Automatic Text Formatting for Social Media based on Linefeed and Comma Insertion

Masaki Murata, Tomohiro Ohno and Shigeki Matsubara

Abstract By appearance of social media, people are coming to be able to transmit information easily on a personal level. However, because users of social media generally spend little time on describing information, low-quality texts are transmitted and it blocks the spread of information. On transmitted texts in social media, commas and linefeeds are inserted incorrectly, and it becomes a factor of low-quality texts. This paper proposes a method for automatically formatting Japanese texts in social media. Our method formats texts by inserting commas and linefeeds appropriately. In our method, the positions where commas and linefeeds should be inserted are decided based on machine learning using morphological information, dependency relation and clause boundary information. An experiment using Japanese spoken language texts has shown the effectiveness of our method.

1 Introduction

Recently, social media such as Facebook, Twitter and blog service are appearing. Facebook has over five hundred million users, and Twitter has over one hundred million users. By using social media, those users can spread information easily to people all over the world through interaction on a personal level. However, users generally spend little time on describing information in social media because they

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
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, 464-8601, Japan, e-mail: matubara@nagoya-u.jp

want easy transmission of information, unlike those in newspaper or TV broadcast. Therefore, the transmitted texts tend to be low-quality. Low-quality texts might prevent information from spreading in social media.

This paper proposes a method for automatically formatting Japanese texts in social media. Our method formats texts by inserting commas and linefeeds at proper positions. On transmitted texts in social media, too many or too few commas are often inserted or proper linefeed insertion may not be done, and it becomes a factor of low-quality texts. However, commas and linefeeds play important roles in the readability of texts.

In our method, first, commas are inserted into a Japanese text, and then, linefeeds are inserted into the text into which commas were inserted. The comma insertion method decides whether or not to insert a comma at each *bunsetsu*[1] boundary by machine learning. The linefeed insertion is also executed by the similar technique.

We conducted an experiment on comma and linefeed insertion using Japanese spoken language data and confirmed the effectiveness of our method.

2 Text Formatting by Comma and Linefeed Insertion

In social media, while users can transmit information easily, users spend little time on describing information. Therefore, the transmitted texts tend to be low-quality. Automatic text formatting is desired so that texts become easy to read.

As related works, there have been researches on automatic text summarization [1], conversion of spoken language into written language [2] and conversion of spoken documents into web documents [3]. In these works, text formatting is realized by editing strings in the texts. However, it is important not only to be written in simple words or simple structure, but also to be displayed properly so that texts are easy to read.

In our work, we focus on comma and linefeed insertion as the method for text formatting. For Japanese, which is a non-segmented language, it is necessary to segment texts properly by commas and linefeeds so that the texts are easy to read.

2.1 Text Formatting by Comma Insertion

There are several usages of commas such as making clear sentence structures or marking word boundaries, and the positions where commas are inserted depend on these usages. It is important to insert commas at proper positions in accordance with the usages of commas. However, while there is a tendency about the positions where commas should be inserted, there is no clear standard for these positions. Therefore, positions where commas are inserted could be different in individuals. In social

[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.

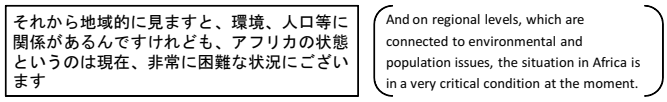

Fig. 1 Text into which linefeeds are inserted forcibly

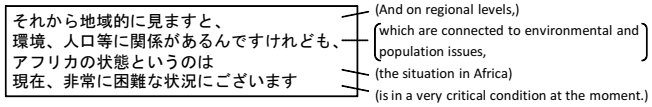

Fig. 2 Text into which linefeeds are inserted properly

media, texts into which commas were inserted incorrectly are created. Because such texts become hard to read, it is required to format these texts by inserting commas at proper positions.

Newspaper articles are one of the texts into which commas were inserted properly. Examples of a text in newspaper articles are shown below:

- 世界の成長センターとなったアジアで、急浮上する中国の存在は、希望にあふれていると同時に、困難な課題も山積している。(In Asia, which becomes a center of the growth of the world, the strong presence of China not only brings hope but also causes difficult problems.)
- むしろ地球規模の環境、人口、食糧など広範に国連の果たさなければならない役割は大きい。 (The United Nations should play a lot of roles in a broad range of fields, such as the global environment, population, and food.)

