DE GRUYTER OPEN

JAISCR, 2015, Vol. 5, No. 4, pp. 247 – 255
10.1515/jaiscr-2015-0033

ORDER ESTIMATION OF JAPANESE PARAGRAPHS BY SUPERVISED MACHINE LEARNING AND VARIOUS TEXTUAL FEATURES

Masaki Murata[1], Satoshi Ito[1], Masato Tokuhisa[1] and Qing Ma[2]

[1] *Department of Information and Electronics, Tottori University, 4-101 Koyama-Minami, Tottori 680-8552, Japan*

[2] *Department of Applied Mathematics and Informatics, Ryukoku University Seta, Otsu, Shiga 520-2194, Japan*

Abstract

In this paper, we propose a method to estimate the order of paragraphs by supervised machine learning. We use a support vector machine (SVM) for supervised machine learning. The estimation of paragraph order is useful for sentence generation and sentence correction. The proposed method obtained a high accuracy (0.84) in the order estimation experiments of the first two paragraphs of an article. In addition, it obtained a higher accuracy than the baseline method in the experiments using two paragraphs of an article. We performed feature analysis and we found that adnominals, conjunctions, and dates were effective for the order estimation of the first two paragraphs, and the ratio of new words and the similarity between the preceding paragraphs and an estimated paragraph were effective for the order estimation of all pairs of paragraphs.

1 Introduction

The estimation of sentence order (sometimes referred to as sentence ordering) is a problem that stems from sentence generation and sentence correction [7, 10, 13]. When generating text that consists of multiple sentences/paragraphs, arranging them in an appropriate order is necessary to understand the text easily. In this study, we employ supervised machine learning to estimate the appropriate order. In addition, we utilize a high-performance support vector machine (SVM) for supervised learning.[1]

Previous studies of the sentence/paragraph order estimation with supervised learning include research by Uchimoto et al. [16] and Hayashi et al. [6], considering word order and sentence order estimations, respectively. Thus, we consider the order estimation of paragraphs.

In this study, we consider two types of problems: original order and reverse order for pairs of paragraphs extracted from a corpus (newspapers). We determine the correct order by machine learning. Furthermore, we analyze features that facilitate paragraph order estimation. This study is conducted in Japanese paragraphs.

The characteristics of this study are described as follows.

– This study employs supervised learning for paragraph order estimation.

– In our supervised method, training data can be automatically constructed from a corpus (with-

[1] This paper is an extended version of our previous conference paper [14].

out tags). Our method does not require a manual construction of training data.

– In our proposed method using supervised learning, we can find an important information in paragraph order estimation by examining the features. In our experiments, we found that adnominals, conjunctions, dates, the ratio of new words and the similarity between the preceding paragraphs and an estimated paragraph were effective for paragraph order estimation.

– When estimating the order of the first two paragraphs, we obtained a high accuracy rate (0.84) using the proposed method.

– When estimating the order of two adjacent paragraphs and the order of two paragraphs (pairs of all paragraphs), the accuracy rates of the proposed method were 0.62 and 0.64, respectively. These are higher than those of the baseline method assuming that a paragraph having more nouns in common with the preceding paragraphs is likely to be the first of the pair.

2 Related work

Uchimoto et al. performed a study of sentence generation to estimate the order of words on the basis of the phrase dependency information using the maximum entropy method [16]. They assumed that the word order in a corpus is correct and therefore built the training data for the word order from the corpus. Their method does not require a manual construction of training data.

For sentence order estimation in newspaper articles, Hayashi et al. performed a study employing supervised machine learning with a large number of features [6]. They selected two sentences from newspaper articles as a pair and generated one sentence pair in the original order (positive example) and another in the reverse order (negative example). They estimated sentence order by judging whether a sentence pair was positive or negative using supervised machine learning. In addition, they referred to a study by Uchimoto et al. and automatically constructed the data for machine learning from a corpus. In their experiments, they utilized three cases for order estimation: the first two sentences in a paragraph, two adjacent sentences in a paragraph, and all pairs of sentences in a paragraph. Furthermore, they compared their results with those from Lapata's study using a probability technique [9] and reported that they obtained higher performance than Lapata's method.

