Detecting sentence boundaries in Japanese speech transcriptions using a morphological analyzer

Sachie Tajima
Interdisciplinary Graduate School of Science and Engineering
Tokyo Institute of Technology, Tokyo
tajima@lr.pi.titech.ac.jp

Hidetsugu Nanba
Graduate School of Information Sciences
Hiroshima City University, Hiroshima
nanba@its.hiroshima-cu.ac.jp

Manabu Okumura
Precision and Intelligence Laboratory
Tokyo Institute of Technology, Tokyo
oku@pi.titech.ac.jp

Abstract

We present a method to automatically detect sentence boundaries(SBs) in Japanese speech transcriptions. Our method uses a Japanese morphological analyzer that is based on a cost calculation and selects as the best result the one with the minimum cost. The idea behind using a morphological analyzer to identify candidates for SBs is that the analyzer outputs lower costs for better sequences of morphemes. After the candidate SBs have been identified, the unsuitable candidates are deleted by using lexical information acquired from the training corpus. Our method had a 77.24% precision, 88.00% recall, and 0.8277 F-Measure, for a corpus consisting of lecture speech transcriptions in which the SBs are not given.

1 Introduction

Textual information is semi-permanent and is easier to use than speech information, which is only accessible sequentially when it is recorded. Therefore, for many purposes, it is convenient to transcribe speech information into textual information. Two methods are currently used for making transcriptions, manual transcription and automatic speech recognition (ASR).

Speech information is generally spoken language. Spoken language is quite different from written language used to describe textual information. For instance, in written language a 'sentence' can be a linguistic unit, but in spoken language, there exists no linguistic unit like 'sentence.' Consequently, SBs are not specified in manual or ASR speech transcriptions.

However, if SBs can be added to transcribed texts, the texts would be much more usable. Furthermore, SBs are required by many NLP technologies. For instance, Japanese morphological analyzers and syntactic analyzers typically regard their input as a sentence.

Since Japanese morphological analyzers regard their input as a sentence, they tend to output incorrect results when the input is a speech transcription without SBs. For instance, if the character string '...tokaitearimasushidekonomoji...' is inputted to the morphological analyzer Chasen (Matsumoto et al., 2002), the output would be '... / to / kai / te / arima / sushi / de / kono / moji / ...' , where '/' indicates the word boundaries specified by the morphological analyzer. The correct one should be '... / to / kai / te / ari / masu / shi / de / kono / moji /...'. If a 'kuten' (period in English) is inserted between 'shi' and 'de', which is a correct SB, the output would be '... / to / kai / te / ari / masu / shi / . / de / kono / moji / ...', which is the correct result.

In this paper, we present a method to automatically detect SBs in Japanese speech transcriptions. Our method is solely based on the linguistic information in a transcription, and it can be integrated with the method that uses prosodic (pause) information mentioned in the next section.

The target of our system is manual transcriptions rather than ASR transcriptions, but we plan to apply it to ASR transcriptions in the future. In the present work, we have used the transcribed speeches of 50 lecturers whose age and sex are not biased (The, 2001; The, 2002), and have con-

structed a corpus of 3499 sentences in which the SBs were manually inserted.

The next section discusses work related to SB detection. Section three describes the method of detecting SBs by using a morphological analyzer, and section four discusses the evaluation of our method.

2 Related work

Despite the importance of a technology that could detect SBs, there has been little work on the topic.

In English, Stevenson and Gaizauskas (Stevenson and Gaizauskas, 2000) have addressed the SB detection problem by using lexical cues. In Japanese, Shitaoka et al. (Shitaoka et al., 2002) and Nobata et al. (Nobata et al., 2002) have done work on SB detection.

Shitaoka et al. (Shitaoka et al., 2002) detected SBs by using the pause length in the speech and information about words that tend to appear just before and after SBs. Basically, the SBs are detected by using the probability $P(pause\ information|period)$. However, since pauses can occur in many places in speech (Seligman et al., 1996), many incorrect insertions occurred when they inserted kutens in all of them. Therefore, they limited the places of kuten insertion to the places just before and after the words such as 'masu', 'masune', 'desu' that tend to appear at the SBs.

