Bottom-up Japanese Word Ordering Using BERT

Masato Yamazoe[1], Tomohiro Ohno[1,2][a] and Shigeki Matsubara[3][b]

[1]*Graduate School of Science and Technology for Future Life, Tokyo Denki University, Tokyo, Japan*

[2]*Graduate School of Advanced Science and Technology, Tokyo Denki University, Tokyo, Japan*

[3]*Information & Communications, Nagoya University, Nagoya, Japan*

Keywords: Sentence Generation, Paraphrasing, Readability, Dependency Structure.

Abstract: Although Japanese is widely regarded as a relatively free word order language, word order in Japanese is not entirely arbitrary and has some sort of preference. As a result, it is an important technique to produce a sentence that is not only grammatically correct but also easy to read. This paper proposes a method for the word ordering of a whole Japanese sentence when the dependency relations between words are given. Using BERT, the method identifies the easy-to-read word order for a syntactic tree based on bottom-up processing. We confirmed the effectiveness of our method through an experiment on word ordering using newspaper articles.

1 INTRODUCTION

Although Japanese has a relatively free word order, the word order is not completely arbitrary and has some sort of preference for readability (Nihongo Kijutsu Bunpo Kenkyukai, 2009). If a sentence is generated without taking such preferences into account, the generated sentence's word order becomes grammatically correct but difficult to read in Japanese. Unless the generator improves its word order, such a sentence is frequently generated, even if the generator is a native Japanese speaker. As a result, it is an important technique to generate a sentence whose word order is not only grammatically correct but also easy to read.

Several studies on word ordering have been performed for sentence elaboration support and sentence generation. Uchimoto et al. (Uchimoto et al., 2000) proposed a method for statistically ordering words based on different factors involved in determining word order in Japanese to learn Japanese word order trends from a corpus. Takasu et al. (Takasu et al., 2020) proposed a method of ordering for a set of *bunsetsus*[1] which are directly dependent on the same bunsetsu, for the purpose of applying it to sentence generation. Both studies by Takasu et al. and Uchimoto et al. have the same problem setting about the input and output. Because the bunsetsus input set is a subset of all bunsetsus in a sentence, neither study performs word ordering for the entire sentence. Kuribayashi et al. (Kuribayashi et al., 2020) proposed to use Japanese Language Models as a tool for examining canonical word order in Japanese, and verified the validity of using LMs for the analysis. However, they limited the number of verbs and bunsetsus in a sentence for simplicity, and thus did not target complex sentences. Although there have been also several studies on estimating appropriate word order in monolingual languages other than Japanese (Filippova and Strube, 2007; Harbusch et al., 2006; Kruijff et al., 2001; Ringger et al., 2004; Shaw and Hatzivassiloglou, 1999; Schmaltz et al., 2016), it is not clear that their studies can apply to Japanese.

As an elemental technique for sentence generation, we propose a method for ordering all bunsetsus constituting a whole sentence so that the word order becomes easy to read. Specifically, our method identifies the appropriate order by conducting bottom-up processing for a tree representing the dependency structure of a whole sentence (hereafter, dependency tree) and by using a model of BERT (Devlin et al.,

[a] https://orcid.org/0000-0001-7015-7714

[b] https://orcid.org/0000-0003-0416-3635

[1] *Bunsetsu* is a linguistic unit in Japanese that roughly corresponds to a basic phrase in English. A bunsetsu is made up of one independent word and zero or more ancillary words. In Japanese, a dependency relation is a modification relation in which a modifier bunsetsu is dependent on a modified bunsetsu. That is, the modifier bunsetsu and the modified bunsetsu work as modifier and modifiee, respectively.

Yamazoe, M., Ohno, T. and Matsubara, S.
Bottom-up Japanese Word Ordering Using BERT.
DOI: 10.5220/0011743600003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 673-681
ISBN: 978-989-758-623-1; ISSN: 2184-433X

2019), given a set of all bunsetsus in the sentence and the dependency relations between their bunsetsus as the input. We performed word ordering experiments using newspaper article sentences, and our results outperformed previous methods.

The contributions of this paper are as follows: 1. We demonstrated the utility of BERT in determining the appropriate order of two bunsetsus in Japanese. 2. We demonstrated the effectiveness of bottom-up processing in word ordering a complete Japanese sentence.

2 WORD ORDERING IN SENTENCE GENERATION

In this study, we assume that a set of bunsetsus constituting a complete sentence, as well as the dependency relations between those bunsetsus, is provided as input, and we attempt to arrange all bunsetsus in the input set in an order that is easy to read. With the applications of sentence generation and machine translation in mind, the assumption that these inputs are provided is made. When creating a sentence, it is assumed that the content to be expressed is fixed.

This assumption is also discovered in the problem setting of the previous studies by Uchimoto et al. (Uchimoto et al., 2000) and Takasu et al (Takasu et al., 2020). In the following, we discuss the above two previous studies.

