Automatic Linefeed Insertion for Improving Readability of Lecture Transcript

Masaki Murata, Tomohiro Ohno and Shigeki Matsubara

Abstract The development of a captioning system that supports the real-time understanding of monologue speech such as lectures and commentaries is required. In monologues, since a sentence tends to be long, each sentence is often displayed in multi lines on the screen and becomes unreadable. In the case, 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 sentence as an elemental technique to generate the readable captions. Our method inserts linefeeds into a sentence by applying the rules based on morphemes, dependencies and clause boundaries. We established the rules by circumstantially investigating the corpus annotated with linefeeds. An experiment using Japanese monologue corpus has shown the effectiveness of our rules.

1 Introduction

Real-time captioning, which displays transcribed texts of monologue speech such as lectures, is a technique for supporting the speech understanding of deaf persons, elderly persons, or foreigners. In monologues, since a sentence tends to be long, each sentences is often displayed in multi lines on the screen. In the case, it is necessary to insert linefeeds into a text so that the text becomes easy to read.

Masaki Murata
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, 464-8601, Japan, e-mail: murata@el.itc.nagoya-u.ac.jp

Tomohiro Ohno
Graduate School of International Development, Nagoya University, Furo-cho, Chikusa-ku, 464-8601, Japan, e-mail: ohno@nagoya-u.jp

Shigeki Matsubara
Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, 464-8601, Japan, e-mail: matubara@nagoya-u.jp

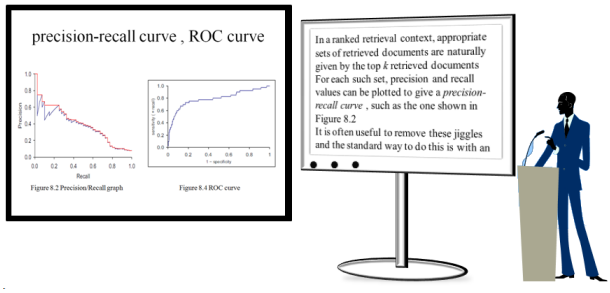

Fig. 1 Caption display of spoken monologue

This paper proposes a technique for inserting linefeeds into a Japanese spoken monologue sentence as an elemental technique to generate readable captions. We assume that a screen which displays only multiline caption is placed to provide the caption information to the audience on the site of lectures and commentaries. In our method, the linefeeds are assumed to be inserted into only the boundaries between *bunsetsus*[1]. Our method applies the rules for inserting linefeeds to a sentence. The rules are created in consideration of the boundary into which linefeeds are not inserted, the boundary into which linefeeds should be inevitably inserted, and the boundary into which linefeeds can be inserted.

We established the rules based on the emerging pattern of morphemes, dependencies and clause boundaries by circumstantially investigating the corpus annotated with linefeeds. We conducted an experiment on inserting linefeeds by using Japanese spoken monologue corpus. As the results, the precision and recall of our method was 82.7% and 79.0%, respectively. Our method improved the performance dramatically compared with the baseline method, which is implemented based on bunsetsu boundaries and the maximum number of characters per line, and has been confirmed to be effective.

2 Caption display of spoken monologue

2.1 Linefeeds insertion in monologue sentences

In our research, as an environment in which captions are displayed on the site of lectures, we assume that a screen for displaying only captions is used. Figure 1 shows our assumed environment in which captions are displayed. One line of the

[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. A *dependency* 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.

Fig. 2 Caption of monologue speech

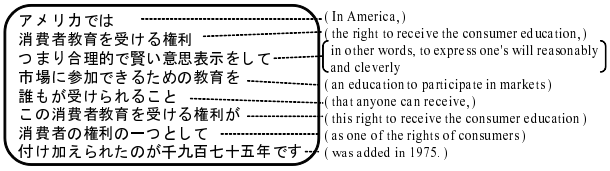

Fig. 3 Caption into which linefeeds are properly inserted

displayed text switches to other line and multiline text is always displayed, being scrolled.

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

In our research, we set the following concepts as the proper points into which linefeeds are inserted on captioning.

- Linefeeds have to be inserted so that each line constitutes a semantically meaningful unit.
- The number of characters in each line have to be less than or equal to the maximum number of characters per line, which is established based on the width of a screen.