Their method employs the following three pause lengths: (1)All pauses are used, (2)The pauses longer than the average length are used, (3)Assuming that the pause length differs depending on the expression, the pauses whose length exceeds a threshold for each expression are used. The best performance of their method, 78.4% recall, 85.1% precision, and 0.816 F-Measure, was obtained for (3).

Nobata et al. (Nobata et al., 2002) proposed a similar method combining pause information with the manually created lexical patterns for detecting SBs in lecture speech transcriptions.

Our method, by contrast, detects SBs by using only linguistic information in the transcription, and it can be integrated with Shitaoka's prosodic method.

Although little work has been done on SB detection, there has been work in the related field of SB disambiguation and comma restoration. SB disambiguation is a problem where punctuation is provided, but the categorization of the punctuation as to whether or not it marks a SB is at issue (Palmer and Hearst, 1994; Reynar and Ratnaparkhi, 1997). Comma restoration is, as it indicates, the task of inserting intrasentential punctuation into ASR output (Beeferman et al., 1998; Shieber and Tao, 2003; Tooyama and Nagata, 2000).

3 Proposed Method

Our method to detect SBs consists of two steps:

1. identify candidate places for inserting SBs,
2. delete unsuitable candidate places.
   - delete unsuitable candidates by using information about words that seldom appear at a SB,
   - delete unsuitable candidates by using information about combinations of words that seldom appears at a SB.

The following subsections explain each step in detail.

3.1 Identifying candidate SBs

To identify candidate places for inserting SBs, we use a Japanese morphological analyzer that is based on a cost calculation and selects as the best result the one with the minimum cost. The cost is determined by learning the suitable size corpus with a tag to the alternative trigram model which used bigram model as the base (Asahara and Matsumoto, 2000). The idea behind using a morphological analyzer to identify candidates is that it outputs lower costs for better sequences of morphemes.

Therefore, by comparing the cost of inserting a SB with the cost of not inserting a SB, if the cost is lower for inserting a boundary, we can judge that the location is a likely candidate and the sequence of morphemes is more correctly analyzed by the morphological analyzer.

Next, we briefly describe the costs used in the Japanese morphological analyzer and illustrate the method of identifying candidate SBs.

3.1.1 Costs used in the morphological analyzer

Cost is usually used for indicating the appropriateness of morphological analysis results, and lower cost results are preferred. A Japanese morphological analyzer usually uses a combination of

morpheme cost (cost for words or POSs (Part of Speech)) and connection cost (cost for two adjacent words or POSs) to calculate the appropriateness of a sequence of morphemes. The Japanese morphological analyzer of Chasen (Matsumoto et al., 2002), which we used in our work, analyzes the input string with the morpheme and connection costs statistically computed from the POS tagged corpus (Asahara and Matsumoto, 2002).

Consider, for example, the following two strings, 'oishii masu(delicious trout)' and 'itashi masu(I do)'. Although the end of both strings is 'masu', their POSs are different ('Noun-General'(NG) and 'Auxiliary Verb-Special MASU'(AVSM)), and their morpheme costs also differ as follows:

- The cost of the Noun- 'masu' is 4302,
- The cost of the Auxiliary Verb- 'masu' is 0.

Since 'oishii(the cost is 2545)' is an 'Adjective-Independence-Basic form'(AIB) and 'itashi(the cost is 3217)' is a 'Verb-Independence-Continuous form'(VIC) , by using the following connection cost,

- The cost of AIB + NG is 404,
- There are no connection rules for AIB + AVSM,
- The cost of VIC + NG is 1567,
- The cost of VIC + AVSM is 1261.

the cost for each sequence of morphemes is calculated as follows:

- 'oishii(Adjective) + masu(Noun)': 2545 + 404 + 4302 = 7251,
- 'oishii(Adjective) + masu(Auxiliary Verb)': unacceptable.
  - Therefore, the analysis result is 'oishii(Adjective) + masu(Noun)'.
- 'itashi(Verb) + masu(Noun)': 3217 + 1567 + 4302 = 9086,
- 'itashi(Verb) + masu(Auxiliary Verb)': 3217 + 1261 + 0 = 4478.
  - Because 9086 $>$ 4487, the analysis result is 'itashi(Verb) + masu(Auxiliary Verb)'.