It is possible to improve the readability of web texts by inserting commas at positions where commas are inserted in newspaper articles.

2.2 Text Formatting by Linefeed Insertion

In order to generate readable texts, it is important how to display texts. In the text in Figure 1, how to display is not considered, and linefeeds are forcibly inserted according to the width. Therefore, the text is not easy to read. In fact, such the texts are written in newspaper articles because newspaper have space constraints. On the other hand, the space constraints are not so hard on the web. Therefore, to display texts into which linefeeds are inserted appropriately is effective.

By inserting linefeeds at semantic boundaries, the texts become easy to read. Figure 2 shows the text into which linefeeds are inserted at proper positions. Linefeeds are inserted at the semantic boundary “関係があるんですけれども (are connected)” and right after the subject of the sentence “状態というのは (the situation)”.

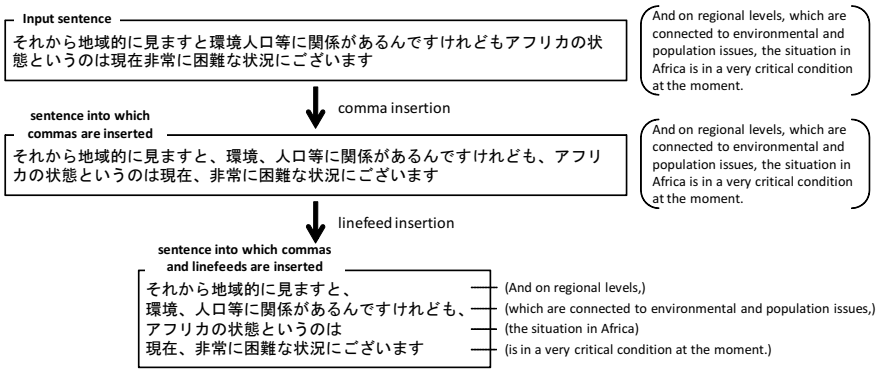

Fig. 3 Example of text formatting by our method

3 Text Formatting Method

Our method formats texts by inserting commas and linefeeds appropriately. In social media, texts into which commas and linefeeds are inserted incorrectly can be created. Therefore, our method ignores these inserted commas and linefeeds, and inserts commas and linefeeds into texts which have no commas and linefeeds.

In our method, we use the comma insertion method [4] and the linefeed insertion method [5]. Figure 3 shows an example of text formatting by our method. First, our method inserts commas into a Japanese sentence, and then, inserts linefeeds so that the number of characters in each line can be less than or equal to the maximum number of characters per line.

3.1 Comma Insertion Method

In comma insertion method, a sentence, on which morphological analysis, bunsetsu segmentation, clause boundary analysis and dependency[2] analysis have been performed, is considered the input. Our method decides whether or not to insert a comma at each bunsetsu boundary in an input sentence. Comma insertion method identifies the most appropriate combination among all combinations of positions where a comma can be inserted, by using the probabilistic model.

In this paper, input sentences which consist of n bunsetsus are represented by $B = b_1 \cdots b_n$, and the results of comma insertion by $R = r_1 \cdots r_n$. Here, r_i is 1 if a comma is inserted right after bunsetsu b_i, and 0 otherwise. Also, $r_n = 1$. We indicate the j-th sequence of bunsetsus created by dividing an input sentence into m sequences as $L_j = b_1^j \cdots b_{n_j}^j (1 \leq j \leq m)$, and then, $r_k^j = 0$ if $1 \leq k < n_j$, and $r_k^j = 1$ if $k = n_j$.

[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.

3.1.1 Probabilistic Model for Comma Insertion

When an input sentence B is provided, our method identifies the comma insertion R that maximizes the conditional probability $P(R|B)$. Assuming that whether or not to insert a comma right after a bunsetsu is independent of other commas except the one appearing immediately before that bunsetsu, $P(R|B)$ can be calculated as follows:

$$
\begin{aligned}
& P(R|B) \qquad (1)\\
=& P(r_1^1=0,\cdots,r_{n_1-1}^1=0,r_{n_1}^1=1,\cdots,r_1^m=0,\cdots,r_{n_m-1}^m=0,r_{n_m}^m=1|B)\\
\cong& P(r_1^1=0|B)\times\cdots\times P(r_{n_1-1}^1=0|r_{n_1-2}^1=0,\cdots,r_1^1=0,B)\\
& \times P(r_{n_1}^1=1|r_{n_1-1}^1=0,\cdots,r_1^1=0,B)\times\cdots\\
& \times P(r_1^m=0|r_{n_{m-1}}^{m-1}=1,B)\times\cdots\times P(r_{n_m-1}^m=0|r_{n_m-2}^m=0,\cdots,r_1^m=0,r_{n_{m-1}}^{m-1}=1,B)\\
& \times P(r_{n_m}^m=1|r_{n_m-1}^m=0,\cdots,r_1^m=0,r_{n_{m-1}}^{m-1}=1,B)
\end{aligned}
$$

