Linefeed Insertion into Japanese Spoken Monologue for Captioning

Tomohiro Ohno
Graduate School of
International Development,
Nagoya University, Japan
ohno@nagoya-u.jp

Masaki Murata
Graduate School of
Information Science,
Nagoya University, Japan
murata@el.itc.nagoya-u.ac.jp

Shigeki Matsubara
Information Technology Center,
Nagoya University, Japan
matubara@nagoya-u.jp

Abstract

To support the real-time understanding of spoken monologue such as lectures and commentaries, the development of a captioning system is required. In monologues, since a sentence tends to be long, each sentence is often displayed in multi lines on one screen, it is necessary to insert linefeeds into a text so that the text becomes easy to read. This paper proposes a technique for inserting linefeeds into a Japanese spoken monologue text as an elemental technique to generate the readable captions. Our method appropriately inserts linefeeds into a sentence by machine learning, based on the information such as dependencies, clause boundaries, pauses and line length. An experiment using Japanese speech data has shown the effectiveness of our technique.

1 Introduction

Real-time captioning is a technique for supporting the speech understanding of deaf persons, elderly persons, or foreigners by displaying transcribed texts of monologue speech such as lectures. In recent years, there exist a lot of researches about automatic captioning, and the techniques of automatic speech recognition (ASR) aimed for captioning have been developed (Boulianne et al., 2006; Holter et al., 2000; Imai et al., 2006; Munteanu et al., 2007; Saraclar et al., 2002; Xue et al., 2006). However, in order to generate captions which is easy to read, it is important not only to recognize speech with high recognition rate but also to properly display the transcribed text on a screen (Hoogenboom et al., 2008). Especially, in spoken monologue, since a sentence tends to be long, each sentence is often displayed as a multi-line text on a screen. Therefore, proper linefeed insertion for the displayed text is desired so that the text becomes easy to read.

Until now, there existed few researches about how to display text on a screen in automatic captioning. As the research about linefeed insertion, Monma et al. proposed a method based on patterns of a sequence of morphemes (Monma et al., 2003). However, the target of the research is closed-captions of Japanese TV shows, in which less than or equal to 2 lines text is displayed on a screen and the text all switches to other text at a time. In the work, the highest priority concept on captioning is that one screen should be filled with as much text as possible. Therefore, a semantic boundary in a sentence is hardly taken into account in linefeed insertion, and the readability of the caption is hardly improved.

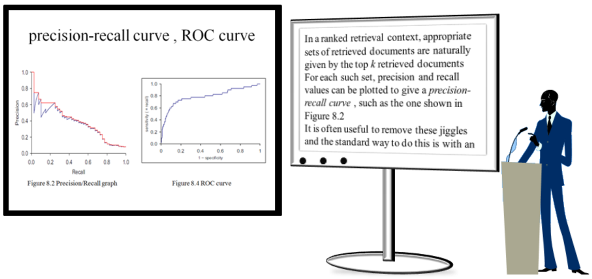
This paper proposes a technique for inserting linefeeds into transcribed texts of Japanese monologue speech as an elemental technique to generate readable captions. We assume that a screen for displaying only multi-line caption is placed to provide the caption information to the audience on the site of a lecture. In our method, the linefeeds are inserted into only the boundaries between *bunsetsus*[1], and the linefeeds are appropriately inserted into a sentence by machine learning, based on the information such as morphemes, dependencies[2], clause boundaries, pauses and line length.

We conducted an experiment on inserting linefeeds by using Japanese spoken monologue data. As the results of inserting linefeeds for 1,714 sentences, the recall and precision of our method were 82.66% and 80.24%, respectively. Our method improved the performance dramatically compared

[1] *Bunsetsu* is a linguistic unit in Japanese that roughly corresponds to a basic phrase in English. A bunsetsu consists of one independent word and zero or more ancillary words.

[2] A dependency in Japanese is a modification relation in which a modifier bunsetsu depends on a modified bunsetsu. That is, the modifier bunsetsu and the modified bunsetsu work as modifier and modifyee, respectively.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 531–539,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP



In a ranked retrieval context, appropriate sets of retrieved documents are naturally given by the top *k* retrieved documents For each such set, precision and recall values can be plotted to give a *precision-recall curve* , such as the one shown in Figure 8.2
It is often useful to remove these jiggles and the standard way to do this is with an

Figure 1: Caption display of spoken monologue

with four baseline methods, which we established for comparative evaluation. The effectiveness of our method has been confirmed.