Thus, by using costs, the morphological analyses will be grammatically correct.

3.1.2 Illustration of the process of identifying candidate SBs

Whether the place between 'shi' and 'de' and the place between 'de' and 'kono' in a string 'kaitearimasu shi de kono' can be a SB is judged according the following procedure:

1. The morphological analysis result of 'kaitearimasushidekono' is 'kai(Verb) / te(Particle) / arima(Noun) / sushi(Noun) / de(Particle) / kono(Attribute)' , and its cost is 16656. To compare it with the cost of the result including a kuten, the morpheme cost of a kuten (200) and the minimum connection cost for a kuten (0) are added to the above cost; therefore, 16656 + 200 + 0 + 0 = 16856 is the total cost for the sequence of morphemes.
2. The morphological analysis result of 'kaitearimasushi. dekono' is 'kai(Verb) / te(Particle) / ari(Verb) / masu(Verb) / shi(Particle) / .(Sign) / de(Conjunction) / kono(Attribute)' , and its cost is 14245.
3. The morphological analysis result of 'kaitearimasushide. kono' is 'kai(Verb) / te(Particle) / arima(Noun) / sushi(Noun) / de(Particle) / .(Sign) / kono(Attribute)' , and its cost is 18018.
4. Because 16856 $>$ 14245 from 1 and 2, the latter can be considered as the better sequence of morphemes. Therefore, the place between 'shi' and 'de' can be a candidate for a SB.
5. Because 16856 $<$ 18018 from 1 and 3, the former can be considered as the better sequence of morphemes. Therefore, the place between 'de' and 'kono' cannot be a SB.

As illustrated above, by inserting a kuten between two morphemes in the input string, calculating the cost, and judging whether the place should be a candidate SB, we can enumerate all the candidate SBs.

3.2 Deleting unsuitable candidates

3.2.1 Deletion using words that seldom appear at a SB

Certain words tend to appear at the beginnings or ends of sentences. Therefore, the candidate places just before and after such words can be considered as suitable, whereas the other candidates may be unsuitable and should be deleted.

The words that tend to appear at a SB can be obtained by calculating for each word that appears just before and after the identified candidate SBs the following ratio in the training corpus: the number of occurrences in which a word appears at the correct SB to the number of occurrences in which the word appears at the candidate SB. The words with higher ratios tend to appear at SB. The sample words with higher ratios are shown in Table1.

Table 1: The sample words with which tend to appear before and after SBs

the words which appear after SBs		
de (324/330)	e (287/436)	ee (204/524)
the words which appear before SBs		
masu (1015/1084)	ta (251/394)	desu (260/367)

By summing the values of the words just before and after each candidate SB, we can judge whether the candidate is suitable or not. If the sum of these values does not exceed a predetermined threshold, the candidate is judged as unsuitable and deleted. The threshold was empirically set to 0.7 in this work.

3.2.2 Deletion using combinations of words that seldom appear at a SB

Even if a word that tends to appear in a SB appears before or after the candidate SB, the candidate might still not be suitable, if the combination of the words seldom appears at a SB.

Consider the following example. In the training corpus, the string 'desuga'(no kuten insertion between 'desu' and 'ga') occurs, but the string 'desu. ga' never occurs, although 'desu' tends to appear at the end of a sentence, as shown in Table1.

Therefore, in case of the string 'kotodesugakono', the method in the last section cannot delete the unsuitable candidate SB between 'desu' and 'ga' because the value of 'desu' exceeds the threshold, as shown in Table1.

- The morphological analysis result of 'kotodesugakono' is 'koto(Noun) / desuga(Conjunction) / kono(Attribute)' . The total cost is 12730 + 200 + 0 + 0 = 12930.
- The morphological analysis result of 'kotodesu. gakono' is 'koto(Noun) / desu(Auxiliary verb) / .(Sign) / ga(Conjunction)/ kono(Attribute)'. The cost is 9938.
- Because 12930 $>$ 9938, the place between 'desu' and 'ga' can be a candidate SB.
- The ratio in the last section for 'desu' is 260/367 = 0.7084 $>$ 0.7; therefore, whatever the value of 'ga' may be, the place between 'desu' and 'ga' will not be deleted as a result of using the method described in the last section.