2.1 Word Ordering by Using Syntactic Information (Uchimoto et al., 2000)

The word order in a sentence is related to the sentence's dependencies. Uchimoto et al. (Uchimoto et al., 2000) performed word ordering using syntactic information under the assumption that dependency parsing had already been performed. They defined word ordering task as identifying the order of bunsetsus in a set of all modifier bunsetsus $B_r = \{b_{r_1}, b_{r_2}, \cdots, b_{r_n}\} (n \geq 2)$ which depend on the same modifiee bunsetsu b_r. They made all possible permutations $\{\mathbf{b}^k | 1 \leq k \leq n!\}$ from B_r and found the easiest-to-read one among $\{\mathbf{b}^k | 1 \leq k \leq n!\}$. Here $\mathbf{b}^k$ is the k-th permutation. They proposed a method of word ordering based on syntactic information using a model trained by the Maximum Entropy Method. However, they focused on the word ordering of B_r, which is a subset of all bunsetsus in a sentence, so they did not identify the order of bunsetsus in a whole sentence and did not evaluate their results in a sentence unit.

2.2 Word Ordering by Using RNNLM and SVM (Takasu et al., 2020)

Takasu et al. (Takasu et al., 2020) inherited the problem setting of Utimoto et al. (Uchimoto et al., 2000), and used not only an SVM model trained based on the main features[2] selected from those of Uchimoto et al., but also the Recurrent Neural Network Language Model(RNNLM), which is expected to capture natural word ordering. In particular, among $\{\mathbf{b}^k | 1 \leq k \leq n!\}$ which means all permutations created by B_r, they identified the permutation $\mathbf{b}^k$ which maximize $Score(\mathbf{b}^k)$ defined by Formula (1) as the most readable one.

$$Score(\mathbf{b}^k) = \alpha S_{RNNLM}(\mathbf{b}^k) + (1 - \alpha) S_{SVM}(\mathbf{b}^k), \quad (1)$$

where $S_{RNNLM}(\mathbf{b}^k)$ and $S_{SVM}(\mathbf{b}^k)$ mean the score for a $\mathbf{b}^k$ by the RNNLM and by the SVM model, respectively.

$S_{SVM}(\mathbf{b}^k)$ is defined by Formula (2). Here, a permutation $\mathbf{b}^k$ is expressed by $\mathbf{b}^k = b_1^k b_2^k \cdots b_n^k$, where b_i^k is the i-th bunsetsu in the k-th permutation $\mathbf{b}^k$.

$$S_{SVM}(\mathbf{b}^k) = \prod_{i=1}^{n-1} \prod_{j=1}^{n-i} P_{SVM}(o_{i,i+j}^k | b_r) \qquad (2)$$

Here, $o_{i,i+j}^k$ means an order relation that the bunsetsu b_i^k is located before b_{i+j}^k, and $P_{SVM}(o_{i,i+j}^k | b_r)$ means the probability predicted[3] by the SVM model, which expresses how appropriate the order relation $o_{i,i+j}^k$ is in readability given the bunsetsu b_r which is depended by the two bunsetsu b_i^k and b_{i+j}^k. Takasu et al.'s SVM model is thought to be a re-implementation of the Maximum Entropy Method used in Utimoto et al. (Uchimoto et al., 2000) to capture the word order tendency based on syntactic information.

$S_{RNNLM}(\mathbf{b}^k)$ is defined by Formula (3). Here, a permutation $\mathbf{b}^k$ is expressed by $\mathbf{b}^k = w_{1,1}^k, w_{1,2}^k \cdots w_{1,m_1}^k \cdots w_{n,1}^k \cdots w_{n,m_n}^k$, where $w_{i,j}^k$ is the j-th word[4] in the i-th bunsetsu in the k-th permutation $\mathbf{b}^k$.

$$S_{RNNLM}(\mathbf{b}^k) = \prod_{i=1}^{n} \prod_{j=1}^{m_i} P_{RNNLM}(w_{i,j}^k | w_{1,1}^k, w_{1,2}^k, \cdots w_{i,j-1}^k), \quad (3)$$

where $P_{RNNLM}(w_{i,j}^k | w_{1,1}^k, w_{1,2}^k, \cdots w_{i,j-1}^k)$ means the probability that $w_{i,j}^k$ comes just after the word sequence $w_{1,1}^k, w_{1,2}^k, \cdots w_{i,j-1}^k$, which is predicted by

[2] As the main features, morpheme information of each bunsetsu connected by dependency relations is used.

[3] The probability is calculated by Platt's scaling (Platt, 2000).

[4] To be more accurate, a symbol showing a boundary between bunsetsus is placed after the final word of each bunsetsu.

