

Japanese Word Reordering Executed Concurrently with Dependency Parsing and Its Evaluation

Tomohiro Ohno^{1,a)} Kazushi Yoshida²⁾ Yoshihide Kato^{3,b)} Shigeki Matsubara^{2,c)}

¹Information Technology Center, Nagoya University, Japan

²Graduate School of Information Science, Nagoya University, Japan

³Information & Communications, Nagoya University, Japan

^{a)}ohno@nagoya-u.jp ^{b)}yoshihide@icts.nagoya-u.ac.jp

^{c)}matubara@nagoya-u.jp

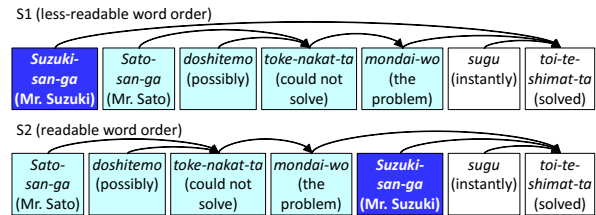
Abstract

This paper proposes a method for re-ordering words in a Japanese sentence based on concurrent execution with dependency parsing so that the sentence becomes more readable. Our contributions are summarized as follows: (1) we extend a probabilistic model used in the previous work which concurrently performs word reordering and dependency parsing; (2) we conducted an evaluation experiment using our semi-automatically constructed evaluation data so that sentences in the data are more likely to be spontaneously written by natives than the automatically constructed evaluation data in the previous work.

1 Introduction

Although Japanese has relatively free word order, Japanese word order is not completely arbitrary and has some sort of preference. Since such preference is incompletely understood, even Japanese natives often write Japanese sentences which are grammatically well-formed but not easy to read. For example, in Figure 1, the word order of S1 is less readable than that of S2 because the distance between the bunsetsu “*Suzuki-san-ga* (Mr. Suzuki)” and its modified bunsetsu “*toi-te-shimat-ta* (solved)” is large and thus the loads on working memory become large (Nihongo Kijutsu Bunpo Kenkyukai, 2009; Uchimoto et al., 2000)

There have been some conventional researches for reordering words in a sentence so that the sentence becomes easier to read (Belz et al., 2011; Filippova and Strube, 2007; Harbusch et al., 2006; Kruijff et al., 2001; Ringger et al., 2004; Shaw and Hatzivassiloglou, 1999; Uchimoto et al., 2000; Yokobayashi et al., 2004). Most of the conventional researches used syntactic information by assuming that an input sentence for word reordering



Note: A box and an arrow express a *bunsetsu*¹ and a dependency relation, respectively. Both the sentences S1 and S2 have the same meaning which is translated as “Mr. Suzuki instantly solved the problem that Mr. Sato could not possibly solve.” in English. The difference between S1 and S2 is just in their word orders in Japanese.

Figure 1: Example of less-readable/readable word order

has been already parsed. There is a problem that the errors of dependency parsing increase when an input sentence is less-readable, and the parsing errors cause negative effects on word reordering. To solve the problem, we previously proposed a method for concurrently performing word reordering and dependency parsing and confirmed the effectiveness of their proposed method using evaluation data created by randomly changing the word order in newspaper article sentences (Yoshida et al., 2014). However, since some of the just automatically created sentences are unlikely to be spontaneously written by a native, the evaluation is thought to be not enough. In addition, the probabilistic model has room for improvement in targeting at sentences which a native is likely to spontaneously write.

This paper proposes a new method on Japanese word reordering based on concurrent execution with dependency parsing by extending the probabilistic model proposed by Yoshida et al. (2014), and describes an evaluation experiment using our

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

evaluation data semi-automatically constructed by adding human judgement after automatically changing word order in newspaper article sentences. The experimental results showed the effectiveness of our method.

2 Word Order and Dependency

In this section, we discuss the relation between word order and dependency in a Japanese sentence using the example shown in Figure 1.

