linguistic word. In this paper unknown words are not acceptable as a 'morpheme'. We define that 'morpheme' is a string of characters which is registered in the dictionary.

3 Calculating the Scores of Sentences

3.1 Scores of Sentences

When the system searches the ways to divide a sentence into morphemes, more than one segmentation come out most of the time. What we want is one (or more) 'correct' segmentation and we do not need any other possibilities. If there are many ways of segmenting, we need to select the best one of them. For that purpose the system introduced the 'scores of sentences'.

3.2 Mutual Information

A mutual information(MI)[1][2][3] is the information of the association of several things. When it comes to NLP, MI is used to see the relationship between two (or more) certain words.

The expression below shows the definition of the MI for NLP:

$$MI(w_1; w_2) = log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \quad (1)$$

w_i	:	a word
$P(w_i)$	:	the probability w_i appears in a corpus
$P(w_1, w_2)$	:	the probability w_1 and w_2 comes out together in a corpus

This expression means that when w_1 and w_2 has a strong association between them, $P(w_1)P(w_2) \ll P(w_1, w_2)$ i.e. $MI(w_1, w_2) \gg 0$. When w_1 and w_2 do not have any special association, $P(w_1)P(w_2) \approx P(w_1, w_2)$ i.e. $MI(w_1, w_2) \approx 0$. And when w_1 and w_2 come out together very rarely, $P(w_1)P(w_2) \gg P(w_1, w_2)$ i.e. $MI(w_1, w_2) \ll 0$.

3.3 Calculating the Score of a Sentence

Using the words in the given dictionary, it is easy to make up a 'sentence'. However, it is hard to consider whether the 'sentence' is a correct one or not. The meaning of 'correct sentence' is a sentence which makes sense. For example, 'I am Tom.' can make sense, however, 'Green the adzabak are the a ran four.' is hardly took as a meaningful sentence. The score is to show how 'sentence-like' the given string of morphemes is. Segmenting a non-separated language sentence, we often get a lot of meaningless strings of morphemes. To pick up seems-like-meaningful strings from the segmentations, we use MI.

Actually what we use in the calculation is not the real MI described in section 3.2. The MI expression in section 3.2 introduced the bigrams. A bigram is a possibility of having two certain words together in a corpus, as you see in the expression(1). Instead of the bigram we use a new method named d-bigram here in this paper[3].

3.3.1 D-bigram

The idea of bigrams and trigrams are often used in the studies on NLP. A bigram is the information of the association between two certain words and a trigram is the information among three. We use a new idea named d-bigram in this paper[3]. A d-bigram is the possibility that two words w_1 and w_2 come out together at a distance of d words in a corpus. For example, if we get 'he is Tom' as input sentence, we have three d-bigram data:

('he' 'is' 1)
('is' 'Tom' 1)
('he' 'Tom' 2)

('he' 'is' 1) means the information of the association of the two words 'he' and 'is' appear at the distance of 1 word in the corpus.

3.4 Calculation

The expression to calculate the scores between two words is[3]:

$$MI_d(w_1, w_2, d) = log\frac{P(w_1, w_2, d)}{P(w_1)P(w_2)} \qquad (2)$$

w_i	:	a word
d	:	distance of the two words w_1 and w_2
$P(w_i)$	:	the possibility the word w_i appears in the corpus
$P(w_1, w_2, d)$	:	the possibility w_1 and w_2 come out d words away from each other in the corpus

As the value of MI_d gets bigger, the more those words have the association. And the score of a sentence is calculated with these MI_d data(expression(2)). The definition of the sentence score is[1]:

$$I_d(W) = \sum_{i=0}^{n}\sum_{d=1}^{m}\frac{MI_d(w_i, w_{i+d}, d)}{d^2} \qquad (3)$$

d	:	distance of the two words
m	:	distance limit
n	:	the number of words in the sentence
W	:	a sentence
w_i	:	The i-th morpheme in the sentence W

This expression(3) calculates the scores with the algorithm below:

1) Calculate MI_d of every pair of words included in the given sentence.

2) Give a certain weight according to the distance d to all those MI_d.

3) Sum up those $\frac{MI_d}{d^2}$. The sum is the score of the sentence.

Church and Hanks said in their paper[1] that the information between two remote words has less meaning in a sentence when it comes to the semantic analysis. According to the idea we put d^2 in the expression so that nearer pair can be more effective in calculating the score of the sentence.