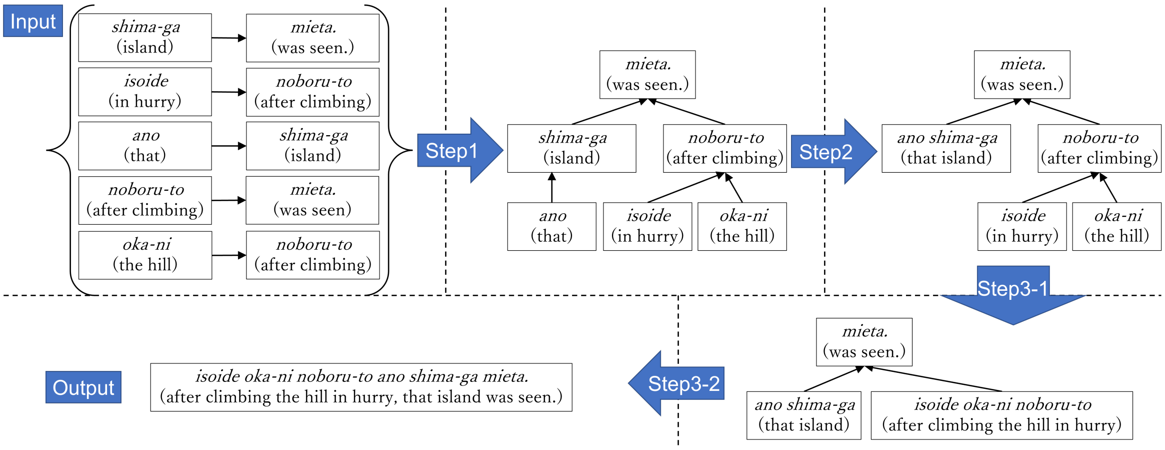

Input
shima-ga (island)
mieta. (was seen.)
isoide (in hurry)
noboru-to (after climbing)
ano (that)
shima-ga (island)
noboru-to (after climbing)
mieta. (was seen)
oka-ni (the hill)
noboru-to (after climbing)
Step1
mieta. (was seen.)
shima-ga (island)
noboru-to (after climbing)
ano (that)
isoide (in hurry)
oka-ni (the hill)
Step2
mieta. (was seen.)
ano shima-ga (that island)
noboru-to (after climbing)
isoide (in hurry)
oka-ni (the hill)
Step3-1
mieta. (was seen.)
ano shima-ga (that island)
isoide oka-ni noboru-to (after climbing the hill in hurry)
Step3-2
Output
isoide oka-ni noboru-to ano shima-ga mieta. (after climbing the hill in hurry, that island was seen.)

Figure 1: Example of word ordering by the bottom-up processing.

RNN having two hidden layers of LSTM (Sundermeyer et al., 2012). Since the number of factors in Formula (2) is different from that of Formula (3), $Score(\mathbf{b}^k)$ in Formula (1) is calculated after the normalization of $S_{RNNLM}(\mathbf{b}^k)$ and $S_{SVM}(\mathbf{b}^k)$.

However, like Uchimoto et al., they focused on the word ordering of B_r, which is a subset of all bunsetsus in a sentence, and thus did not identify the order of bunsetsus in a whole sentence or evaluate their results in a sentence unit. Furthermore, when RNNLM calculated the score, they did not take into account the modifier bunsetsus, which depend on each bunsetsu in a set B_r. That is, because they only targeted child nodes in the dependency tree and ignored descendant nodes, the string generated by ordering the target bunsetsus did not always appear in actual texts.

3 BOTTOM-UP WORD ORDERING USING BERT

All bunsetsus that compose a complete sentence, as well as their dependency relations, are considered input in our method. Our method outputs a sentence in which the input bunsetsus are arranged so that the word order is easy to read. At that point, our method determines the appropriate word order among the input bunsetsus using bottom-up processing for a dependency tree constructed from the input bunsetsus and their dependency relations, as well as a BERT model to judge the word order between two bunsetsus.

3.1 Bottom-Up Processing

Our approach conducts word ordering by the following bottom-up processing.

1. A dependency tree is created from all input bunsetsus and dependency relations. Each bunsetsu is placed as a node, and the nodes are connected via edges that express dependency relations. Here, in the following steps 2 and 3, multiple nodes are merged into one node. As a result, strictly speaking, a node is a sequence of bunsetsus, including a sequence of length one, and an edge is a dependency relationship in which the final bunsetsu of a bunsetsu sequence constituting a child node depends on the final bunsetsu of a bunsetsu sequence constituting a parent node.
2. A parent node which has only one leaf child node and the child node are merged into one node. The merge process is performed by concatenating the child node's bunsetsu sequence before the parent node's bunsetsu sequence, while keeping in mind the Japanese syntactic constraints that no dependency relations are directed from right to left. This step is repeated until no parent node has only one leaf child node.
3. For a parent node which has multiple leaf child nodes, a parent node and the child nodes are merged into one node. First, a BERT model is used to determine the correct order of the child nodes in the merge process. Second, the calculated order is used to concatenate the bunsetsu sequences of each child node. Finally, the bunsetsu sequence of the parent node is concatenated after the above-created bunsetsu sequence.
4. Step 2 and step 3 are repeated until the dependency tree has only one root node.

Figure 1 shows an example of the bottom-up processing above. In step 1, the six input bunsetsus and the five dependency relations between the six input bunsetsus are used to create the dependency tree in the

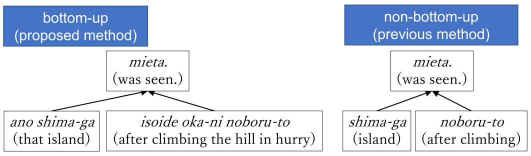


Figure 2: Differences between the bottom-up processing and non-bottom-up processing.

top center of Figure 1. In step 2, “*shima-ga* (island),” which is a parent node having only one leaf child node, and the child node “*ano* (that)” are merged into one node “*ano shima-ga* (that island).” In step 3-1, the appropriate word order among the leaf child nodes “*isoide* (in hurry)” and “*oka-ni* (the hill),” which depend on their parent node “*noboru-to* (after climbing),” are calculated and the child nodes and parent node are merged into one child node “*isoide oka-ni noboru-to* (after climbing the hill in hurry)” according to the calculated order. In step 3-2, the correct word order among the leaf child nodes “*ano shima-ga* (that island)” and “*isoide oka-ni noboru-to* (after climbing the hill in hurry),” which depend on “*mieta.* (was seen.),” are calculated and the three bunsetsus are merged into one node “*isoide oka-ni noboru-to ano shima-ga mieta.* (after climbing the hill in hurry, that island was seen.).” Since there is now only one root node in the tree, the word ordering is complete.