The aforementioned studies considered word or sentence order estimation. In contrast, our study considers paragraph order.

Lapata regarded existing sentences as training data and calculated the probabilities of features appearing in two adjacent sentences in the training data [9]. By utilizing the total product of probabilities, she calculated the probability that the second sentence was placed after the first sentence and determined the sentence order based on the probability. She utilized verb order, common nouns, and sentence structures from two sentences as features.

In her study, she did not employ machine learning for order estimation. In contrast, our study employs machine learning.

For constructing summaries from multiple documents [2, 3, 12, 15], Okazaki et al. performed a study to estimate the order of extracted sentences [15]. By considering the order of sentences in an original text prior to constructing a summary, they estimated the order of extracted sentences by utilizing the original order. Danushka et al. also studied sentence order estimation for constructing summaries from multiple documents [2]. Their estimation employed supervised machine learning with various features, such as time information, the semantic closeness of content, and the order of sentences in the original documents before constructing summaries.

In these studies, the information from the original documents was utilized before constructing summaries. In contrast, our study does not utilize such information. If the sentence/paragraph order can be estimated without such information, we can use the method also for tasks other than summarization, which include the correction of sentences/paragraphs that are not in an appropriate order.

For knowledge extraction, Agrawal et al. [1] and Giannotti et al. [5] extracted frequent sequential patterns. The data used in frequent sequential pattern mining is a collection of time-stamped item sets, e.g. customers' purchases, logged web accesses, etc. These studies are similar to our study in handling data sets related to order. However, data sets are different between their studies and our study. The data set handled in their studies is a collection of time-stamped item sets. In contrast, the data set handled in our study is a collection of paragraphs. In addition, the following differences exist. Their studies extracted frequent patterns. In contrast, our study determined the order of paragraphs.

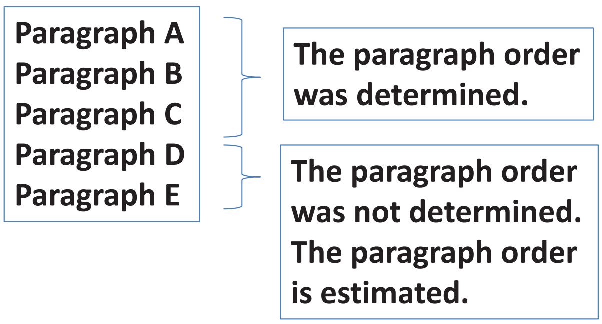


Figure 1. The model of the task

3 Task and proposed method

3.1 The task

The task in this study is as follows. An article is the input and the order of only the first several paragraphs is determined. The order of the remaining paragraphs is not determined. The task is to estimate the order of two paragraphs among the remaining undetermined paragraphs. The information that can be utilized for estimation are the two paragraphs to be estimated and the paragraphs that precede the two target paragraphs (see Figure 1).

3.2 Proposed method

We need to estimate the order of two paragraphs: A and B. These paragraphs are the input to the system, and our method judges whether the order "A→B" is correct by employing SVM.

The training and test data are composed of two paragraphs extracted from a text. From these paragraphs, we construct two sequences: original order and reverse order. The paragraphs in the original order are a positive example, and the paragraphs in reverse order are a negative example. We refer to the studies performed by Uchimoto et al. [16] and Hayashi et al. [6] and automatically construct the training and test data for machine learning from a corpus by assuming that the paragraph order in the corpus is correct.

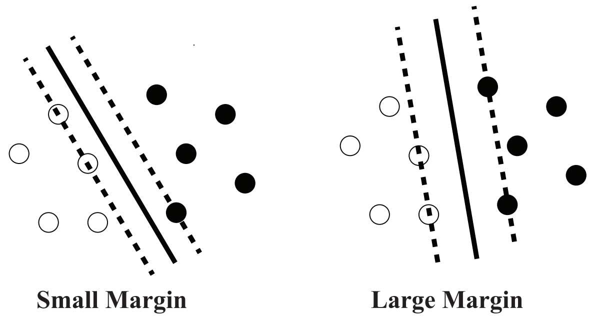


Figure 2. Maximizing the margin

3.3 Support vector machine

In this section, we explain the SVM that we use for machine learning.