4 The System MSS

4.1 Overview

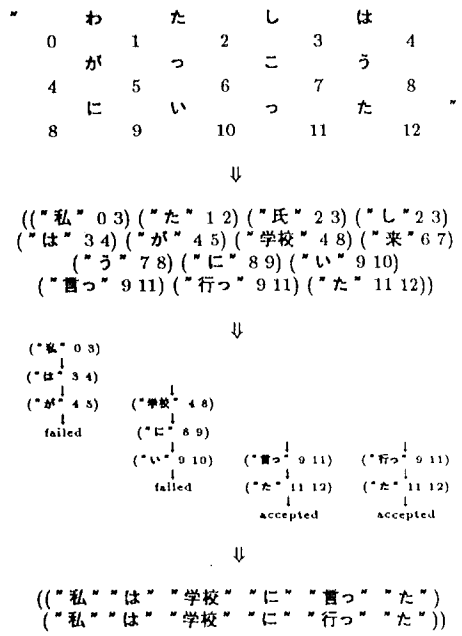
MSS takes a Hiragana sentence as its input. First, MSS picks up the morphemes found in the given sentence with checking the dictionary. The system reads the sentence from left to right, cutting out every possibility. Each segment of the sentence is looked up in the dictionary and if it is found in the dictionary the system recognize the segment as a morpheme. Those morphemes are replaced by its corresponded Kanji(or Hiragana, Katakana or mixed) morpheme(s). As it is told in section 2.4, a Hiragana morpheme can have several corresponded Kanji (or other lettered) morphemes. In that case all the segments corresponded to the found Hiragana morpheme are memorized as morphemes found in the sentence. All the found morphemes are numbered by its position in the sentence.

After picking up all the morphemes in the sentence the system tries to put them together and brings them up back to sentence(table 1).

Input a Hiragana sentence.
⇓
Cut out the morphemes.
⇓
Make up sentences with the morphemes.
⇓
Calculate the score of sentences
using the mutual information.
⇓
Compare the scores of all the made-up sentences
and get the best-marked one
as the most 'sentence-like' sentence.

Then the system compares those sentences made up with found morphemes and sees which one is the

Table 1: MSS example



most 'sentence-like'. For that purpose this system calculate the score of likelihood of each sentences(section 3.4).

4.2 The Corpus

A corpus is a set of sentences. These sentences are of target language. For example, when we apply this system to Japanese morphological analysis we need a corpus of Japanese sentences which are already segmented.

The corpus prepared for the paper is the translation of English textbooks for Japanese junior high school students. The reason why we selected junior high school textbooks is that the sentences in the textbooks are simple and do not include too many words. This is a good environment for evaluating this system.

4.3 The Dictionary

The dictionary for MSS is made of two part. One is the heading words and the other is the morphemes corresponded to the headings. There may be more than one morphemes attached to one heading word. The second part which has morphemes is of type list, so that it can have several morphemes.

Japanese : ("いっ" ("行っ" "言っ"))
heading word morphemes

Chinese : ("jing" ("京" "静"))
heading word morphemes

5 Results

Implement MSS to all input sentences and get the score of each segmentation. After getting the list of segmentations, look for the 'correct' segmented-sentence and see where in the list the right one is. The data shows the scores the 'correct' segmentations got(table 2).

Table 2: Experiment in Japanese

corpus	:	about 630 Japanese sentences (with three kinds of letters mixed)
dictionary	:	about 1500 heading words (includes morphemes not in the corpus)
input	:	non-segmented Japanese sentences using Hiragana only
number of input sentence	:	about 100 each
distance limit	:	5

	best score	~ 2nd best	~ 3rd best
α	99 %	100 %	100 %
β	100 %	100 %	100 %
γ	100 %	100 %	100 %
δ	95 %	98 %	98 %
ε	80 %	90 %	95 %

α : the very sentences in the corpus
β : replaced one morpheme in the sentence (the buried morpheme is in the corpus)
γ : replaced one morpheme in the sentence (the buried morpheme is not in the corpus)
δ : sentences not in the corpus (the morphemes are all in the corpus)
ε : sentences not in the corpus (include morphemes not in the corpus)

5.1 Experiment in Japanese

According to the experimental results(table 2), it is obvious that MSS is very useful. The table 2 shows that most of the sentences, no matter whether the sentences are in the corpus or not, are segmented correctly. We find the right segmentation getting the best score in the list of possible segmentations. α is the data when the input sentences are in corpus. That is, all the 'correct' morphemes have association between each other. That have a strong effect in calculating the scores of sentences. The condition is almost same for β and γ. Though the sentence has one word replaced, all other words in the sentence have relationship between them. The sentences in γ include one word which is not in the corpus, but still the 'correct' sentence can get the best score among the possibilities. We can say that the data α, β and γ are very successful.