Based on Figure 2, we outline the differences between the bottom-up processing employed by our method and the non-bottom-up one employed by Takasu et al.’s method (Takasu et al., 2020). In the non-bottom-up processing, when calculating the appropriate word order among the leaf child nodes of the root node “*mieta.* (was seen.),” the child nodes “*shima-ga* (island)” and “*noboru-to* (after climbing,)” remain composed of one bunsetsu because those nodes are not created by their descendant nodes being merged into. As a result, the non-bottom-up processing judges which of two bunsetsu sequences ”*shima-ga noboru-to* (island after climbing)” and ”*noboru-to shima-ga* (after climbing, island),” which are respectively composed of two bunsetsus, is easier to read by using a machine learning model.

However, in the bottom-up processing, when determining the correct word order among the leaf child nodes of the root node, the child nodes “*ano shima-ga* (that island)” and “*isoide oka-ni noboru-to* (after climbing the hill in hurry)” have been created by their descendant nodes being merged into during the bottom-up processing. Therefore, ”*ano shima-ga isoide oka-ni noboru-to* (that island after climbing the hill in hurry)” and ”*isoide oka-ni noboru-to ano shima-ga* (after climbing the hill in hurry, that island)” are the focus of the judgment by a machine learning model.

3.2 Identification of Child Node Order Using BERT

As explained in Section 3.1, a model based on BERT (Devlin et al., 2019) is used when determining the appropriate order among leaf child nodes of a parent node. Hereinafter, we explain the calculation in case of ordering the leaf child nodes $V_r = \{v_{r_1}, v_{r_2}, \cdots, v_{r_n}\}$ of a parent node v_r. When the leaf child nodes $V_r = \{v_{r_1}, v_{r_2}, \cdots, v_{r_n}\}$ is provided, among $\{\mathbf{v}^k | 1 \leq k \leq n!\}$ which means all permutations[5] created by V_r, our method identifies the permutation $\mathbf{v}^k$ which maximize $S(\mathbf{v}^k)$ defined by Formula (4) as the most readable one. Concretely speaking, when a permutation $\mathbf{v^k}$ is a sequence of nodes $v^k_1 v^k_2 \cdots v^k_n$, where v^k_i means the i-th node in a permutation $\mathbf{v}^k$, $S(\mathbf{v^k})$ is calculated as follows.

$$S(\mathbf{v^k}) = \prod_{i=1}^{n-1} \prod_{j=1}^{n-i} P_{BERT}(o^k_{i,i+j} | v_r) \tag{4}$$

Here, $o^k_{i,i+j}$ means an order relation that the node v^k_i is located before v^k_{i+j}, instead of an order relation between two bunsetus in Section 2.2. $P_{BERT}(o^k_{i,i+j} | v_r)$ means the probability predicted by our BERT model, which expresses how appropriate the order relation $o^k_{i,i+j}$ is in readability when the two nodes v^k_i and v^k_{i+j} depend on a parent node v_r.

Figure 3 shows the outline of our BERT model. The input to BERT is the subword sequence of the concatenation of the two leaf child nodes v^k_i and v^k_{i+j} and the parent node v_r in this order with [CLS] at the beginning and [SEP] after each of the three nodes. Our approach takes only the output of BERT corresponding to [CLS], which is the representation of the totality of the input token sequence, and via the two Linear layers and Sigmoid, outputs the probability that the order relation that v^k_i is located before v^k_{i+j} is easy to read. Each sequence length of the leaf child nodes is altered when the input sequence length surpasses the BERT upper limit. To adjust, some subwords are removed from the front of each node while ensuring that the ratio of the two nodes’ sequence length is maintained, while each sequence length is not shorter than the length of the longest bunsetsu measured in training data.

[5] Since the number of bunsetsus which depend on a modifiee is at most 7, the number of all possible permutations is within computable range.

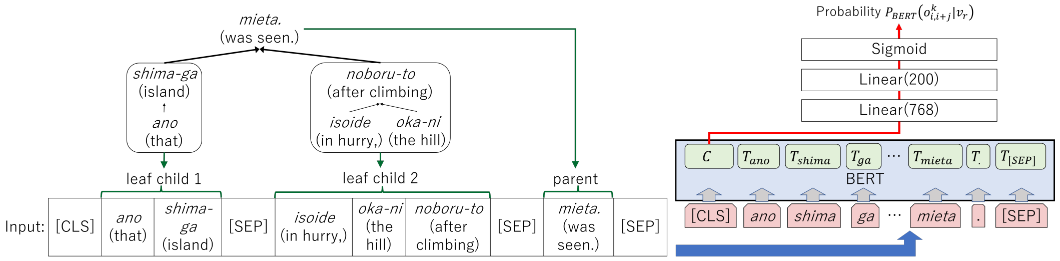


Figure 3: Our BERT model to predict $P_{BERT}(o^k_{i,i+j}|v_r)$ based on BERT.

4 EXPERIMENT

We conducted an experiment using newspaper articles to gauge the effectiveness of our word ordering strategy. The word order in newspaper articles is considered to be readable in this paper.