This paper is organized as follows: The next section describes our assumed caption and the preliminary analysis. Section 3 presents our linefeed insertion technique. An experiment and discussion are reported in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

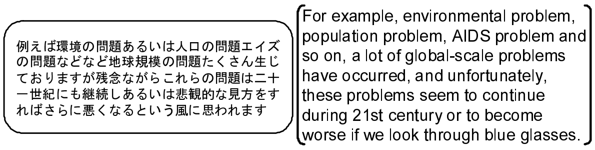
2 Linefeed Insertion for Spoken Monologue

In our research, in an environment in which captions are displayed on the site of a lecture, we assume that a screen for displaying only captions is used. In the screen, multi lines are always displayed, being scrolled line by line. Figure 1 shows our assumed environment in which captions are displayed.

As shown in Figure 2, if the transcribed text of speech is displayed in accordance with only the width of a screen without considering the proper points of linefeeds, the caption is not easy to read. Especially, since the audience is forced to read the caption in synchronization with the speaker's utterance speed, it is important that linefeeds are properly inserted into the displayed text in consideration of the readability as shown in Figure 3.

To investigate whether the line insertion facilitates the readability of the displayed texts, we conducted an experiment using the transcribed text of lecture speeches in the Simultaneous Interpretation Database (SIDB) (Matsubara et al., 2002). We randomly selected 50 sentences from the data, and then created the following two texts for each sentence based on two different concepts about linefeed insertion.

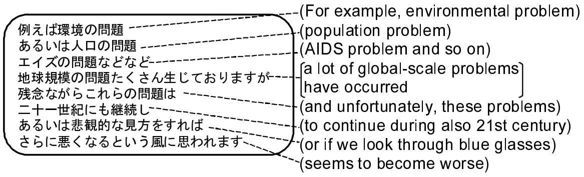
(1) Text into which linefeeds were forcibly inserted once every 20 characters



例えば環境の問題あるいは人口の問題エイズ
の問題などなど地球規模の問題たくさん生じ
ておりますが残念ながらこれらの問題は二十
一世紀にも継続しあるいは悲観的な見方をす
ればさらに悪くなるという風に思われます

For example, environmental problem,
population problem, AIDS problem and
so on, a lot of global-scale problems
have occurred, and unfortunately,
these problems seem to continue
during 21st century or to become
worse if we look through blue glasses.

Figure 2: Caption of monologue speech



例えば環境の問題
あるいは人口の問題
エイズの問題などなど
地球規模の問題たくさん生じておりますが
残念ながらこれらの問題は
二十一世紀にも継続し
あるいは悲観的な見方をすれば
さらに悪くなるという風に思われます

(For example, environmental problem)
(population problem)
(AIDS problem and so on)
[a lot of global-scale problems
have occurred]
(and unfortunately, these problems)
(to continue during also 21st century)
(or if we look through blue glasses)
(seems to become worse)

Figure 3: Caption into which linefeeds are properly inserted

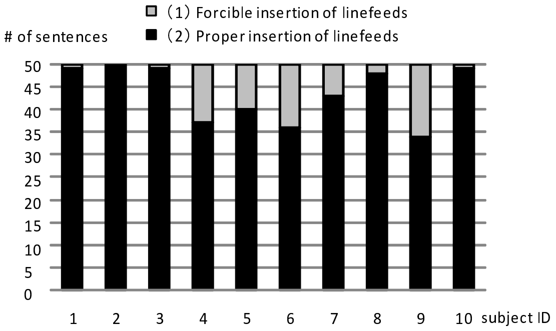


Figure 4: Result of investigation of effect of linefeed insertion into transcription

(2) Text into which linefeeds were properly inserted in consideration of readability by hand[3]

Figure 2 and 3 show examples of the text (1) and (2), respectively. 10 examinees decided which of the two texts was more readable. Figure 4 shows the result of the investigation. The ratio that each examinee selected text (2) was 87.0% on average. There was no sentence in the text group (1) which was selected by more than 5 examinees. These indicates that a text becomes more readable by proper insertion of linefeeds.

Here, since a bunsetsu is the smallest semantically meaningful language unit in Japanese, our method adopts the bunsetsu boundaries as the candidates of points into which a linefeed is inserted. In this paper, hereafter, we call a bunsetsu boundary into which a linefeed is inserted a **linefeed point**.

[3] 3 persons inserted linefeeds into the 50 sentences by discussing where to insert the linefeeds.