On the ground that dependency is one of fundamental contributing factors which decide the appropriate word order (Nihongo Kijutsu Bunpo Kenkyukai, 2009), the conventional method (Uchimoto et al., 2000) reordered words using syntactic information obtained by dependency parsing which was assumed to be beforehand performed. However, the accuracy of dependency parsing decreases when an input sentence has less-readable word order such as S1 because dependency parsers are usually trained on syntactically annotated corpora in which sentences have the readable word order such as S2.

On the other hand, if word reordering is performed before dependency parsing, the accuracy of the word reordering is thought to decrease because syntactic information can not be utilized. In fact, to change the word order in S1 to the appropriate one such as S2, it is necessary to comprehend the dependency structure of S1.

The above discussion indicates that word reordering and dependency parsing depend on each other. Therefore, we can consider it is more desirable to concurrently perform the two processes than to sequentially perform them.

3 Word Reordering Method

In our method, a sentence, on which morphological analysis and bunsetsu segmentation have been performed, is considered as the input. We assume that the input sentence might have word order which is not easy to read but grammatically well-formed. Our method identifies the suitable word order which is easy to read by being executed concurrently with dependency parsing.

We realize the concurrent execution of dependency parsing and word reordering by searching for the maximum-likelihood pattern of word order and dependency structure for an input sentence. We use the same search algorithm as one proposed by Yoshida et al. (2014), which can effi-

ciently find the approximate solution from a huge number of candidates of the pattern by extending CYK algorithm used in conventional dependency parsing. In this paper, we refine the probabilistic model proposed by Yoshida et al. (2014) to improve the accuracy. Note our method reorders bunsetsus in a sentence without paraphrasing and does not reorder morphemes within a bunsetsu. In addition, we assume there are not any inverted structures and commas in an input sentence.

3.1 Probabilistic Model for Word Reordering

When a sequence of bunsetsus in an input sentence $B = b_1 \cdot \dots \cdot b_n$ is provided, our method identifies the structure S which maximizes $P(S|B)$. The structure S is defined as a tuple $S = \langle O, D \rangle$ where $O = \{o_{1,2}, o_{1,3}, \dots, o_{1,n}, \dots, o_{i,j}, \dots, o_{n-2,n-1}, o_{n-2,n}, o_{n-1,n}\}$ is the word order pattern after reordering and $D = \{d_1, \dots, d_{n-1}\}$ is dependency structure. Here, $o_{i,j}$ ($1 \leq i < j \leq n$) expresses the order between b_i and b_j after reordering. $o_{i,j}$ is 1 if b_i is located before b_j , and is 0 otherwise. In addition, d_i expresses the dependency relation whose modifier bunsetsu is b_i .

In the probabilistic model proposed by Yoshida et al. (2014), $P(S|B)$ was calculated as follows:

$$\begin{aligned} P(S|B) &= P(O, D|B) \\ &= \sqrt{P(O|B) \times P(D|O, B)} \quad (1) \\ &\quad \times \sqrt{P(D|B) \times P(O|D, B)} \end{aligned}$$

We extend the above model and calculate $P(S|B)$ as follows:

$$\begin{aligned} P(S|B) &= \{P(O|B) \times P(D|O, B)\}^\alpha \quad (2) \\ &\quad \times \{P(D|B) \times P(O|D, B)\}^{1-\alpha} \end{aligned}$$

where α is a weight and $0 \leq \alpha \leq 1$. Formula (2) is obtained for the weighted geometric average² between the following two Formulas (3) and (4).

$$P(O, D|B) = P(O|B) \times P(D|O, B) \quad (3)$$

$$P(O, D|B) = P(D|B) \times P(O|D, B) \quad (4)$$

Here, Formulas (3) and (4) are derived by expanding $P(O, D|B)$ based on multiplication theorem. Formula (3) is thought to represent the processing flow in which dependency parsing is executed after word reordering, and Formula (4) is thought to

²We pre-experimentally confirmed that the calculated result of the weighted geometric average was better than that of the weighted arithmetic average.