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 comma is inserted right after a bunsetsu b_k^j when the sequence of bunsetsus B is provided and the position of j-th comma 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 comma 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 comma insertion, and calculated by dynamic programming.

3.1.2 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 features in Table 1. To decide the features, we grouped the usages of commas, and investigated the appearance tendency for each category by using Japanese newspaper articles. For more details, please refer to the literature [4].

3.2 Linefeed Insertion Method

We adopt the same method as comma insertion for linefeed insertion. A sentence, on which morphological analysis, bunsetsu segmentation, clause boundary analysis and dependency analysis have been performed, is considered the input. Linefeed insertion method decides whether or not to insert a linefeed at each bunsetsu boundary in an input sentence. Under the construction that the number of characters in each line has to be less than or equal to the maximum number of characters per line, linefeed insertion method identifies the most appropriate combination among all combinations of the positions into which linefeeds can be inserted, by using the probabilistic model. To estimate $P(r_k^j=1|r_{k-1}^j=0,\cdots,r_1^j=0,r_{n_{j-1}}^{j-1}=1,M)$ and

Table 1 Features used for the maximum entropy method (comma insertion)

morphological information	the rightmost independent morpheme, i.e. head word, (part-of-speech and inflected form) and rightmost morpheme (part-of-speech) of a bunsetsu b_k^j
	the rightmost morpheme (a surface form) of b_k^j if the rightmost morpheme is a particle
	the first morpheme (part-of-speech) of b_{k+1}^j
commas inserted between clauses	whether or not a clause boundary exists right after b_k^j
	type of a clause boundary right after b_k^j if there exists a clause boundary
commas indicating clear dependency relations	whether or not b_k^j depends on the next bunsetsu
	whether or not b_k^j depends on a bunsetsu located after the final bunsetsu of the clause including the next bunsetsu of b_k^j
	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 is closed
commas avoiding reading mistakes and reading difficulty	whether or not both the rightmost morpheme of b_k^j and first morpheme of b_{k+1}^j are *kanji* characters
	whether or not both the rightmost morpheme of b_k^j and first morpheme of b_{k+1}^j are *katakana* characters
commas indicating the subject	whether or not there exists a clause boundary "topicalized element-*wa*" right after b_k^j and b_k^j depends on the next bunsetsu
	whether or not there exists a clause boundary "topicalized element-*wa*" right after b_k^j and the string of characters right before b_k^j is "では (*dewa*)"
	the number of characters in a phrase indicating the subject[3] if there exists a clause boundary "topicalized element-*wa*" right after b_k^j
	whether or not a clause boundary "topicalized element-*wa*" exists right after b_k^j and a bunsetsu whose rightmost morpheme is a verb depends on the modified bunsetsu of b_k^j
commas inserted after a conjunction or adverb at the beginning of a sentence	whether or not b_k^j appears at the beginning of a sentence and its rightmost morpheme is a conjunction
	whether or not b_k^j appears at the beginning of a sentence and its rightmost morpheme is an adverb
commas inserted between parallel words or phrases	whether or not both the rightmost morphemes of b_k^j and b_{k+1}^j are nouns
	whether or not a predicate at the sentence end is depended on by b_k^j whose rightmost independent morpheme is a verb and by any of the bunsetsus which are located after b_k^j and of which the rightmost independent morpheme is a verb
number of characters from b_1^j to b_k^j	one of the following 4 categories if the number of characters from b_1^j to b_k^j is found there ([num = 1], [2 ≤ num ≤ 3], [4 ≤ num ≤ 21], [22 ≤ num])

$P(r_k^j = 0 | r_{k-1}^j = 0, \cdots, r_1^j = 0, r_{n_{j-1}}^{j-1} = 1, M)$ by the maximum entropy method, we used the features in Table 2. The features were decided based on an analysis of linefeed data.For more details, please refer to the literature [5].