Table 1: Size of analysis data

sentence	221
bunsetsu	2,891
character	13,899
linefeed	883
character per line	13.2

3 Preliminary Analysis about Linefeed Points

In our research, the points into which linefeeds should be inserted is detected by using machine learning. To find the effective features, we investigated the spoken language corpus. In our investigation, we used Japanese monologue speech data in the SIDB (Matsubara et al., 2002). The data is annotated by hand with information on morphological analysis, bunsetsu segmentation, dependency analysis, clause boundary detection, and linefeeds insertion. Table 1 shows the size of the analysis data. Among 2,670 ($= 2,891 - 221$) bunsetsu boundaries, which are candidates of linefeed points, there existed 833 bunsetsu boundaries into which linefeeds were inserted, that is, the ratio of linefeed insertion was 31.2%.

The linefeeds were inserted by hand so that the maximum number of characters per line is 20. We set the number in consideration of the relation between readability and font size on the display. In the analysis, we focused on the clause boundary, dependency relation, line length, pause and morpheme of line head, and investigated the relations between them and linefeed points.

3.1 Clause Boundary and Linefeed Point

Since a clause is one of semantically meaningful language units, the clause boundary is considered to be a strong candidate of a linefeed point. In the analysis data, there existed 969 clause boundaries except sentence breaks. Among them, 490 were the points into which linefeeds were inserted, that is, the ratio of linefeed insertion was 51.1%. This ratio is higher than that of bunsetsu boundaries. This indicates that linefeeds tend to be inserted into clause boundaries.

We investigated the ratio of linefeed insertion about 42 types[4] of clause boundaries, which were seen in the analysis data. Table 2 shows the top 10 clause boundary types about the occurrence frequency, and each ratio of linefeed insertion. In the case of "compound clause-*ga*" and "compound clause-*keredomo*," the ratio of linefeed insertion was 100%. On the other hand, in the case of "quotational clause," "adnominal clause" and "adnominal clause-*toiu*," the ratio of linefeed insertion was less than 30%. This means that the likelihood of linefeed insertion is different according to the type of the clause boundary.

Table 2: Ratio of linefeed insertion for clause boundary type

type of clause boundary	ratio of linefeed insertion (%)
topicalized element-*wa*	50.8
discourse marker	12.0
quotational clause	22.1
adnominal clause	23.3
compound clause-*te*	90.2
supplement clause	68.0
compound clause-*ga*	100.0
compound clause-*keredomo*	100.0
condition clause-*to*	93.5
adnominal clause-*toiu*	27.3

3.2 Dependency Structure and Linefeed Point

When a bunsetsu depends on the next bunsetsu, it is thought that a linefeed is hard to be inserted into the bunsetsu boundary between them because the sequence of such bunsetsus constitutes a semantically meaningful unit. In the analysis data, there existed 1,459 bunsetsus which depend on the next bunsetsu. Among the bunsetsu boundaries right after them, 192 were linefeed points, that is, the ratio of linefeed insertions was 13.2%. This ratio is less than half of that for all the bunsetsu boundaries. On the other hand, when the bunsetsu boundary right after the bunsetsu which does not depend on the next bunsetsu, the ratio of linefeed insertion was 52.7%.

Next, we focused on the type of the dependency relation, by which the likelihood of linefeed insertion is different. For example, when the bunsetsu boundary right after a bunsetsu on which the final bunsetsu of an adnominal clause depends, the ratio of linefeed insertion was 43.1%. This ratio is higher than that for all the bunsetsu boundaries.

In addition, we investigated the relation be-

[4] In our research, we used the types of clause boundaries defined by the Clause Boundary Annotation Program (Kashioka and Maruyama, 2004).

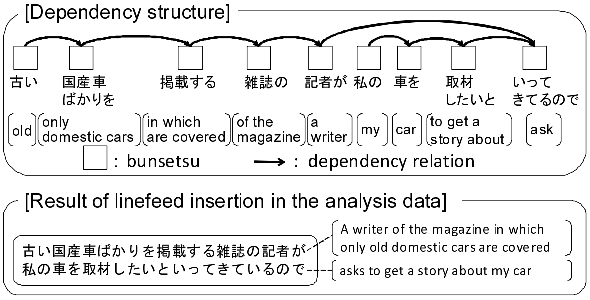

[Dependency structure]
古い
国産車ばかりを
掲載する
雑誌の
記者が
私の
車を
取材したいと
いってきてるので
old
only domestic cars
in which are covered
of the magazine
a writer
my
car
to get a story about
ask
: bunsetsu
: dependency relation
[Result of linefeed insertion in the analysis data]
古い国産車ばかりを掲載する雑誌の記者が
私の車を取材したいといってきているので
A writer of the magazine in which only old domestic cars are covered
asks to get a story about my car

Figure 5: Relation between dependency structure and linefeed points

tween a dependency structure and linefeed points, that is, whether the dependency structure is closed within a line or not. Here, a line whose dependency structure is closed means that all bunsetsus, except the final bunsetsu, in the line depend on one of bunsetsus in the line. Since, in many of semantically meaningful units, the dependency structure is closed, the dependency structure of a line is considered to tend to be closed. In the analysis data, among 883 lines, 599 lines' dependency structures were closed.