4.1 Outline of Experiment

In our experiment, we utilized the Kyoto University text corpus Ver. 4.0 (Kawahara et al., 2002), which is annotated with information on morphological analysis, bunsetsu segmentation, and dependency analysis. We used 25,388 sentences from newspaper articles published in 1995 between January 1st and January 8th and from January 10th to June 9th for training purposes[6]. Among the 2,368 sentences from articles appearing on January 9th and from June 10th to June 30th, 1,050 sentences were used for development data, and 1,164 sentences for test data. Because the order of a complete sentence could be determined solely by syntactic information by simply repeating step 2 in Section 3.1, the remaining 154 sentences were not used in this instance.

We obtained the following two measurements to assess word ordering. The first is **sentence agreement**: the percentage of the sentences whose whole word order agrees with that in the original sentence. The second is **pair agreement**: the percentage of the pairs of bunsetsus whose word order agrees with that in the original sentence[7]. For each agreement, we calculated the mean of the five models made with the same hyperparameters.

We prepared the following four methods for comparison.

- **[BERT$^-$]:** is as same as our method except using non-bottom-up processing instead of bottom-up processing.
- **[RNNLM$^-$+SVM$^-$]:** is the method proposed by Takasu et al (Takasu et al., 2020), which employs the non-bottom-up processing. This is built on setting $\alpha = 0.15$ of Formula (1).
- **[RNNLM+SVM]:** is as same as the above method [RNNLM$^-$+SVM$^-$] except utilizing the bottom-up processing instead of the non-bottom-up processing. This is built on setting $\alpha = 0.19$ of Formula (1).
- **[SVM$^-$]:** is the method which singly uses SVM in [RNNLM$^-$+SVM$^-$], that is, which is built on setting α =0 in Formula (1). We assume that this method is equivalent to the Uchimoto et al. method that has been re-implemented (Uchimoto et al., 2000).

We implemented our model in Figure 3 using PyTorch[8]. For the Japanese pre-trained model of BERT, we used the model published by Kyoto University (BASE WWM ver.)[9]. The two Linear layers' respective unit counts were set at 768 and 200, and their respective drop-out rates were set at 0.1. An optimizer named AdamW was employed. Parameters were updated using mini-batch learning (learning rate 1e-6, batch size 16). BCELoss was used as the loss function. The sentence agreement for each epoch was measured using the development data, and thus, epoch 6 was found to be the best epoch.

4.2 Experimental Results

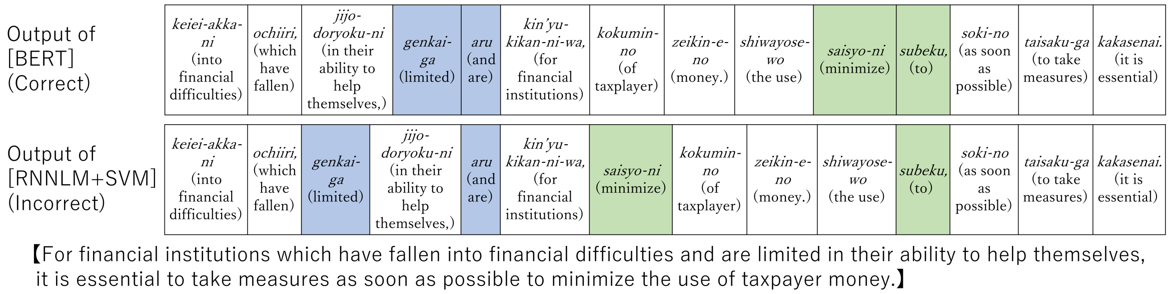
Table 1 shows the experimental results. Our method [BERT] significantly outperformed all other methods

[6] In creating training data, we extracted all pairs of two nodes from the set of multiple nodes which depends on the same bunsetsu, and then, we made two order relations for each pair and labeled an order relation same as the correct one as a positive and the other incorrect one as a negative.

[7] To avoid counting redundant pairs, the pairs whose order relation is fixed by that of each bunsetsu's ancestor are not measured.

[8] https://pytorch.org/

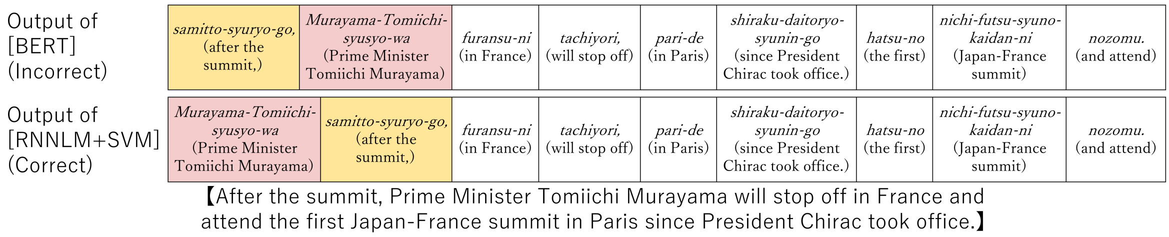
[9] https://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese/