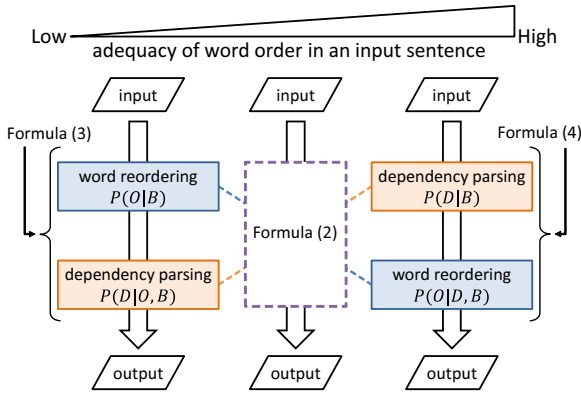


Figure 2: Relationships among Formulas (2) – (4).

represent the inverse flow. According to the probability theory, the calculated result of Formula (2) is equal to those of Formulas (3) and (4). However, in practice, since each factor in the formulas is estimated based on a training corpus, the results of these formulas are different from each other.

Figure 2 shows a conceptual diagram which represents relations among Formulas (2) – (4). If an input sentence has low adequacy of word order, it is thought that performing word reordering before dependency parsing enables $S = \langle O, D \rangle$ to be identified with higher accuracy, and thus, we can conceive an idea of calculating $P(O, D|B)$ by Formula (3). Conversely, if an input sentence has high adequacy of word order, it is probably better to perform word reordering after dependency parsing, and thus, we can think of calculating $P(O, D|B)$ by Formula (4). Therefore, we mix Formulas (3) and (4) by adjusting the weight α depending on the adequacy of word order in an input sentence, instead of using the constant 0.5 in the previous model proposed by Yoshida et al. (2014).

Each factor in Formula (2) is estimated by the maximum entropy method in the same approximation procedure as that of Yoshida et al. (2014).

4 Experiment

To evaluate the effectiveness of our method, we applied our method to less-readable sentences artificially created by changing the word order of Japanese newspaper article sentences, and evaluated how much our method could reproduce the word order of the original sentences.

4.1 Construction of Evaluation Data

From a viewpoint of utilizing our method for support revision, it is desirable to use less-readable sentences spontaneously written by Japanese natives in the experiment. However, it is not easy to collect a large amount of pairs composed of such a sentence and the corresponding sentence which was modified by hand so that the word order becomes readable, and also, such data is unavailable. In addition, since spontaneously written sentences have many factors other than word order which decrease the readability, it is difficult to conduct the evaluation with a focus solely on word order.

Therefore, our previous work (Yoshida et al., 2014) artificially generated sentences which were not easy to read, by just automatically changing the word order of newspaper article sentences in Kyoto Text Corpus³ based on the dependency structure. However, just automatically changing the word order may create sentences which are unlikely to be written by a native. To solve the problem, we semi-automatically constructed the evaluation data by adding human judgement. That is, if a subject judges that a sentence generated by automatically changing the word order in the same way as the previous work (Yoshida et al., 2014) may have spontaneously written by a native. Our constructed data has 552 sentences including 4,906 bunsetsus.

4.2 Outline of Experiment

Since our method needs to decide the weight α in Formula (2) in advance, we conducted 5-fold cross validation using the evaluation data constructed in Section 4.1. Concretely, we divided 552 sentences into 5 sets, and then, we repeated an experiment 5 times, in which we used one set from among 5 sets as the test data and the others as the held-out data to decide α . As the training data to estimate each probability in Formula (2), we used 7,976 sentences in Kyoto Text Corpus, which were different from the 552 sentences. Here, we used the Maximum Entropy Modeling Toolkit for Python and C++⁴ with the default options except “-i (iteration) 1000.”