Figure 5 shows the relation between dependency structure and linefeed points. In this example, linefeeds are not inserted right after bunsetsus which depend on the next bunsetsu (e.g. "私の (my)" and "車を (car)"). Instead, a linefeed is inserted right after a bunsetsu which does not depend on the next bunsetsu ("記者が (a writer)"). In addition, the dependency structure in each line is closed.

3.3 Line Length and Linefeed Point

An extremely-short line is considered to be hardly generated because the readability goes down if the length of each line is very different. In the analysis data, a line whose length is less than or equal to 6 characters occupied only 7.59% of the total. This indicates that linefeeds tend to be inserted into the place where a line can maintain a certain length.

3.4 Pause and Linefeed Point

It is thought that a pause corresponds to a syntactic boundary. Therefore, there are possibility that a linefeed becomes more easily inserted into a bunsetsu boundary at which a pause exists. In our research, a pause is defined as a silent interval equal to or longer than 200ms. In the analysis data, among 748 bunsetsu boundaries at which a pause exists, linefeeds were inserted into 471 bunsetsu boundaries, that is, the ratio of linefeed insertion was 62.97%. This ratio is higher than that for all the bunsetsu boundaries, thus, we confirmed that linefeeds tend to be inserted into bunsetsu boundaries at which a pause exists.

3.5 Morpheme Located in the Start of a Line

There exist some morphemes which are unlikely to become a line head. We investigated the ratio that each leftmost morpheme of all the bunsetsus appears at a line head. Here, we focused on the basic form and part-of-speech of a morpheme. The morphemes which appeared 20 times and of which the ratio of appearance at a line head was less than 10% were as follows:

- Basic form:
  "思う (think) [2/70]", "問題 (problem) [0/42]", "する (do) [3/33]", "なる (become) [2/32]", "必要 (necessary) [1/21]"
- Part-of-speech:
  noun-non_independent-general [0/40],
  noun-*nai*_adjective_stem [0/40],
  noun-non_independent-adverbial [(0/27]

If the leftmost morpheme of a bunsetsu is one of these, it is thought that a linefeed is hardly inserted right after the bunsetsu.

4 Linefeed Insertion Technique

In our method, a sentence, on which morphological analysis, bunsetsu segmentation, clause boundary analysis and dependency analysis are performed, is considered the input. Our method decides whether or not to insert a linefeed into each bunsetsu boundary in an input sentence. Under the condition that the number of characters in each line has to be less than or equal to the maximum number of characters per line, our method identifies the most appropriate combination among all combinations of the points into which linefeeds can be inserted, by using the probabilistic model.

In this paper, we describe an input sentence which consists of n bunsetsus as $B = b_1 \cdots b_n$, and the result of linefeeds insertion as $R = r_1 \cdots r_n$. Here, r_i is 1 if a linefeed is inserted right after bunsetsu b_i, and is 0 otherwise. We describe a sequence of bunsetsus in the j-th line among the m lines created by dividing an input sentence as $L_j = b_1^j \cdots b_{n_j}^j (1 \leq j \leq m)$, and then, $r_k^j = 0$ if $k \neq n_j$, and $r_k^j = 1$ otherwise.

4.1 Probabilistic Model for Linefeed Insertion

When an input sentence B is provided, our method identifies the result of linefeeds insertion R, which maximizes the conditional probability $P(R|B)$. Assuming that whether or not a linefeed is inserted right after a bunsetsu is independent of other linefeed points except the linefeed point of the start of the line which contains the bunsetsu, $P(R|B)$ can be calculated as follows:

$$\begin{aligned} & P(R|B) \qquad (1) \\ = \ & P(r_1^1 = 0, \cdots, r_{n_1}^1 = 1, \cdots, r_1^m = 0, \cdots, r_{n_m}^m = 1|B) \\ \cong \ & P(r_1^1 = 0|B) \times P(r_2^1 = 0|r_1^1 = 0, B) \times \cdots \\ & \times P(r_{n_1}^1 = 1|r_{n_1-1}^1 = 0, \cdots, r_1^1 = 0, B) \times \cdots \\ & \times P(r_1^m = 0|r_{n_{m-1}}^{m-1} = 1, B) \times \cdots \\ & \times P(r_m^m = 1|r_{n_m-1}^m = 0, \cdots, r_1^m = 0, r_{n_{m-1}}^{m-1} = 1, B) \end{aligned}$$