In SVM, data consisting of two categories are classified by dividing space with a hyperplane. When the margin between examples that belong to one category and the other category in the training data is larger (see Figure 2[2]), the probability of incorrectly selecting categories in open data is believed to be smaller. The hyperplane maximizing the margin is determined, and classification is done by using this hyperplane. Although the basics of the method are as described above, for extended versions of the method in general, the inner region of the margin in the training data can include a small number of examples, and the linearity of the hyperplane is converted to nonlinearity by using kernel functions. Classification in the extended methods is equivalent to classification using the following discernment function, and the two categories can be classified on the basis of whether the output value of the function is positive or negative [4, 8]:

$$
\begin{aligned}
f(\mathbf{x}) &= sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \qquad (1)\\
b &= -\frac{max_{i,y_i=-1} b_i + min_{i,y_i=1} b_i}{2}\\
b_i &= \sum_{j=1}^{l} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i),
\end{aligned}
$$

[2]In the figure, the white and black circles indicate examples that belong to one category and the other category, respectively. The solid line indicates the hyperplane dividing space, and the broken lines indicate planes at the boundaries of the margin regions.

where $\mathbf{x}$ is the context (a set of features) of an input example; $\mathbf{x}_i$ and $y_i (i = 1, ..., l, y_i \in \{1, -1\})$ indicate the context of the training data and its category, respectively; the function *sgn* is defined as follows:

$$sgn(x) = \begin{cases} 1 & (x \geq 0), \\ -1 & (otherwise). \end{cases} \quad (2)$$

Each $\alpha_i (i = 1, 2...)$ is fixed when the value of $L(\alpha)$ in Equation (3) is maximum under the conditions of Equations (4) and (5).

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \quad (3)$$

$$0 \leq \alpha_i \leq C \; (i = 1, ..., l) \quad (4)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (5)$$

K is called a kernel function. Various types of kernel functions can be used; however, in this paper, we use a polynomial function as follows:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d, \quad (6)$$

where C and d are constants set by experimentation. In this paper, C and d are fixed as 1 and 2 for all experiments, respectively.[3] A set of $\mathbf{x}_i$ that satisfies $\alpha_i > 0$ is called a support vector, and the portion used to perform the sum in Equation (1) is calculated by only using examples that are support vectors. We used the software TinySVM [8], developed by Kudoh, as the SVM.

Table 1. Features

ID	Explanation
a1	Words and their parts of speech (POS) in paragraph A (or B).
a2	Words and their POS in the first-half (or second-half) parts of sentences that are divided by a Japanese postpositional particle *wa* in paragraph A (or B).
a3	Whether an adnominal or a conjunction appears at the beginning of paragraph A (or B).
a4	Whether a date (day) appears in paragraph A (or B).
a5	The number of nouns appearing in paragraphs A and B.
a6	The number of nouns appearing in paragraph B (or A) and not appearing in paragraph A (or B).
a7	The difference between the values of a6 when A and B are exchanged.
a8	The number of nouns appearing in paragraph A (or B) and in the first-half parts of sentences that are divided by a Japanese postpositional particle *wa* in paragraph B (or A).
a9	The number of nouns appearing in the first-half parts of sentences that are divided by a Japanese postpositional particle *wa* in paragraph B (or A) and not appearing in paragraph A (or B).
a10	The difference between the values of a8 when A and B are exchanged.
a11	The difference between the values of a9 when A and B are exchanged.
a12	The number of nouns appearing in paragraph A (or B) and in the paragraphs before paragraphs A and B.
a13	The number of nouns appearing in the paragraphs before paragraphs A and B and not appearing in paragraph A (or B).
a14	The difference between the values of a12 when A and B are exchanged.
a15	The difference between the values of a13 when A and B are exchanged.
a16	The number of nouns appearing in the first-half parts of a2 of paragraph A (or B) and in the paragraphs before paragraphs A and B.
a17	The number of nouns appearing in the paragraphs before paragraphs A and B and not appearing in the first-half parts of a2 of paragraph A (or B).
a18	The difference between the values of a16 when A and B are exchanged.