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 linefeeds are inserted.

2.2 Related works

There exist a lot of researches about captioning [1, 8, 2]. However, there are few conventional researches about inserting linefeeds on captioning except the following researches. Monma et al.[5] proposed the method for inserting linefeeds based on patterns of a sequence of morphemes. They analyzed the point into which linefeeds were inserted on the closed-captions of Japanese TV shows, and then made the rules

for inserting linefeeds. However, in this research, the linefeeds are inserted on the constraint that the text displayed in a screen all switches other text at a time, that is, the readability in case of our assumed caption display system is not considered.

Saikou et al.[7] proposed the method for captioning based on the gradually chunking. This method chunks morphemes into a "*constituent*," which corresponds to the nominative, predicates, case elements and so on in a sentence, and then chunks "*constituents*" into a "*phrase*." This method can insert linefeeds so that each line becomes a linguistic unit, by concatenating character string of the *constituents*, which do not constitutes a different *phrase* mutually, until the length of each line reaches 15 characters. However, this research did not verified the relation between the proper point into which linefeeds should be inserted, the *constituent* and the *phrase*.

3 Linefeeds in monologue sentences

We investigated the actual spoken monologue data to make the rules for inserting linefeeds based on the concepts described in Section 2.1. In our investigation, we used the spoken monologue corpus "Asu-Wo-Yomu[2]," annotated with information on morphological analysis, clause boundary detection, bunsetsu segmentation, dependency analysis[6], and linefeeds inserted by hands. In what follows, we organize bunsetsu boundaries by classifying them into the following three categories: the boundary into which linefeeds are not inserted, the boundary into which linefeeds should be inevitably inserted, and the boundary into which linefeeds can be inserted.

3.1 Boundaries into which linefeeds are not inserted

As the result of the investigation, we observed that linefeeds were not inserted into the following bunsetsu boundaries.

- The end boundary of the bunsetsu of which the part-of-speech of the rightmost morpheme is "adnominal particle," "case particle-*toyu*," "case particle-*no*," "auxiliary verb," or "adnominal" and depends on the right-hand neighbor bunsetsu.
- The start boundary of the bunsetsu which consists of one of 36 different verbs such as "あります," "しました," which play an auxiliary role and have no corresponding word in English.

3.2 Boundaries into which linefeeds are inevitably inserted

Since a clause constitutes a semantically meaningful language unit, a clause boundary can be widely-accepted as the candidate of the proper point into which linefeeds

[2] Asu-Wo-Yomu is a collection of transcriptions of a TV commentary program of the Japan Broadcasting Corporation (NHK). The commentator speaks on current social issues for 10 minutes.

Table 1 Strong clause boundary

compound clause	*-toka, -ga, -shi, -de, -keredomo, -tari*
condition clause	*-kagiri, -ba, -tara, -kekka, -tokoro*
time clause	*-tokini, -atoni, -ima, -ato, -tokino, -tokiniwa, -sonota*
reason clause	*-node, -kara*
adverbial clause	*-tameniha, -tame, -nagara, -nado, -tewa, -yo, -sonota*
others	Indeclinable words stopping, Interjection

should be inserted. However, the role of each clause on a sentence is different by the types. This means that the likelihood that a linefeed is inserted into a clause boundary is different by the type of the clause boundary. As the result of the above-mentioned analysis, there existed 29 types of clause boundaries into which linefeeds should be inevitably inserted. Table 1 shows the clause boundary types into which linefeeds should be inevitably inserted. We call these clause boundaries the **strong clause boundary** as a whole hereafter. The strong clause boundary accounted for 49.1% of clause boundaries which appear in the analysis data.

3.3 Boundaries into which linefeeds can be inserted

3.3.1 Insertion of linefeeds based on clause boundaries

There exist clause boundaries into which linefeeds are not necessarily inserted but are inserted with high probability in a context. As such a clause boundary type, there are “adverbial clause,” “adverbial adjective clause,” “supplement clause,” “concessive clause-*temo*,” “indirect interrogative.” In this paper, we call these 5 types of a clause boundary the **weak clause boundary**. The weak clause boundary becomes the point into which linefeeds are inserted, if there does not exist the strong clause boundary around it. Furthermore, the clause boundary “condition clause-*to*” and “compound clause-*te*” tend to become the point into which linefeeds are inserted although the tendency is not as great as as that of the strong clause boundary and weak clause boundary.