where $P(r_k^j = 1|r_{k-1}^j = 0, \cdots, r_1^j = 0, r_{n_{j-1}}^{j-1} = 1, B)$ is the probability that a linefeed is inserted right after a bunsetsu b_k^j when the sequence of bunsetsus B is provided and the linefeed point of the start of the j-th line is identified. Similarly, $P(r_k^j = 0|r_{k-1}^j = 0, \cdots, r_1^j = 0, r_{n_{j-1}}^{j-1} = 1, B)$ is the probability that a linefeed is not inserted right after a bunsetsu b_k^j. These probabilities are estimated by the maximum entropy method. The result R which maximizes the conditional probability $P(R|B)$ is regarded as the most appropriate result of linefeed insertion, and calculated by dynamic programming.

4.2 Features on Maximum Entropy Method

To estimate $P(r_k^j = 1|r_{k-1}^j = 0, \cdots, r_1^j = 0, r_{n_{j-1}}^{j-1} = 1, B)$ and $P(r_k^j = 0|r_{k-1}^j = 0, \cdots, r_1^j = 0, r_{n_{j-1}}^{j-1} = 1, B)$ by the maximum entropy method, we used the following features based on the analysis described in Section 2.2.

Morphological information

- the rightmost independent morpheme (a part-of-speech, an inflected form) and rightmost morpheme (a part-of-speech) of a bunsetsu b_k^j

Clause boundary information

- whether or not a clause boundary exists right after b_k^j
- a type of the clause boundary right after b_k^j (if there exists a clause boundary)

Dependency information

- whether or not b_k^j depends on the next bunsetsu
- whether or not b_k^j depends on the final bunsetsu of a clause
- whether or not b_k^j depends on a bunsetsu to which the number of characters from the start of the line is less than or equal to the maximum number of characters
- whether or not b_k^j is depended on by the final bunsetsu of an adnominal clause
- whether or not b_k^j is depended on by the bunsetsu located right before it
- whether or not the dependency structure of a sequence of bunsetsus between b_k^j and b_1^j, which is the first bunsetsu of the line, is closed
- whether or not there exists a bunsetsu which depends on the modified bunsetsu of b_k^j, among bunsetsus which are located after b_k^j and to which the number of characters from the start of the line is less than or equal to the maximum number of characters

Line length

- any of the following is the class into which the number of characters from the start of the line to b_k^j is classified
  - less than or equal to 2
  - more than 2 and less than or equal to 6
  - more than 6

Pause

- whether or not a pause exists right after b_k^j

Leftmost morpheme of a bunsetsu

- whether or not the basic form or part-of-speech of the leftmost morpheme of the next bunsetsu of b_k^j is one of the morphemes enumerated in Section 3.5.

5 Experiment

To evaluate the effectiveness of our method, we conducted an experiment on inserting linefeeds by using discourse speech data.

5.1 Outline of Experiment

As the experimental data, we used the transcribed data of Japanese discourse speech in the SIDB (Matsubara et al., 2002). All the data are annotated with information on morphological analysis, clause boundary detection and dependency analysis by hand. We performed a cross-validation experiment by using 16 discourses. That is, we repeated the experiment, in which we used one discourse from among 16 discourses as the test data and the others as the learning data, 16 times. However, since we used 2 discourse among 16 discourses as the preliminary analysis data, we evaluated the experimental result for the other 14 discourses (1,714 sentences, 20,707 bunsetsus). Here, we used the maximum entropy method tool (Zhang, 2008) with the default options except "-i 2000."

In the evaluation, we obtained recall, precision and the ratio of sentences into which all linefeed points were correctly inserted (hereinafter called **sentence accuracy**). The recall and precision are respectively defined as follows.

$$recall = \frac{\#\ of\ correctly\ inserted\ LFs}{\#\ of\ LFs\ in\,the\ correct\ data}$$

$$precision = \frac{\#\ of\ correctly\ inserted\ LFs}{\#\ of\ automatically\ inserted\ LFs}$$

For comparison, we established the following four baseline methods.

1. Linefeeds are inserted into the rightmost bunsetsu boundaries among the bunsetsu boundaries into which linefeeds can be inserted so that the length of the line does not exceed the maximum number of characters **(Linefeed insertion based on bunsetsu boundaries)**.
2. Linefeeds are inserted into the all clause boundaries **(Linefeed insertion based on clause boundaries)**.
3. Linefeeds are inserted between adjacent bunsetsus which do not depend on each other **(Linefeed insertion based on dependency relations)**.
4. Linefeeds are inserted into the all bunsetsu boundaries in which a pause exists **(Linefeed insertion based on pauses)**.