[3] We confirmed that d = 2 produced good performance in preliminary experiments.

Table 2. Features

ID	Explanation
a19	The difference between the values of a17 when A and B are exchanged.
a20	The difference between the number of words (new words) not appearing in the paragraphs before two paragraphs A and B and appearing in paragraph A and the number of words not appearing in the paragraphs before paragraphs A and B and appearing in paragraph B.
a21	The difference between the ratio of new words appearing in paragraph A and that appearing in paragraph B.
a22	The number of words appearing in the last sentence of paragraph A and in the first sentence of paragraph B.
a23	The difference between the values of a22 when A and B are exchanged.
a24	The number of words appearing in paragraph A and in paragraph B where the number is weighted using appearing places.
a25	The difference between the values of a24 when A and B are exchanged.
a26	The number of words appearing in the last sentence of the paragraphs before paragraphs A and B and in the first sentence of paragraph A (or B).
a27	The ratio between the values of a26 when A and B are exchanged.
a28	The number of words appearing in the paragraphs before paragraphs A and B and in paragraph A (or B) where the number is weighted using appearing places.
a29	The ratio between the values of a28 when A and B are exchanged.

3.4 Features used in our proposed method

Here, we explain features (information utilized for classification) that are required for machine learning. The features utilized in this study are shown in Tables 1 and 2. Each feature has additional information indicating whether it appears in the first or second paragraph of the two target paragraphs, denoted as A and B, respectively. To extract words and parts of speech, we utilize the *ChaSen* morphological analyzer [11]. All features are binary-valued.

Some features are explained in more detail as follows.

a1: Words and their parts of speech (POS) in paragraph A (or B)

The parts of speech used in a1 are a noun, an adjective, an adjectival noun, a verb, an adverb, an adnominal, and a conjunction. Only the words whose parts of speech are the same as those above are used as a feature of a1.

a2: Words and their POS in the first half (or second half) of sentences that are divided by a Japanese postpositional particle *wa* in paragraph A (or B)

Because a paragraph comprises plural sentences, a Japanese postpositional particle *wa* often occurs in a paragraph. We divide a paragraph into component sentences. In a sentence including a particle *wa*, we divide the sentence into two parts, the first half and the second half, using the particle *wa*. A sentence without a particle *wa* is entirely handled as a second-half part. We use words and their POS in the first-half parts of sentences as features and those in the second-half parts of sentences as different features. In Japanese, old information is described in the part before *wa*, whereas new information is described in the part after *wa*. The old and new information is related to paragraph order; thus, we use this feature a2 in our method.

a3: Whether an adnominal or a conjunction appears at the beginning of paragraph A (or B)

When a demonstrative (including *kono* (This), *sono* (Its), and so on) appears at the beginning of a paragraph, it must refer to a word appearing beforehand. In addition, when a conjunction (including *matawa* (Otherwise), *shikashi* (However), and so on) is used, it must be used for the relationship with a previous context. Therefore, when an adnominal or a conjunction appears in the beginning of a sentence, it is believed that there is a preceding paragraph.

a4: Whether a date (day) appears in paragraph A (or B)

When a paragraph includes important events in newspaper articles, a date (day) is likely to be written in the paragraph. Important events are likely to be written at the beginning of the article. Therefore, a paragraph where a date (day) is written is likely to appear at the beginning of the article. To exploit this tendency, we use feature a4.