3.3.2 Insertion of linefeeds based on dependency relations

A dependency relation is a modification relation in which a modifier bunsetsu depends on a modified bunsetsu. A sequence of bunsetsus from the modifier bunsetsu to the modified bunsetsu constitutes a semantically meaningful unit. Therefore, linefeeds tend to be inserted into the end boundaries of modified bunsetsus although the tendency is not greater than that of clause boundaries.

The bunsetsu boundaries, into which linefeeds are most easily to be inserted among the linefeed locations based on dependency relations, the bunsetsu bound-

aries into which linefeeds are easiest to be inserted are the end boundaries of modified bunsetsus of adnominal clauses. Here, in Japanese, the rightmost morpheme of an adnominal clause is congruent with that of a sentence end. Since, if a linefeed is inserted into the end boundary of adnominal clause, the end of the line is misunderstood as a sentence end, a linefeed is inserted not there but into the end boundary of the modified bunsetsu of an adnominal clause.

Since the clause boundary "topicalized element *wa*" does not strictly represent clause boundaries but can be regarded as syntactically independent elements, it is the dominant candidate of the linefeed points. However, in case that the number of characters in the clause boundary "topicalized element *wa*" which appears at the start of a sentence is few like "これは (this)," it is not appropriate to insert a linefeed into the boundary. If the length of the character string between the start of a line and the clause boundary "topicalized element *wa*" is long to some extent, a linefeed tends to be inserted into the clause boundary "topicalized element *wa*."

In addition, the dependency structure of a line displayed as a caption tends to be closed. That is to say, all bunsetsus, except the final bunsetsu, in a line tend to depend on one of bunsetsus in the line. Conversely, a linefeed tends to be inserted into the end boundary of the modified bunsetsu of which the dependency distance is long.

4 Automatic insertion of linefeeds

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 outputs the sentence into which linefeeds are inserted. The insertion of linefeeds is executed as follows by using the rules for deciding the point into which linefeeds should be inserted.

We made the rules for inserting linefeeds based on the analysis described in the previous section. Table 2 shows the rules. The each rule number indicates the priority order in which each rule is applied, and the application of rules is performed in accordance with the priority order until the length of all lines becomes less than or equal to the maximum number of characters per line. The first rule is for the boundaries into which linefeeds are not inserted, and the second is for inevitable insertion. Furthermore, the rules 2-4 are for the insertion based on clause boundaries, 5-9 are on dependency relations, and 10 is on the number of characters of line.

Figure 4 shows the processing flow of linefeed insertion. The candidates of points into which linefeeds are inserted are denoted by a slash "/." First, the end boundaries of the bunsetsus which are the boundaries into which linefeeds are not inserted are excluded from the candidates of linefeed points. Next, a linefeed is inserted into the end boundary of the bunsetsu "受けているので (because they have already been subjected)," which is the strong clause boundary "reason clause-*node*." As mentioned above, the rules for linefeed insertion are applied in accordance with the priority order. The texts of the caption are finally generated so that the length of each line is less than or equal to the maximum number of characters per line.

Table 2 Ten rules for inserting linefeeds

	rule
1	Exclude bunsetsus boundaries into which linefeeds are not inserted from the candidates.
2	Insert into the strong clause boundary.
3	Insert into the weak clause boundary.
4	Insert into the clause boundary "condition clause-*to*."
5	Insert into the clause boundary "compound clause-te."
6	Insert into the end boundary of a modifier bunsetsu of "adnominal clause."
7	Insert into the clause boundary "topicalized element-*wa*," if the number of characters between the start of a line and "*wa*" is over than 70% of the maximum number of characters per line.
8	Insert into the end boundary of the **long dependency distance bunsetsu** in case that there exits only one **long dependency distance bunsetsu**. within the maximum number of characters from the start of the line.
9	Insert into the end boundary of the leftmost bunsetsu among **long dependency distance bunsetsus** of which the modified bunsetsu is different from that of the next one.
10	Insert into the rightmost bunsetsu boundary within the maximum number of characters.