To cope with the above problem, we need another method to delete unsuitable candidate places, i.e., one that uses the combination of words which seldom appears at a SB:

1. Identify in the corpus all the combination of words which tend to appear just before and after a SB,
2. If the occurrence of the combination of words without kuten insertion exceeds the preset threshold in the training corpus, select the combination as one that seldom appears in a SB. (The threshold was set to 10 in this work.) Furthermore, to prevent incorrect deletions, do not select the combination which occur once or more with kuten insertion in the training corpus.
3. If the combination of words just before or after the identified candidate SB is one that seldom appears at a SB, the candidate is deleted.

This method can cope with the above example; that is, it deletes the candidate SB between 'desu' and 'ga'.

4 Evaluation

4.1 Evaluation measure

Precision, recall, and F-Measure were the measures used for the evaluation. They were defined as follows: Precision is the ratio of the number of correct SBs identified by the method to the number of boundaries identified by the method. Recall is the ratio of the number of correct SBs identified by the method to the total number of correct boundaries. The F-Measure was calculated with following formula:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

The corpus, consisting of 3499 sentences for which kutens were manually inserted, was divided into five parts, and the experiments used a 5-fold cross validation.

4.2 Determining the direction of identifying the candidate boundaries

The identification of the candidate SBs using a morphological analyzer in section 3.1 can be performed in two directions: from the beginning to the end of the input string, or vice versa. If it is performed from the beginning, the place after the first morpheme in the string is tried first, and the place after the second is tried second, and so on. [1]

We first conducted experiments in both directions. The F-Measures for either direction were equal 0.8215, but the places identified sometimes differed according to direction. Therefore, we calculated the intersection and union of the places for the two directions. F-Measure for the intersection is 0.8227 and the union is 0.8218.

From these results, we can conclude that the intersection of both directions yields the best performance; therefore, we will use the intersection result hereafter.

4.3 Evaluating the effect of each method

Four experiments were conducted to investigate the effect of each method described in section 3:

1. Use only the method to identify candidate boundaries,
2. Use the method to identify the candidate boundaries and the deletion method using the words which seldom appear at a SB,
3. Use the method to identify the candidate boundaries and the deletion method using the combination of words which seldom appears at a SB,
4. Use all the methods.

The results are shown in Table 2. The recall of the identification method turns out to be about 82%. Since recall becomes lower by using the deletion methods, it is desirable that the identification method have a higher recall.

Comparing 1 and 2 of Table 2, the deletion of seldom appearing words can improve precision by about 40%, while lowering recall by about 4%. A similar result can be seen by comparing 3 and 4.

Table 2: The results for each experiment

	1	2	3	4
Recall	0.8184	0.7813	0.8182	0.7724
Precision	0.4602	0.8571	0.4719	0.8800
F-Measure	0.5889	0.8174	0.5982	0.8227

Comparing 1 and 3 of Table 2, the deletion of seldom appearing combinations of words can slightly improve precision with almost no lowering of recall. A similar result can be seen by comparing 2 and 4.

From these results, we can conclude that since both deletion methods can raise F-Measure, they can be considered as effective.

4.4 Error Analysis

The following are samples of errors caused by our method:

1. 'itashimashitadeeenettono'(FN; False Negatives) [2]
2. 'mierunda. keredomokoreha'(FP; False Positives) [3]

The reasons for the errors are as follows:

1. The SB between 'ta' and 'de' cannot be detected for 'itashimashi ta de ee nettono', because the string contains a filler 'ee'('ah' in English), and the morphological analyzer could not correctly analyze the string.

   When the input string contains fillers and repairs, the morphological analyzer sometimes analyzes the string incorrectly.
2. The place between 'da' and 'keredomo' was incorrectly detected as a SB for 'mierun da. keredomo koreha', because the combination of the words 'da' and 'keredomo' seldom appears at a SB but the number of occurrences is not zero; the combination was not selected as one that seldom appears at a SB.

5 Conclusion

In this paper, we presented a method that uses a Japanese morphological analyzer to automatically detect SBs in Japanese speech transcriptions.

[1] (Liu and Zong, 2003) described the same problem, and tries to resolve it by multiplying the probability for the normal and opposite directions.