However, we should remember that not all the sentences in the given corpus would get the best score through the list. MSS does not check the corpus itself when it calculate the score. It just use the MI_d, the essential information of the corpus. That is, whether the input sentence is written in the corpus or not does not make any effect in calculating scores directly. However, since MSS uses MI_d to calculate the scores, the fact that every two morphemes in the sentence have connection between them raises the score higher.

When it comes to the sentences which are not in corpus themselves, the ratio that the 'correct' sentence get the best score gets down (see table 2, data δ, ε).

The sentences of δ and ε are not found in the corpus. Even some sentences which are of spoken language and not grammatically correct are included in the input sentences. It can be said that those δ and ε sentences are nearer to the real world of Japanese language. For δ sentences we used only morphemes which are in the corpus. That means that all the morphemes used in the δ sentences have their own MI_d. And ε sentences have both morphemes in the corpus and the ones not in the corpus. The morphemes which are not in the corpus do not have any MI_d. Table 2 shows that MSS gets quite good result even though the input sentences are not in the corpus. MSS do not take the necessary information directly from the corpus and it uses the MI_d instead. This method makes the information generalized and this is the reason why δ and ε can get good results too. MI_d comes to be the key to use the effect of the MI between morphemes indirectly so that we can put the information of the association between morphemes to practical use. This is what we expected and MSS works successfully at this point.

5.2 The Corpus

In this paper we used the translation of English textbooks for Japanese junior high school students. Primary textbooks are kind of a closed world which have limited words in it and the included sentences are mostly in some fixed styles, in good grammar. The corpus we used in this paper has about 630 sentences which have three types of Japanese letters all mixed. This corpus is too small to take as a model of the real world, however, for this paper it is big enough. Actually, the results of this paper shows that this system works efficiently even though the corpus is small.

The dictionary and the statistical information are got from the given corpus. So, the experimental result totally depends on the corpus. That is, selecting which corpus to take to implement, we can use this system in many purposes(section 5.5).

5.3 Comparison with the Other Methods

It is not easy to compare this system with other segmenting methods. We compare with the least-*bunsetsu* method here in this paper.

The least-*bunsetsu* method segment the given sentences into morphemes and find the segmentations with least *bunsetsu*. This method makes all the segmentation first and selects the seems-like-best segmentations. This is the same way MSS does. The difference is that the least-*bunsetsu* method checkes the number of the *bunsetsu* instead of calculating the scores of sentences.

Let us think about implementing a sentence the morphemes are not in the dictionary. That means that the morphemes do not have any statistical informations between them. In this situation MSS can not use statistical informations to get the scores. Of course MSS caliculate the scores of sentences according to the statistical informations between given morphemes. However, all the MI_d say that they have no association between the morphemes. When there is no possibility that the two morphemes appears together in the corpus, we give a minus score as the MI_d value, so, as the result, with more morphemes the score of the sentence gets lower. That is, the segmentation which has less segments in it gets better scores. Now compare it with the least-*bunsetsu* method. With using MSS the least-morpheme segmentations are selected as the good answer. That is the same way the least-*bunsetsu* method selects the best one. This means that MSS and the least-*bunsetsu* method have the same efficiency when it comes to the sentences which morphemes are not in the corpus. It is obvious that when the sentence has morphemes in the corpus the efficiency of this system gets much higher(table 2).

Now it is proved that MSS is, at least, as efficient as the least-*bunsetsu* method, no matter what sentence it takes. We show a data which describes this(table 3).