a5: The number of nouns appearing in paragraphs A and B

Many nouns appear in a paragraph. From this fact, we make a feature to observe the number of nouns (called the common noun number) appearing in both paragraphs A and B. Based on the common noun number, we make the following 15 cases: a range of more than 0, more than 1, ..., more than 9, a range of 0-1, 2-3, ..., 6-7, and a range of more than 7. These 15 cases are used as features.

a6: The number of nouns appearing in paragraph B (or A) and not appearing in paragraph A (or B)

We calculate the number of nouns appearing in paragraph B (or A) and not appearing in paragraph A (or B). We make some cases based on the number and use them as features, as in a5.

a12: The number of nouns appearing in paragraph A (or B) and in the paragraphs before paragraphs A and B

When the content of adjacent paragraphs is similar, the paragraph order is better estimated. From this, we establish the feature a12 such that, between A and B, the paragraph in which there are more common nouns with all preceding paragraphs is judged to appear earlier.

a20: The difference between the number of words (new words) not appearing in the paragraphs preceding A and B and appearing in paragraph A, and the number of words not appearing in the paragraphs preceding A and B and appearing in paragraph B

We first calculate the number of words (called new words) not appearing in the paragraphs preceding A and B and appearing in paragraph A. i.e. we calculate the number of new words first appearing in paragraph A. We call this number N_A. We also calculate the same kind of number against paragraph B. We call this number N_B and calculate $N_A - N_B$. Based on the calculated results, we make the following cases; a range of less than 0 and a range of more than 0. The two cases are used as features. In this feature a20, we use only words whose parts of speeches are used in a1.

a21: The difference between the ratio of new words appearing in paragraph A and in paragraph B

We calculate the ratio of new words appearing in paragraph A (or B). We call the ratio R_A (or R_B). Here, the ratio of new words is the resultant value dividing the number of new words by the number of all words appearing in the paragraph. We calculate $R_A - R_B$. Based on this, we make features such that a paragraph whose ratio of new words is larger is judged to appear later.

a22: The number of words appearing in the last sentence of paragraph A and in the first sentence of paragraph B

This feature uses the number of words appearing in the last sentence of paragraph A and in the first sentence of paragraph B. When paragraph B follows paragraph A, the last sentence of paragraph A and the first sentence of paragraph B will be similar and will have many nouns in common. Feature a22 can verify this.

a24: The number of words appearing in paragraph A and in paragraph B where the number is weighted using appearing places.

This feature uses the number of words appearing in paragraph A and in paragraph B and uses the places where the words appear. We make weights so as that the weight is 1 in the boundary between paragraphs A and B, the weight is smaller when the place is farther from the boundary, and the weight is 0 in the first part of paragraph A and in the last part of paragraph B. When a word appearing in paragraph A and in paragraph B, we multiply the product of the weight based on places of paragraph A where the word appears and the weight based on places of paragraph B where the word appears to the frequency (the number of words). When paragraph B follows paragraph A, the last parts of paragraph A and the first parts of paragraph B will be similar and will have many nouns in common. Feature a24 can verify this.

4 Baseline method

Information in two adjacent paragraphs will possibly be very similar. Therefore, we utilize the baseline method as follows. The two paragraphs for estimation are denoted as A and B. We count the number of words that appear in the paragraphs immediately preceding paragraphs A and B that also appear in paragraph A (or B). When the number of repeated words in paragraph A is higher than that in paragraph B, "A$\rightarrow$B" is judged by the baseline method to be the correct order.

In this study, we compare the performance of the baseline method with the performance of our proposed method.

5 Experiment

5.1 Experimental conditions

We utilized Mainichi newspaper articles (July 1992) as training data.

We utilized the following three cases for pairs of paragraphs. Case 1: The first two paragraphs in an article. Case 2: Pairs of all adjacent paragraphs in an article. Case 3: Pairs of all paragraphs. The baseline method cannot be utilized for Case 1, because the paragraphs preceding the estimated paragraphs are required.

For Case 1, features a12-a21 and a26-a29 were not utilized because they require the preceding paragraphs. For Cases 2 and 3, estimating the order by utilizing conjunctions or adnominals is difficult. Thus, in Cases 2 and 3, we did not utilize a3.