[3] Phrases indicating the subject is a sequence of bunsetsus consisting of b_k^j and all bunsetsus that are connected to b_k^j when we trace their dependency relationship in modifier-to-modifyee direction.

Table 2 Features used for the maximum entropy method (linefeed insertion)

Morphological information	the rightmost independent morpheme i.e. head word, (part-of-speech and inflected form) and rightmost morpheme (part-of-speech) of a bunsetsu b_k^j
Clause boundary information	whether or not a clause boundary exists right after b_k^j
	type of a 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	one of the following 3 categories if the number of characters from the start of the line to b_k^j is found there ([num $\leq$ 2], [3 $\leq$ num $\leq$ 6], [7 $\leq$ num])
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 following morphemes (Basic form: “思う (think)”, “問題 (problem),” “する (do),” “なる (become),” “必要 (necessary)” Part-of-speech: noun-non independent-general, noun-nai adjective stem, noun-non independent-adverbial)
Comma	whether or not a comma exists right after b_k^J

4 Experiment

We conducted an experiment on inserting commas and linefeeds. We assume that our method can be used to format various types of texts. In the experiment, we evaluate the effectiveness of our method by using Japanese spoken language data.

4.1 Outline of Experiment

As the experimental data, we used the Japanese spoken language texts in the SIDB [6]. All the data were annotated with information on morphological analysis, dependency analysis and clause boundary detection by hand. The correct data of comma and linefeed insertion were created by inserting commas and linefeeds properly into this data by hand. Table 3 shows the size of the correct data. We performed a 16-fold cross-validation experiment by using this data. However, since we used 221

Table 3 Size of correct data

sentence	bunsetsu	comma	linefeed
1,935	23,598	4,833	5,841

Table 4 Experimental result

	recall	precision	F-measure
commas	71.65% (2,899/4,046)	81.07% (2,899/3,576)	76.07
linefeeds	76.91% (3,878/5,042)	69.19% (3,878/5,605)	72.85

sentences among 1,935 sentences as the analysis data in our research [5], we evaluated the experimental result for the other 1,714 sentences (20,707 bunsetsus). Here, we used the maximum entropy method tool [7]. As for options, the repeat count on learning algorithm is set to 2000, and other options are set to default. In the experiment, we defined the maximum number of characters per line as 20.

In the evaluation, we obtained the recall and precision. The recall and precision of comma insertion are respectively defined as follows.

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

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

The recall and precision of linefeed insertion are defined similarly.

4.2 Experimental Result

Table 4 shows the experimental result. Both F-measures were higher than 70, which showed an effectiveness of our method.

Figure 4 shows the results of comma and linefeed insertion into the following two sentences by our method:

- 姉の結婚式にまつわりましてまずパンフレットの作成それから私は母親の看病そして会社に行っておりましたので仕事の方をしたりとか毎日大変な雑用とかをしておりました (As for my sister's wedding, I was taking charge of doing various things every day, such as making a brochure for the wedding ceremony, nursing my mother, and going to work as a salaried worker.)
- トースター用のマイクロコードつまりソフトウェアですがそれと衛星のフライトダイナミックス航空力学のソフトですねこれを作ろうというのに同じプロセスメソッド組織ツールそれらを使って両方作ってしまおうとは考えないでしょう (The micro code for toasters means software for toasters and the flight dynamics for satellites means software for aeronautic dynamics. In order to manufacture both of them, you may not think you would use the same process method, the same organization and the same tools.)

As shown in Figure 4, readable texts have been generated by our method.

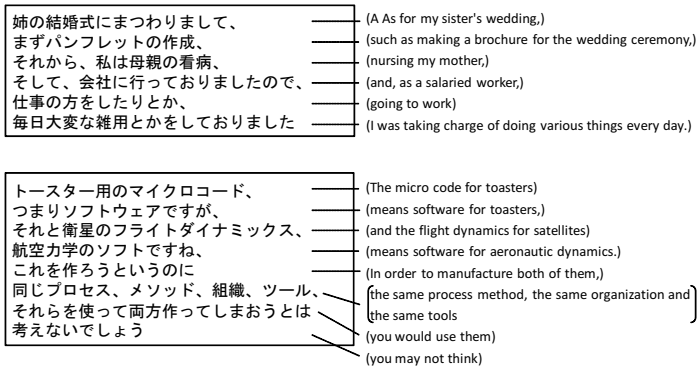

Fig. 4 Examples of comma and linefeed insertion by our method

Table 5 Recalls at positions where a comma and a linefeed existed in the evaluation data

commas	79.24% (1,996/2,519)
linefeeds	87.89% (2,214/2,519)

4.3 Discussion

In this subsection, we focus on the relation between positions where commas were inserted and positions where linefeeds were inserted in the evaluation data.