Output of [BERT] (Correct)	*keiei-akka-ni* (into financial difficulties)	*ochiiri,* (which have fallen)	*jijo-doryoku-ni* (in their ability to help themselves,)	*genkai-ga* (limited)	*aru* (and are)	*kin'yu-kikan-ni-wa,* (for financial institutions)	*kokumin-no* (of taxplayer)	*zeikin-e-no* (money.)	*shiwayose-wo* (the use)	*saisyo-ni* (minimize)	*subeku,* (to)	*soki-no* (as soon as possible)	*taisaku-ga* (to take measures)	*kakasenai.* (it is essential)
Output of [RNNLM+SVM] (Incorrect)	*keiei-akka-ni* (into financial difficulties)	*ochiiri,* (which have fallen)	*genkai-ga* (limited)	*jijo-doryoku-ni* (in their ability to help themselves,)	*aru* (and are)	*kin'yu-kikan-ni-wa,* (for financial institutions)	*saisyo-ni* (minimize)	*kokumin-no* (of taxplayer)	*zeikin-e-no* (money.)	*shiwayose-wo* (the use)	*subeku,* (to)	*soki-no* (as soon as possible)	*taisaku-ga* (to take measures)	*kakasenai.* (it is essential)

【For financial institutions which have fallen into financial difficulties and are limited in their ability to help themselves, it is essential to take measures as soon as possible to minimize the use of taxpayer money.】

Figure 4: Example of sentences correctly ordered by [BERT] but incorrectly by [RNNLM+SVM].



Output of [BERT] (Incorrect)	*samitto-syuryo-go,* (after the summit,)	*Murayama-Tomiichi-syusyo-wa* (Prime Minister Tomiichi Murayama)	*furansu-ni* (in France)	*tachiyori,* (will stop off)	*pari-de* (in Paris)	*shiraku-daitoryo-syunin-go* (since President Chirac took office.)	*hatsu-no* (the first)	*nichi-futsu-syuno-kaidan-ni* (Japan-France summit)	*nozomu.* (and attend)
Output of [RNNLM+SVM] (Correct)	*Murayama-Tomiichi-syusyo-wa* (Prime Minister Tomiichi Murayama)	*samitto-syuryo-go,* (after the summit,)	*furansu-ni* (in France)	*tachiyori,* (will stop off)	*pari-de* (in Paris)	*shiraku-daitoryo-syunin-go* (since President Chirac took office.)	*hatsu-no* (the first)	*nichi-futsu-syuno-kaidan-ni* (Japan-France summit)	*nozomu.* (and attend)

【After the summit, Prime Minister Tomiichi Murayama will stop off in France and attend the first Japan-France summit in Paris since President Chirac took office.】

Figure 5: Example of sentences incorrectly ordered by [BERT] but correctly by [RNNLM+SVM].

Table 1: Experimental results (pair and sentence agreement).

	pair	sentence
[BERT]	92.01%	71.53%
[BERT$^-$]	89.48%	65.36%
[RNNLM+SVM]	85.56%	58.68%
[RNNLM$^-$+SVM$^-$]	85.84%	58.59%
[SVM$^-$]	85.49%	57.47%

in terms of both pair agreement and sentence agreement ($p < 0.01$). As a result, we confirmed the effectiveness of bottom-up word ordering using BERT in word ordering.

5 DISCUSSION

5.1 Effects of Using BERT

We compare our method [BERT] with [RNNLM+SVM], which employs the same bottom-up processing as [BERT] but uses the different machine learning RNNLM and SVM from BERT, to analyze the outcomes of using BERT.

First, we investigate the positive effect of using BERT. There were 231 sentences in which all bunsetsus were correctly ordered by our method [BERT][10], but incorrectly ordered by [RNNLM+SVM]. A typical example is depicted in Figure 4, where a box expresses a bunsetsu. In this example, our method successfully concatenated two bunsetsus which tend to be continuous and expressed as a single phrase like an idiom, such as "*genkai-ga aru* (and are limited)" and "*saisyo-ni subeku* (to minimize)", without interrupting other bunsetsus between the two ones.

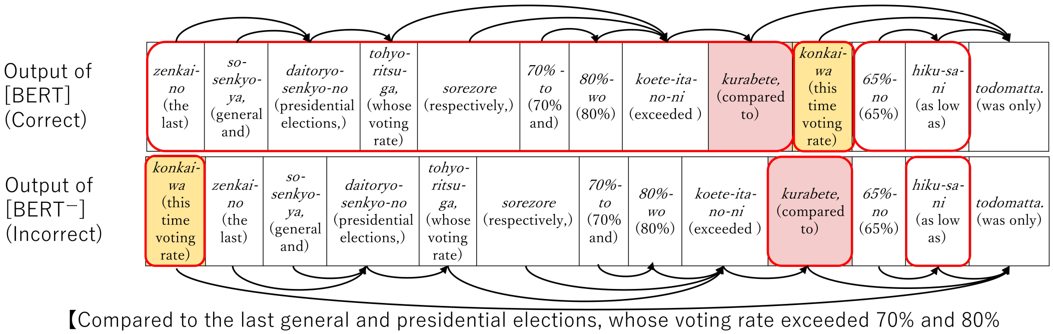
Next, we analyzed the negative effect of using BERT. There were 78 sentences in which all of the bunsetsus were correctly arranged by [RNNLM+SVM] but were arranged improperly by our method [BERT]. A typical example is shown in Figure 5, where a box expresses a bunsetsu. In this example, [BERT] made a mistake by placing "*samitto-syuryo-go* (After the summit)" before "*Murayama-Tomiichi-syusyo-wa* (Prime Minister Tomiichi Murayama)." However, the [BERT] output sentence is thought to be just as readable as the gold sentence, which is a sentence from a newspaper article. Although many of the sentences produced by [BERT] were found to be incorrect, they were considered to be as readable as the gold sentence. Note that this analysis was performed without context. Future research will need to examine the relationships between word order and context.