In the baseline 2, 3 and 4, if each condition is not fulfilled within the maximum number of characters, a linefeed is inserted into the rightmost bunsetsu boundary as well as the baseline 1.

In the experiment, we defined the maximum number of characters per line as 20. The correct data of linefeed insertion were created by experts who were familiar with displaying captions. There existed 5,497 inserted linefeeds in the 14 discourses, which were used in the evaluation.

Table 3: Experimental results

	recall (%)	precision (%)	F-measure
our method	82.66 (4,544/5,497)	80.24 (4,544/5,663)	81.43
baseline 1	27.47 (1,510/5,497)	34.51 (1,510/4,376)	30.59
baseline 2	69.34 (3,812/5,497)	48.65 (3,812/7,834)	57.19
baseline 3	89.48 (4,919/5,497)	53.73 (4,919/9,155)	67.14
baseline 4	69.84 (3,893/5,497)	55.60 (3,893/6,905)	61.91

5.2 Experimental Result

Table 3 shows the experimental results of the baselines and our method. The baseline 1 is very simple method which inserts linefeeds into the bunsetsu boundaries so that the length of the line does not exceed the maximum number of characters per line. Therefore, the recall and precision were the lowest.

In the result of baseline 2, the precision was low. As described in the Section 3.1, the degree in which linefeeds are inserted varies in different types of clause boundaries. In the baseline 2, because linefeeds are also inserted into clause boundaries which have the tendency that linefeeds are hardly inserted, the unnecessary linefeeds are considered to have been inserted.

The recall of baseline 3 was very high. This is because, in the correct data, linefeeds were hardly inserted between two neighboring bunsetsus which are in a dependency relation. However, the precision was low, because, in the baseline 3, linefeeds are invariably inserted between two neighboring bunsetsus which are not in a dependency relation.

In the baseline 4, both the recall and precision were not good. The possible reason is that the bunsetsu boundaries at which a pause exists do not necessarily correspond to the linefeed points.

On the other hand, the F-measure and the sentence accuracy of our method were 81.43 and 53.15%, respectively. Both of them were highest among those of the four baseline, which showed an effectiveness of our method.

5.3 Causes of Incorrect Linefeed Insertion

In this section, we discuss the causes of the incorrect linefeed insertion occurred in our method. Among 1,119 incorrectly inserted linefeeds, the most frequent cause was that linefeeds were in-

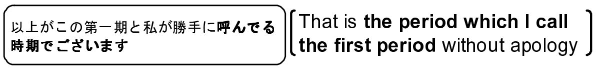

以上がこの第一期と私が勝手に呼んでる
時期でございます
That is the period which I call
the first period without apology

Figure 6: Example of incorrect linefeed insertion in "adnominal clause."

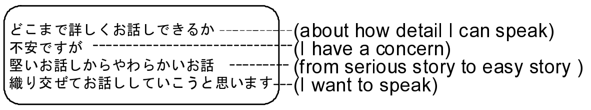

どこまで詳しくお話しできるか
不安ですが
堅いお話しからやわらかいお話
織り交ぜてお話ししていこうと思います
(about how detail I can speak)
(I have a concern)
(from serious story to easy story)
(I want to speak)

Figure 7: Example of extra linefeed insertion

serted into clause boundaries of a "adnominal clause" type. The cause occupies 10.19% of the total number of the incorrectly inserted linefeeds. In the clause boundaries of the "adnominal clause" type, linefeeds should rarely be inserted fundamentally. However, in the result of our method, a lot of linefeeds were inserted into the "adnominal clause." Figure 6 shows an example of those results. In this example, a linefeed is inserted into the "adnominal clause" boundary which is located right after the bunsetsu "呼んでる (call)." The semantic chunk "呼んでる時期でございます (is the period which I call)" is divided.

As another cause, there existed 291 linefeeds which divide otherwise one line according to the correct data into two lines. Figure 7 shows an example of the extra linefeed insertion. Although, in the example, a linefeed is inserted between "どこまで詳しくお話しできるか (about how detail I can speak)" and "不安ですが (I have a concern)," the two lines are displayed in one line in the correct data. It is thought that, in our method, linefeeds tend to be inserted even if a line has space to spare.

6 Discussion

In this section, we discuss the experimental results described in Section 5 to verify the effectiveness of our method in more detail.

6.1 Subjective Evaluation of Linefeed Insertion Result

The purpose of our research is to improve the readability of the spoken monologue text by our linefeed insertion. Therefore, we conducted a subjective evaluation of the texts which were generated by the above-mentioned experiment.