For training data, 3,124 paragraph pairs were utilized for Case 1, 15,020 for Case 2, and 82,808 for Case 3.

We used accuracy rates for evaluation. An accuracy rate is the result dividing the number of correctly estimated pairs by the number of input pairs.

5.2 Comparison between the proposed method and the baseline method

We utilized Mainichi newspaper articles (August 1, 1992) for test data. We utilized 412 paragraph pairs for Case 1, 1,620 for Case 2, and 6,624 for Case 3. Table 3 shows the accuracy rates of the proposed method and the baseline method.

In Case 1, our proposed method obtained high accuracy (0.84). In Cases 2 and 3, the accuracies of our proposed method (0.62 and 0.64) were not as high as that of Case 1; however, they were higher than those of the baseline method (0.53 and 0.58).

The baseline method uses the similarity between paragraphs. Our method use many kinds of information on the basis of features used in machine learning. Because the accuracies of our method were higher than those of the baseline method, we found that the use of many kinds of information was better than using the similarity between paragraphs only.

Table 3. Accuracy rates of the proposed method and the baseline method

	Our method	Baseline method
Case 1	0.84	-
Case 2	0.62	0.53
Case 3	0.64	0.58

5.3 Comparison with manual estimation

We compared the proposed method and manual estimation. Manual estimation was separately performed by two individuals (subjects), A and B.

We randomly selected 50 paragraph pairs from Mainichi newspaper articles as test data for Cases 1 to 3. The pairs for Case 1 were from July 1993, Case 2 were from August 1993, and Case 3 were from August 1993.

We show the accuracy of our proposed method and manual estimation in Table 4. “Average” shows the average accuracy of manual estimations.

In Table 4, the performance of the proposed method (0.82) was slightly lower than that of the manual estimation (0.88) in Case 1. In Case 2, the performance of the proposed method (0.66) was the same as that of the manual estimation (0.66). Because the performance of the manual estimation was also relatively low (0.66) in Case 2, it is believed that estimating order in this case was particularly difficult. In Case 3, the performance of the proposed method (0.72) was slightly lower than that of the manual estimation (0.77).

Table 4. Accuracy rates of the proposed method and manual estimation

	Our methhod	Subject (Manual)		
		A	B	Average
Case 1	0.82	0.92	0.84	0.88
Case 2	0.66	0.68	0.64	0.66
Case 3	0.72	0.84	0.70	0.77

5.4 Feature analysis

We utilized the following method for feature analysis. We constructed a data item with only one feature and classified it by SVM. The feature with a larger distant against a separating hyperplane is likely to be more important.

We found that in Case 1, adnominals, conjunctions, and dates (features a3 and a4) were effective for order estimation of the first two paragraphs. In Cases 2 and 3, we found that the similarity between the preceding paragraphs and an estimated paragraph (features a12 and a14), the number of new words and new word ratios (features a20 and a21), and the similarity between the preceding paragraphs and an estimated paragraph calculated by using appearing places (features a29) for the order estimation of all pairs of adjacent paragraphs and the order estimation of all pairs of paragraphs.

6 Conclusion

In this study, we proposed a method to estimate the order of paragraphs by employing supervised machine learning. In the experiments on the paragraph order estimation of the first two paragraphs of an article, our proposed method obtained a high accuracy rate of 0.84. In addition, in the order estimation of all pairs of adjacent paragraphs and all pairs of paragraphs in articles, the proposed method obtained the accuracy rates of 0.62 and 0.64. These accuracy rates were higher than those of the baseline method. From feature analysis, we found that adnominals, conjunctions, and dates were effective for the order estimation of the first two paragraphs, and the ratio of new words and the similarity between the preceding paragraphs and an estimated paragraph were effective for the order estimation of all pairs of paragraphs.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 26330252.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. Proceedings of ICDEf95, pages 3–14, 1995.

[2] Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. A bottom-up approach to sentence ordering for multi-document summarization. Proceedings of the 44th Annual Meeting of the Association of Computational Linguistics, pages 385–392, 2006.