5.2 Effects of Bottom-up Processing

By contrasting [BERT] and [BERT$^-$], which employs the same machine learning BERT as [BERT] but non-bottom-up processing, we can describe the outcomes of using the bottom-up processing in our method [BERT].

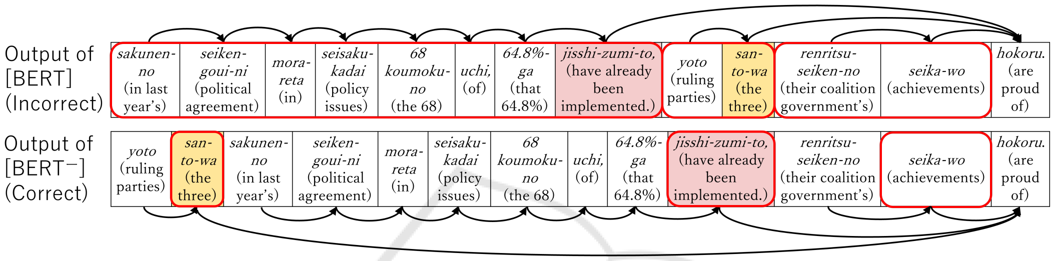
First, we examine the positive effect of employ-

[10] We analyzed the experimental results using the model with the highest sentence agreement among the five models created as our method.


Output of [BERT] (Correct)
zenkai-no (the last)
so-senkyo-ya, (general and)
daitoryo-senkyo-no (presidential elections,)
tohyo-ritsu-ga, (whose voting rate)
sorezore (respectively,)
70% -to (70% and)
80%-wo (80%)
koete-ita-no-ni (exceeded)
kurabete, (compared to)
konkai-wa (this time voting rate)
65%-no (65%)
hiku-sa-ni (as low as)
todomatta. (was only)
Output of [BERT−] (Incorrect)
konkai-wa (this time voting rate)
zenkai-no (the last)
so-senkyo-ya, (general and)
daitoryo-senkyo-no (presidential elections,)
tohyo-ritsu-ga, (whose voting rate)
sorezore (respectively,)
70%-to (70% and)
80%-wo (80%)
koete-ita-no-ni (exceeded)
kurabete, (compared to)
65%-no (65%)
hiku-sa-ni (as low as)
todomatta. (was only)

【Compared to the last general and presidential elections, whose voting rate exceeded 70% and 80% respectively, this time voting rate was only as low as 65%.】

Figure 6: Example of sentences correctly ordered by [BERT] but incorrectly by [BERT^{-}].


Output of [BERT] (Incorrect)
sakunen-no (in last year's)
seiken-goui-ni (political agreement)
mora-reta (in)
seisaku-kadai (policy issues)
68 koumoku-no (the 68)
uchi, (of)
64.8%-ga (that 64.8%)
jisshi-zumi-to, (have already been implemented.)
yoto (ruling parties)
san-to-wa (the three)
renritsu-seiken-no (their coalition government's)
seika-wo (achievements)
hokoru. (are proud of)
Output of [BERT−] (Correct)
yoto (ruling parties)
san-to-wa (the three)
sakunen-no (in last year's)
seiken-goui-ni (political agreement)
mora-reta (in)
seisaku-kadai (policy issues)
68 koumoku-no (the 68)
uchi, (of)
64.8%-ga (that 64.8%)
jisshi-zumi-to, (have already been implemented.)
renritsu-seiken-no (their coalition government's)
seika-wo (achievements)
hokoru. (are proud of)

【The three ruling parties are proud of their coalition government's achievements that 64.8 % of the 68 policy issues in last year's political agreement have already been implemented.】

Figure 7: Example of sentences incorrectly ordered by [BERT] but correctly by [BERT^{-}].

ing bottom-up processing. There existed 134 sentences in which all bunsetsus were appropriately ordered by our method [BERT], but incorrectly ordered by [BERT^{-}]. A typical example is shown in Figure 6, where a box and an arrow express a bunsetsu and a dependency relation between bunsetsus respectively. In Japanese, it is known that a long modifier phrase has a strong preference to be located at the beginning of a sentence (Nihongo Kijutsu Bunpo Kenkyukai, 2009). In this example, according to the preference, our method arranged suitably three nodes expressed by red rounded rectangles at the top of Figure 6 by placing the longest modifier phrase "*zenkai-no* ⋯ *kurabete* (last ⋯ Compared to)" at the beginning of a sentence. It is conceivable that the BERT model in our method can take the information into account because each node targeted by our method is composed of multiple bunsetsus by being combined during bottom-up processing. In contrast, [BERT^{-}] made a mistake by placing the same longest modifier phrase in the middle of the sentence. This is because each node targeted by [BERT^{-}] only contains one bunsetsu, as indicated by the red rounded rectangle at the bottom of Figure 6. As a result, the BERT model is unable to take into account the information of the other bunsetsus that make up the modifier phrase.