In the subjective evaluation, examinees looked at the two texts placed side-by-side between which the only difference is linefeed points, and then selected the one which was felt more readable. Here, we compared our method with the baseline 3, of which F-measure was highest among four baselines described in Section 5.1. Ten examinees evaluated 50 pairs of the results generated from the same 50 randomly selected sentences.

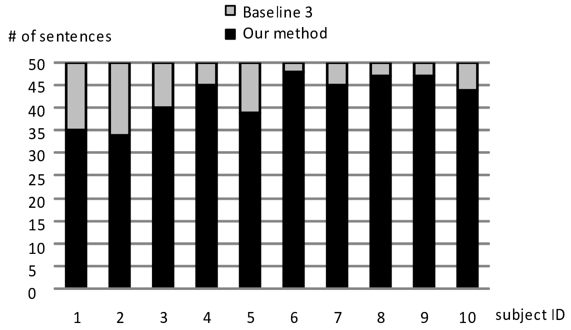


Figure 8: Result of subjective evaluation

Figure 8 shows the result of subjective evaluation. This graph shows the number of each method selected by each examinee. The ratio that our method was selected was 94% in the highest case, and 68% even in the lowest case. We confirmed the effectiveness of our method for improving the readability of the spoken monologue text.

On the other hand, there existed three sentences for which more than 5 examinees judged that the results of baseline 3 were more readable than those of our method. From the analysis of the three sentences, we found the following phenomena caused text to be less readable

- Japanese syllabary characters (Hiragana) are successionally displayed across a bunsetsu boundary.
- The length of anteroposterior lines is extremely different each other.

Each example of the two causes is shown in Figure 9 and 10, respectively. In Figure 9, a bunsetsu boundary existed between Japanese syllabary characters "私もですね (I)" and "かくゆう (if truth be told)" and these characters are successionally displayed in the same line. In these cases, it becomes more difficult to identify the bunsetsu boundary, therefore, the text is thought to become difficult to read. In Figure 10, since the length of the second line is extremely shorter than the first line or third line, the text is thought to become difficult to read.

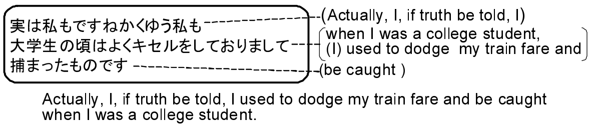

実は私もですねかくゆう私も
大学生の頃はよくキセルをしておりまして
捕まったものです
(Actually, I, if truth be told, I)
when I was a college student,
(I) used to dodge my train fare and
(be caught)
Actually, I, if truth be told, I used to dodge my train fare and be caught when I was a college student.

Figure 9: Example of succession of hiragana

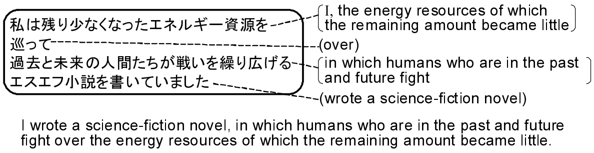

私は残り少なくなったエネルギー資源を
巡って
過去と未来の人間たちが戦いを繰り広げる
エスエフ小説を書いていました
I, the energy resources of which the remaining amount became little
(over)
in which humans who are in the past and future fight
(wrote a science-fiction novel)
I wrote a science-fiction novel, in which humans who are in the past and future fight over the energy resources of which the remaining amount became little.

Figure 10: Lines that have extremely different length

Table 4: Other annotator's results

	recall (%)	precision (%)	F-measure
by human	89.82 (459/511)	89.82 (459/511)	89.82
our method	82.19 (420/511)	81.71 (420/514)	81.95

6.2 Comparison with Linefeeds Inserted by Human

The concept of linefeed insertion for making the caption be easy to read varies by the individual. When multiple people insert linefeeds for the same text, there is possibility that linefeeds are inserted into different points.

Therefore, for one lecture data (128 sentences, 511 bunsetsus) in the experimental data, we conducted an experiment on linefeed insertion by an annotator who was not involved in the construction of the correct data. Table 4 shows the recall and the precision. The second line shows the result of our method for the same lecture data. In F-measure, our method achieved 91.24% (81.95/89.82) of the result by the human annotator.

6.3 Performance of Linefeed Insertion Based on Automatic Natural Language Analysis

In the experiment described in Section 5, we used the linguistic information provided by human as the features on the maximum entropy method. However, compared with baseline 1, our method uses a lot of linguistic information which should be provided not by human but by natural language analyzers under the real situation. Therefore, to fairly evaluate our method and four baselines, we conducted an experiment on linefeed insertion by using the automatically provided information on clause boundaries and dependency structures[5].