Among positions where a comma existed (4,046 positions) and positions where a linefeed existed (5,042 positions) in the evaluation data, there existed 2,519 positions where a comma and a linefeed were existed. Table 5 shows the recall of comma insertion and the recall of linefeed insertion at these positions. Each recall was higher than the recall shown in Table 4. This is because positions where a comma and a linefeed are inserted may be semantic boundaries, and, our features can capture these positions well. Also, there existed 1,527 positions where only a comma existed, and the recall at these positions was 59.14% (903/1,527). And, there existed 2,467 positions where only a linefeed existed, and the recall at these positions was 65.95% (1,664/2,523). Each recall was lower than the recall shown in Table 4.

Among positions where commas existed in the evaluation data, there existed 462 positions where our method inserted linefeeds incorrectly. In case that linefeeds are inserted at positions where commas existed in the evaluation data, short lines will be generated unnecessarily and the texts will become hard to read. On the other hand, among positions where linefeeds existed in the evaluation data, there existed 273 positions where our method inserted commas incorrectly. Commas inserted too much harm readability of the texts. Because our method inserts commas and linefeeds sequentially, our method cannot consider linefeeds when our method decides whether or not to insert a comma. To realize more flexible comma and linefeed insertion, to develop a method which inserts commas and linefeeds at the same time is desired.

5 Conclusion

This paper proposed a method for formatting Japanese texts in social media. Our method formats texts by inserting commas and linefeeds at proper positions. In our method, commas and linefeeds are inserted into a sentence by machine learning, based on the information such as morpheme, dependencies and clause boundaries. An experiment by using Japanese spoken language data showed F-measure of comma insertion was 76.07 and F-measure of linefeed insertion was 72.85, and we confirmed the effectiveness of our method.

Our method identifies positions where commas are inserted first, and then, inserts linefeeds. Therefore, our method cannot judge a bunsetsu boundary as a position where a comma and a linefeed are inserted, or where a comma is not inserted but a linefeed is inserted at the same time. However, humans would not insert commas and linefeeds sequentially. In the future, we will develop more flexible method which inserts commas and linefeeds at the same time.

Acknowledgements This research was partially supported by the Grant-in-Aids for Scientific Research (B) (No. 22300051) and Young Scientists (B) (No. 21700157), and by the Continuation Grants for Young Researchers of The Asahi Glass Foundation.

References

1. Knight, K., & Marcu D. (2002). Summarization beyond sentence extraction: a probabilistic approach to sentence compression. Artificial Intelligence (Vol.139, No.1).
2. Shitaoka, K., Nanjo, H. & Kawahara, T. (2004). Automatic Transformation of Lecture Transcription into Document Style Using Statistical Framework. In Proceedings of 8th International Conference on Spoken Language Processing (pp. 2169-2172).
3. Ito, M., Ohno, T., & Matsubara, S. (2009). Text-Style Conversion of Speech Transcript into Web Document for Lecture Archive. Journal of Advanced Computational Intelligence and Intelligent Informatics (Vol. 13, No. 4, pp. 499-505).
4. Murata, M., Ohno, T., & Matsubara, S. (2010). Automatic Comma Insertion for Japanese Text Generation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 892-901).
5. Ohno, T., Murata, M., & Matsubara, S. (2009). Linefeed Insertion into Japanese Spoken Monologue for Captioning. In Proceedings of Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (pp. 531-539).
6. Matsubara, S., Takagi, A., Kawaguchi N., & Inagaki, Y. (2002). Bilingual Spoken Monologue Corpus for Simultaneous Machine Interpretation Research. In Proceedings of Language Resources and Evaluation Conference. (pp. 153-159).
7. Le, Z. (2008). Maximum entropy modeling toolkit for python and c++. `http://homepages.inf.ed.ac.uk/s0450736/maxenttoolkit.html` [Online; accessed 1-March-2008].