[2] Errors where the method misses the correct boundaries

[3] Errors where the method incorrectly inserts boundaries

Our method could yield a 77.24% precision, 88.00% recall, and 0.8277 F-Measure for a corpus consisting of lecture speech transcriptions in which SBs are not given. We found that by detecting SBs with our method, the morphological analysis could be performed more accurately and the error rate of the analyzer could be reduced, although the quantitative evaluation was not performed.

Our method could outperform Shitaoka et al.'s method (Shitaoka et al., 2002), which uses pause information and yields 78.4% precision, 85.1% recall, and 0.816 F-Measure, although this assessment is somewhat subjective as the corpus for their evaluations was different from ours. Our method can be integrated with the method that uses prosodic (pause) information, and such an integration would improve the overall performance.

As we mentioned in section 4.3, our method's recall was only 77.24%. A future work would therefore be to improve the recall, which would be possible if we had a larger training corpus in which SBs are manually tagged. Furthermore, we would like to apply our method to ASR speech transcriptions in the future.

We think our method can also be applied to English if a POS tagger is used in place of the Japanese morphological analyzer.

References

Masayuki Asahara and Yuji Matsumoto. 2000. Extended Hidden Markov Model for Japanese Morphological Analyzer. In *IPSJ SIG Notes on Spoken Language Processing, No.031.* in Japanese.

Masayuki Asahara and Yuji Matsumoto, 2002. *IPADIC user's manual version 2.5.*

Doug Beeferman, Adam Berger, and John Lafferty. 1998. CYBERPUNC: A lightweight punctu IEEE International Conference on Acoustics, Speech and Signaation annotation system for speech. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 689–692.

Ding Liu and Chengqing Zong. 2003. Utterance Segmentation Using Combined Approach Based on Bi-directional N-gram and Maximum Entropy. In *Proc. of ACL-2003 Workshop: The Second SIGHAN Workshop on ChineeseLanguage Processing*, pages 16–23.

Yuji Matsumoto, Akira Kitauchi, Yoshitaka Hirano Tatsuo Yamashita, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara, 2002. *Morphological Analysis System ChaSen version2.2.9 Manual*.

Chikashi Nobata, Satoshi Sekine, Kiyotaka Uchimoto, and Hitoshi Isahara. 2002. Sentence Segmentation and Sentence Extraction. In *Proc. of the Second Spontaneous Speech Science and Technology Workshop*, pages 527–534. in Japanese.

David D. Palmer and Marti A. Hearst. 1994. Adaptive sentence boundary disambiguation. In *Proc. of the fourth Conference on Applied Natural Language Processing*, pages 78–83.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proc. of the fifth Conference on Applied Natural Language Processing*, pages 16–19.

Mark Seligman, Junko Hosaka, and Harald Singer. 1996. "Pause Units" and Analysis of Spontaneous Japanese Dialogues: Preliminary Studies. In *ECAI-96 workshop on "Dialogue Processing in Spoken Language Systems"*, pages 100–112.

Stuart M. Shieber and Xiaopeng Tao. 2003. Comma restoration using constituency information. In *Proc. of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 221–227.

Kazuya Shitaoka, Tatsuya Kawahara, and Hiroshi G. Okuno. 2002. Automatic Transformation of Lecture Transcription into Document Style using Statistical Framework. In *IPSJ SIG Notes on Spoken Language Processing, 41-3.* in Japanese.

Mark Stevenson and Robert Gaizauskas. 2000. Experiments on Sentence Boundary Detection. In *Proc. of ANLP-NAACL2000*, pages 84–89.

The National Institute for Japanese Language, 2001. *'The Corpus of Spontaneous Japanese(The monitor version 2001) Guidance of monitor public presentation.* `http://www.kokken.go.jp/public/monitor_kokai001.html`.

The National Institute for Japanese Language, 2002. *'The Corpus of Spontaneous Japanese(The monitor version 2002) Guidance of monitor public presentation.* `http://www.kokken.go.jp/public/monitor_kokai002.html`.

Yosuke Tooyama and Morio Nagata. 2000. Insertion methods of punctuation marks for speech recognition systems. In *Technical Report of IEICE, NLC2000-5.* in Japanese.