Next, we analyzed the negative effect of employing bottom-up processing. There existed 59 sentences in which all bunsetsus were correctly ordered by [BERT^{-}], but incorrectly ordered by [BERT]. A typical example is depicted in Figure 7, where a box, an arrow and a red rounded rectangle express the same as those of Figure 6. In this example, our method made a mistake by placing the longest modifier phrase "*sakunen-no* ⋯ *jisshi-zumi-to* (in last year's ⋯ have already been implemented.) before "*yoto san-to-ha* (The three ruling parties)" when ordering the three child nodes of "*hokoru.* (are proud of)". Although our method could have arranged the nodes so that a long modifier phrase would preferably appear on the front side of a sentence, this was a relatively uncommon word order in the gold sentence, where the long modifier phrase would appear in the middle. However, [BERT^{-}] could arrange the three phrases in the same order as that of the gold sentence because of not being influenced by the information of the other bunsetsus composing the modifier phrase. The output sentence by [BERT] in this example is also considered to be just as readable as the gold sentence. The subjective evaluation is an issue for the future.

5.3 Analysis on Sentence Length

The increased number of bunsetsus in a sentence as well as the permutations are thought to be the main reasons why word ordering generally has lower ac-

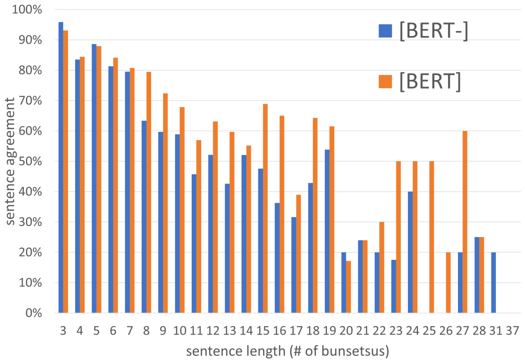


Figure 8: Sentence agreement relative to sentence length.

curacy for longer sentences. It is assumed that the bottom-up processing of our method, which takes into account all modifiers (descendant bunsetsus) of each bunsetsu, is effective for the word ordering of long sentences, in which each bunsetsu tends to have more modifiers. In this section, we explain the analysis of sentence length in terms of whether the bottom-up or non-bottom-up processing is carried out.

Figure 8 demonstrates the sentence agreement of our method [BERT] and the method [BERT$^-$], which uses non-bottom-up processing, relative to sentence length. At almost every length, [BERT] outperformed [BERT$^-$]. Particularly, the agreement of [BERT$^-$] decreases as the number of bunsetsus in a sentence rises, whereas [BERT] maintains a relatively high agreement no matter how many bunsetsus are present.

In our method, the input to BERT includes the modifiers (descendant bunsetsus) of each bunsetsu because of conducting the bottom-up processing. Therefore, it is conceivable that our method could properly consider the information included in the modifiers, and thus could perform word ordering with a high agreement for long sentences also.

6 CONCLUSION

In this paper, we propose a method for Japanese word ordering. By processing the dependency tree from the bottom up and utilizing BERT, our method can determine the appropriate order for a set of bunsetsus that make up a sentence. The experimental results verified the effectiveness of the BERT model and bottom-up word ordering. In the future, we would like to enhance the evaluation including the use of operators' subjective judgments.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Number JP19K12127.

REFERENCES

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Filippova, K. and Strube, M. (2007). Generating constituent order in German clauses. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 320–327.

Harbusch, K., Kempen, G., van Breugel, C., and Koch, U. (2006). A generation-oriented workbench for performance grammar: Capturing linear order variability in German and Dutch. In *Proceedings of the 4th International Natural Language Generation Conference*, pages 9–11.

Kawahara, D., Kurohashi, S., and Hasida, K. (2002). Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 2008–2013.

Kruijff, G.-J. M., Kruijff-Korbayovà, I., Bateman, J., and Teich, E. (2001). Linear order as higher-level decision: Information structure in strategic and tactical generation. In *Proceedings of the 8th European Workshop on Natural Language Generation*, pages 74–83.

Kuribayashi, T., Ito, T., Suzuki, J., and Inui, K. (2020). Language models as an alternative evaluator of word order hypotheses: A case study in Japanese. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 488–504.

Nihongo Kijutsu Bunpo Kenkyukai (2009). *Gendai nihongo bunpo 7(Contemporary Japanese Grammer 7)*. Kuroshio Shuppan. (In Japanese).

Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

Ringger, E., Gamon, M., Moore, R. C., Rojas, D., Smets, M., and Corston-Oliver, S. (2004). Linguistically informed statistical models of constituent structure for ordering in sentence realization. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 673–679.

Schmaltz, A., Rush, A. M., and Shieber, S. (2016). Word ordering without syntax. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324.

Shaw, J. and Hatzivassiloglou, V. (1999). Ordering among premodifiers. In *Proceedings of the 37th Annual Meet-*

ing of the Association for Computational Linguistics, pages 135–143.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). LSTM neural networks for language modeling. In *Proceedings of Interspeech 2012*, pages 194–197.

Takasu, M., Ohno, T., and Matsubara, S. (2020). Japanese word ordering using RNNLM and SVM. In *In Proceedings of the 82nd National Convention of IPSJ, Volume 2020(1)*, pages 453–454. (In Japanese).

Uchimoto, K., Murata, M., Ma, Q., Sekine, S., and Isahara, H. (2000). Word order acquisition from corpora. In *Proceedings of the 18th International Conference on Computational Linguistics, Volume 2*, pages 871–877.