[3] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 335–336, 1998.

[4] Nello Cristianini and John Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.

[5] Fosca Giannotti, Mirco Nanni, and Dino Pedreschi. Efficient mining of temporally annotated sequences. Proceedings of the 2006 SIAM International Conference on Data Mining, pages 348–359, 2006.

[6] Yuya Hayashi, Masaki Murata, Liangliang Fan, and Masato Tokuhisa. Japanese sentence order estimation using supervised machine learning with rich linguistic clues. In Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2013), pages 1–12, 2013.

[7] Nikiforos Karamanis and Hisar Maruli Manurung. Stochastic text structuring using the principle of continuity. In Proceedings of the second International Natural Language Generation Conference (INLGf02), pages 81–88, 2002.

[8] Taku Kudoh. TinySVM: Support Vector Machines. http://cl.aist-nara.ac.jp/ taku-ku// software/TinySVM/ index.html, 2000.

[9] Mirella Lapata. Probablistic text structuring: Experiments with sentence ordering. Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics, pages 542–552, 2003.

[10] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. Text, 8(3):243–281, 1988.

[11] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, and

Masayuki Asahara. Japanese morphological analysis system ChaSen version 2.0 manual 2nd edition. 1999.

[12] Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In Proceedings of AAAI/IAAI, pages 453–460, 1999.

[13] Masaki Murata and Hitoshi Isahara. Automatic detection of mis-spelled Japanese expressions using a new method for automatic extraction of negative examples based on positive examples. IEICE Transactions on Information and Systems, E85–D(9):1416–1424, 2002.

[14] Masaki Murata, Satoshi Ito, Masato Tokuhisa, and Qing Ma. Order estimation of Japanese paragraphs by supervised machine learning. Proceedings of SCIS-ISIS 2014, pages 1096–1101, 2014.

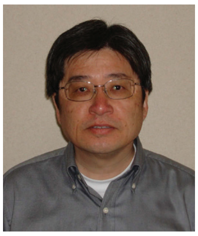
[15] Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. Improving chronological sentence ordering by precedence relation. In Proceedings of the 20th International Conference on Computational Linguistics (COLING 04), pages 750–756, 2004.

[16] Kiyotaka Uchimoto, Masaki Murata, Qing Ma, Satoshi Sekine, and Hitoshi Isahara. Word order acquisition from corpora. In COLING '2000, pages 871–877, 2000.

Masaki Murata received his Bachelor's, Master's, and Doctorate degrees in engineering from Kyoto University in 1993, 1995, and 1997, respectively. He worked in the Communications Research Laboratory (Now: National Institute of Information and Communications Technology (NICT)), Japan, from 1998 to 2010. In 2010, he moved to Tottori University, Japan, as a Professor in the Department of Information and Electronics, Graduate School of Engineering. His research interests include natural language processing, machine learning, machine translation, information retrieval, and sentence generation.

Satoshi Ito received his Bachelor's, and Master's degrees in engineering from Tottori University, Japan, in 2013 and 2015, respectively. He is currently working in Meitec, Japan.

Masato Tokuhisa received his Master's degree in computer science from Kyushu Institute of Technology, Japan, in 1995. He received his Doctor's degree in engineering from Tottori University, Japan, in 2008. He was a research associate in Kyushu Institute of Technology from 1995 to 2002 and in Tottori University from 2002 to 2010. He is currently a junior associate professor in Tottori University.



Qing Ma received his B.S. degree in electrical engineering from Beijing University of Aeronautics and Astronautics (BUAA), China, in 1983 and his M.S. and Dr. Eng. degrees in computer science from University of Tsukuba, Japan, in 1987 and 1990, respectively. He worked in the Communications Research Laboratory (Now: National Institute of Information and Communications Technology (NICT)), Japan, from 1993 to 2003. In 2003, he moved to Ryukoku University, Japan, as a professor in the Department of Applied Mathematics and Informatics. His research interests include machine learning and natural language processing.