Table 5: Experimental results when information of features are automatically provided

	recall (%)	precision (%)	F-measure
our method	77.37 (4,253/5,497)	75.04 (4,253/5,668)	76.18
baseline 1	27.47 (1,510/5,497)	34.51 (1,510/4,376)	30.59
baseline 2	69.51 (3,821/5,497)	48.63 (3,821/7,857)	57.23
baseline 3	84.01 (4,618/5,497)	52.03 (4,618/8,876)	64.26
baseline 4	69.84 (3,893/5,497)	55.60 (3.893/6,905)	61.91

Table 5 shows the result. Compared with Table 3, it shows the decreasing rate of the performance of our method was more than those of four baselines which use simply only basic linguistic information. However, the F-measure of our method was more than 10% higher than those of four baselines.

7 Conclusion

This paper proposed a method for inserting linefeeds into discourse speech data. Our method can insert linefeeds so that captions become easy to read, by using machine learning techniques on features such as morphemes, dependencies, clause boundaries, pauses and line length. An experiment by using transcribed data of Japanese discourse speech showed the recall and precision was 82.66% and 80.24%, respectively, and we confirmed the effectiveness of our method.

In applying the linefeed insertion technique to practical real-time captioning, we have to consider not only the readability but also the simultaneity. Since the input of our method is a sentence which tends to be long in spoken monologue, in the future, we will develop more simultaneous a technique in which the input is shorter than a sentence. In addition, we assumed the speech recognition system with perfect performance. To demonstrate practicality of our method for automatic speech transcription, an experiment using a continuous speech recognition system will be performed in the future.

Acknowledgments

This research was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 20300058) and Young Scientists (B) (No. 21700157) of JSPS, and by The Asahi Glass Foundation.

[5] We used CBAP (Kashioka and Maruyama, 2004) as a clause boundary analyzer and CaboCha (Kudo and Matsumoto, 2002) with default learning data as a dependency parser.

References

G. Boulianne, J.-F. Beaumont, M. Boisvert, J. Brousseau, P. Cardinal, C. Chapdelaine, M. Comeau, P. Ouellet, and F. Osterrath. 2006. Computer-assisted closed-captioning of live TV broadcasts in French. In *Proceedings of 9th International Conference on Spoken Language Processing*, pages 273–276.

T. Holter, E. Harborg, M. H. Johnsen, and T. Svendsen. 2000. ASR-based subtitling of live TV-programs for the hearing impaired. In *Proceedings of 6th International Conference on Spoken Language Processing*, volume 3, pages 570–573.

R. B. Hoogenboom, K. Uehara, T. Kanazawa, S. Nakano, H. Kuroki, S. Ino, and T. Ifukube. 2008. An application of real-time captioning system using automatic speech recognition technology to college efl education for deaf and hard-of-hearing students. *Gunma University Annual Research Reports, Cultural Science Series*, 57.

T. Imai, S. Sato, A. Kobayashi, K. Onoe, and S. Homma. 2006. Online speech detection and dual-gender speech recognition for captioning broadcast news. In *Proceedings of 9th International Conference on Spoken Language Processing*, pages 1602–1605.

H. Kashioka and T. Maruyama. 2004. Segmentation of semantic units in Japanese monologues. In *Proceedings of ICSLT2004 and Oriental-COCOSDA2004*, pages 87–92.

T. Kudo and Y. Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of 6th Conference on Computational Natural Language Learning*, pages 63–69.

S. Matsubara, A. Takagi, N. Kawaguchi, and Y. Inagaki. 2002. Bilingual spoken monologue corpus for simultaneous machine interpretation research. In *Proceedings of 3rd International Conference on Language Resources and Evaluation*, pages 153–159.

T. Monma, E. Sawamura, T. Fukushima, I. Maruyama, T. Ehara, and K. Shirai. 2003. Automatic closed-caption production system on TV programs for hearing-impaired people. *Systems and Computers in Japan*, 34(13):71–82.

C. Munteanu, G. Penn, and R. Baecker. 2007. Web-based language modelling for automatic lecture transcription. In *Proceedings of 8th Annual Conference of the International Speech Communication Association*, pages 2353–2356.

M. Saraclar, M. Riley, E. Bocchieri, and V. Goffin. 2002. Towards automatic closed captioning: Low latency real time broadcast news transcription. In *Proceedings of 7th International Conference on Spoken Language Processing*, pages 1741–1744.

J. Xue, R. Hu, and Y. Zhao. 2006. New improvements in decoding speed and latency for automatic captioning. In *Proceedings of 9th International Conference on Spoken Language Processing*, pages 1630–1633.

L. Zhang. 2008. Maximum entropy modeling toolkit for Python and C++. `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`. [Online; accessed 1-March-2008].