*A bunsetsu which is located within the maximum number of characters from the start of the line and which depends on a bunsetsu located outside the maximum number of characters from the start of the line is called **long dependency distance bunsetsu**.

5 Experiment

To evaluate the effectiveness of our method, we conducted an experiment on inserting linefeeds by using Japanese spoken monologue data.

5.1 Outline of experiment

We used 3 programs (219 sentences, 2,121 bunsetsus) in the syntactically annotated spoken monologue corpus "Asu-Wo-Yomu" as the test data, annotated with information on morphological analysis, clause boundary detection, and dependency analysis by hands.

We applied our method to the test data. In addition, we compared our method with the baseline one, which inserts linefeeds into the rightmost bunsetsu boundary 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. We obtained the precision (the ratio that the points of the inserted linefeeds correspond with the correct ones) and the recall (the ratio that linefeeds were inserted into the correct ones).

Two experts collectively created the correct1 data. They created the data independently and then they decided the correct data through the consultation based on each data. We considered that there are other acceptable linefeed points other than those of the correct data. Therefore, we also obtained the precision in case that the

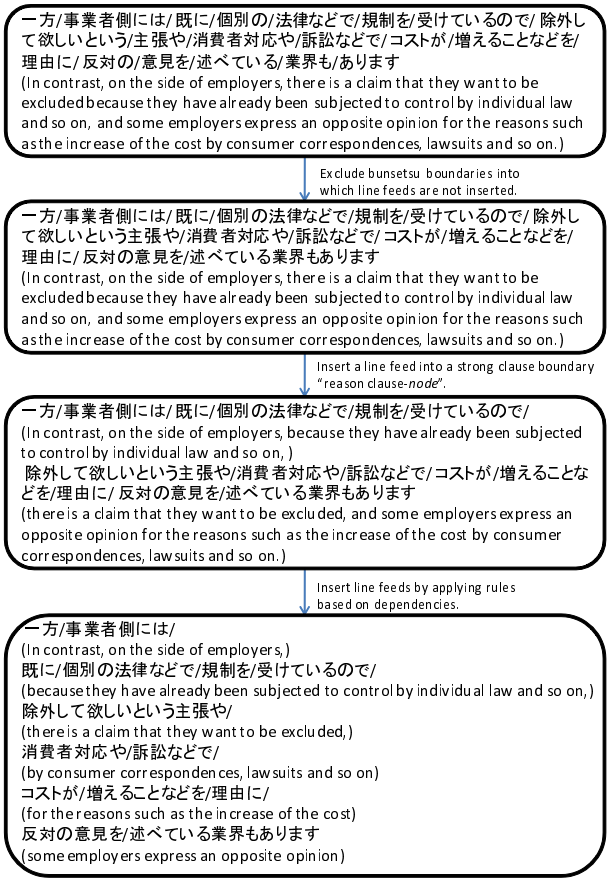

Fig. 4 Processing flow of linefeeds insertion

union of the linefeed points in the two data which two experts created independently is considered to be the correct points (hereinafter called **acceptable precision**).

5.2 Experimental results

Table 3 shows the experimental results. The recall and precision were 82.7% and 79.0% respectively, and we confirmed that our method had higher performance than the baseline method. Furthermore, the acceptable precision was 86.8% (468/539). As mentioned above, we confirmed the effectiveness of our method.