Table 3 is a good example of the case when the input sentence has few morphemes which are in the corpus. This data shows that in this situation there is an outstanding relation between the number of morphemes and the scores of the segmented sentences. This example(table 3) has an ambiguity how to segment the sentence using the registered morphemes, and all the morphemes which causes the ambiguity are not in the given corpus. Those morphemes not in the corpus do not have any statistical information between them and we have no way to select which is better. So, the scores of sentences are up to the length of the segmented sentence, that is, the number how many morphemes the sentence has. The segmented sentence which has least segments gets the best score, since MSS gives a minus score for unknown association between morphemes. That means that with more segments in the sentence the score gets lower. This sit-

Table 3: MSS and The least-*bunsetsu* method

input : a non-segmented
Japanese Hiragana sentence
not in the corpus
all unknown morphemes in the sentence
are registered in the dictionary
(some morphemes in the corpus
are included)

"	すもも	も	もも	も	もも	の	うち	"
"	sumomo	mo	momo	mo	momo	no	uchi	"
the number of the morphemes		6	7	8	9		10	
the scores of the sentences		-65.0	-79.6	-94.3	-108.9		-123.5	
the number of the segmented sentences		5	20	21	8		1	
the 'correct' segmentation			★					
MSS		○						
the least-*bunsetsu* method		○						

morphemes included in the corpus	:	" の "	" 物 "
	:	" no "	" mono "
morphemes not included in the corpus	:	" 内 "	" 住も "
	:	" uchi "	" sumo "
		" 李 "	" も "
		" sumomo "	" mo "
		" 桃 "	
		" momo "	

uation is resemble to the way how the least-*bunsetsu* method selects the answer.

5.4 Experiment in Chinese

The theme of this paper is to segment non-separated language sentences into morphemes. In this paper we described on segmentation of Japanese non-segmented sentences only but we are working on Chinese sentences too. This MSS is not for Japanese only. It can be used for other non-separated languages too. To implement for other languages, we just need to prepare the corpus for that and make up the dictionary from it.

Here is the example of implementing MSS for Chinese language(table 4). The input is a string of characters which shows the pronounciations of a Chinese sentence. MSS changes it into Chinese character senteces, segmenting the given string.

5.5 Changing the Corpus

To implement this MSS system, we only need a corpus. The dictionary is made from the corpus. This

Table 4: Experiment in Chinese

input : nashiyizhangditu.

correct answer	output sentences	scores
★	那 是 一張 地図 .	15.04735
	那 視 一張 地図 .	-14.80836
	那 使 一張 地図 .	-14.80836

gives MSS system a lot of usages and posibilities. Most of the NLP systems need grammatical informations, and it is very hard to make up a certain grammatical rule to use in a NLP. The corpus MSS needs to implement is very easy to get. As it is described in the previous section, a corpus is a set of real sentences. We can use MSS in other languages or in other purposes just getting a certain corpus for that and making up a dictionary from the corpus. That is, MSS is available in many purposes with very simple, easy preparation.

6 Conclusion

This paper shows that this automatic segmenting system MSS is quite efficient for segmentation of non-separated language sentences. MSS do not use any grammatical information to divide input sentences. Instead, MSS uses MI between morphemes included in the input sentence to select the best segmentation(s) from all the possibilities. According to the results of the experiments, MSS can segment almost all the sentences 'correctly'. This is such a remarkable result. When it comes to the sentences which are not in the corpus the ratio of selecting the right segmentation as the best answer get a little bit lower, however, the result is considerably good enough.

The result shows that using MI_d between morphemes is a very effective method of selecting 'correct' sentences, and this means a lot in NLP.

References

[1] Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. Parsing, Word Associations and Typical Predicate-Argument Relations. *International Parsing Workshop*, 1989.

[2] Frank Smadja. How to compile a bilingual collocational lexicon automatically. *Statistically-based Natural Language Programming Techniques*, pages 57–63, 1992.

[3] Junya Tsutsumi, Tomoaki Nitta, Kotaro Ono, and Shiho Nobesawa. A Multi-Lingual Translation System Based on A Statistical Model(written in Japanese). *JSAI Technical report, SIG-PPAI-9302-2*, pages 7–12, 1993.

[4] David M.Magerman and Mitchell P.Marcus. Parsing a Natural Language Using Mutual Information Statistics. *AAAI*, 1990.

[5] P.Brown, J.Cocke, S.Della Pietra, V.Della Pietra, F.Jelinek, R.Mercer, and P.Roossin. A Statistical Approach to Language Translation. *Proc. of COLING-88*, pages 71–76, 1989.