Figure 4 shows the causes of incorrectly inserting linefeeds. The largest cause is the line feed insertion based on the dependency distance. There were a lot of cases that the length of a line becomes short by inserting a linefeed into the end boundary

Table 3 Experimental results

	our method	baseline
recall	82.7% (426/515)	30.1% (155/515)
precision	79.0% (426/539)	37.9% (155/409)

Table 4 Causes of incorrect linefeed insertion

causes	#
exclusion of bunsetsu boundary from the candidates	10
insertion into the strong clause boundary	6
insertion into the weak clause boundary	9
insertion based on “adnominal clause”	1
insertion based on “topicalized element *wa*”	10
insertion based on dependency distance	24
others	11
total	71

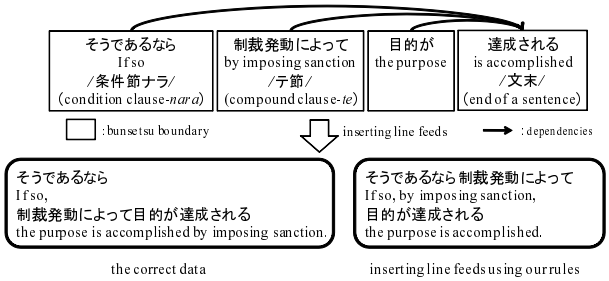

Fig. 5 Example of the clause boundary which did not appear in the analysis data

of a bunsetsu which depends on a distant bunsetsu. We need to establish detailed rules based on not only the dependency distance but also the number of characters in a line and the morphological information such as the part-of-speech of a particle.

On the other hand, one of the reasons for not inserting linefeeds into the correct linefeed points is the existence of the clause boundaries which did not appear in the analysis data. Figure 5 shows an example. In this example, the clause boundary “condition clause-*nara* ” becomes correct linefeed point. However, there did not exist the clause boundary “condition clause-*nara*. ”Therefore, the linefeed was incorrectly inserted into the clause boundary “compound clause-*te*.” Since the current rules are not covered enough, we need to increase the size of the learning data.

6 Conclusions

This paper proposed a method for inserting linefeeds into Japanese monologue sentences to support the understanding of monologue speech by the deaf persons, elderly persons or foreigners. Our method can insert linefeeds so that captions become easy to be read by applying the rules which are established based on the emerging pattern of morphemes, dependencies, clause boundaries, pauses, fillers and so on. An experiment on inserting linefeeds by using monologue corpus showed the recall and precision was 82.7% and 79.0%, respectively, and we confirmed the effectiveness of our method.

To make the rules linefeeds by hands has limitations in enlarging, refining and organizing them. Future research will include considering the automatic acquisition of the rules. In addition, we will plan to reveal the linguistic relation between the points into which linefeeds should be inserted on captioning and the points into which pauses should be inserted on speech synthesis [4].

Acknowledgements This research was partially supported by the Grant-in-Aid for Scientific Research (B) of JSPS and by The Telecommunications Advancement Foundation.

References

1. Boulianne, G., et al. (2006). Computer-Assisted Closed-Captioning of Live TV Broadcasts in French. In *Proceedings of ICSLP-2006* (pp. 273–276).
2. Daelemans, W., Hothker, A., & Sang, E. T. K. (2004). Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of LREC-2004* (pp. 1045–1048).
3. Holter, T., Harborg, E., Johnsen, M. H., & Svendsen, T. (2000). Asr-based subtitling of live TV-programs for the hearing impaired. In *Proceedings of ICSLP-2000* (pp. 570–573).
4. Iwata, K., Mitome, Y., & Watanabe, T. (1990). Pause Rule for Japanese Text-To-Speech Conversion Using Pause Insertion Probability. In *Proceedings of ICSLP-90* (pp. 837–840).
5. Monma, T., Sawamura, E., Fukushima, T., Maruyama, I., Ehara, T., & Shirai, K. (2003). Automatic Closed-Caption Production System on TV Programs for hearing-Impaired People. *Journal of Systems and Computers in Japan* (pp. 71–82).
6. Ohno, T., Matsubara, S., Kashioka, H., Kato, N., & Inagaki, Y. (2006). A Syntactically Annotated Corpus of Japanese Spoken Monologue. In *Proceedings of LREC-2006* (pp. 1590–1595).
7. Saiko, M., Takanashi, K., & Kawahara, T. (2006). Cascaded Chunking of Spontaneous Japanese using Bunsetsu Dependency and Pause Information. *IPSJ SIG Technical Reports* (pp. 19–24). (in Japanese)
8. Xue, J., Hu, R., & Zhao, Y. (2006). New Improvements in Decoding Speed and Latency for Automatic Captioning. In *Proceedings of ICSLP-2006* (pp. 1630–1633).