In the evaluation of word reordering, we obtained the **complete agreement** (the percentage of the sentences in which all words’ order completely agrees with that of the original sentence)

³<http://nlp.ist.i.kyoto-u.ac.jp/EN/>

⁴http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

Table 1: Experimental results (word reordering)

	pair agreement	complete agreement
our method	83.82% (19,474/23,232)	30.98% (171/552)
Yoshida	82.90% (19,259/23,232)*	30.25% (167/552)
sequential 1	82.39% (19,140/23,232)*	26.99% (149/552)*
sequential 2	83.35% (19,365/23,232)	26.63% (147/552)*
input order	76.78% (17,838/23,232)*	0% (0/552)*

Note: The agreements followed by * differ significantly from those of our method (McNemar’s test; $p < 0.05$).

and **pair agreement** (the percentage of the pairs of bunsetsus whose word order agrees with that in the original sentence), which are defined by Uchimoto et al. (2000). Here, when deciding α using the held-out data, we calculate the α to two places of decimals which maximizes the pair agreement. In the evaluation of dependency parsing, we obtained the **dependency accuracy** (the percentage of correctly analyzed dependencies out of all dependencies) and **sentency accuracy** (the percentage of the sentences in which all the dependencies are analyzed correctly), which were defined by Uchimoto et al. (1999).

We compared our method to Yoshida’s method (Yoshida et al., 2014) and two conventional sequential methods. Both the sequential methods execute the dependency parsing primarily, and then, perform the word reordering by using the conventional word reordering method (Uchimoto et al., 1999). The difference between the two is the method of dependency parsing. The sequential methods 1 and 2 use the dependency parsing method proposed by Uchimoto et al. (2000) and the dependency parsing tool CaboCha⁵, respectively. All of the methods used the same training features as those described in Yoshida et al. (2014).

4.3 Experimental Results

Table 1 shows the experimental results on word reordering of each method. Here, the last row shows the agreements measured by comparing the input word order with the correct word order. The agreements mean the values which can be achieved with no reordering. The both agreements of our method are micro averages for the agreements of each of the 5 sets. As the result of decision of α by using the held-out data, the α for 3 sets was 0.66, and the α for the other two sets was 0.75. The both agreements of our method were highest among all. We can confirm the effectiveness of our method.

⁵<http://taku910.github.io/cabochoa/>

Table 2: Experimental results (dep. parsing)

	dependency accuracy	sentence accuracy
our method	83.39% (3,631/4,354)	40.04% (221/552)
Yoshida	82.75% (3,603/4,354)	39.49% (218/552)
sequential 1	84.75% (3,690/4,354)*	36.78% (203/552)
sequential 2	86.08% (3,748/4,354)*	37.50% (207/552)

Note: The accuracies followed by * differ significantly from those of our method (McNemar’s test; $p < 0.05$).

Although the purpose of our method is reordering to improve readability, our method generates a dependency structure as a by-product. Here, for reference, we show the experimental results on dependency parsing in Table 2. The dependency accuracy of our method was significantly lower than that of the two sequential methods, and was higher than that of Yoshida’s method although there was no significant difference. On the other hand, the sentence accuracy of our method was highest among all the methods although there were no significant differences in them. As a result of analysis, especially, our method and Yoshida’s method tended to improve the sentence accuracy very well in case of short sentences. On the other hand, CaboCha, which is a dependency parser in sequential 2, tended not to depend very well on the length of sentences.

5 Conclusion

This paper proposed the method for reordering bunsetsus in a Japanese sentence based on executing concurrently with dependency parsing. Especially, we extended the probabilistic model proposed by Yoshida et al. (2014) to deal with sentences spontaneously written by a native. In addition, we conducted the experiment using our semi-automatically constructed evaluation data so that the sentences are likely to be spontaneously written by a native. The experimental results showed the effectiveness of our method.

In the future, we would like to develop a word reordering method which can take account of comma positions by integrating our method with a method for identifying proper comma positions (for example, Murata et al., 2010).

Acknowledgments

This research was partially supported by the Grant-in-Aid for Young Scientists (B) (No.25730134) and Scientific Research (B) (No.26280082) of JSPS.

References

- Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG2011)*, pages 217–226.
- Katja Filippova and Michael Strube. 2007. Generating constituent order in German clauses. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL2007)*, pages 320–327.
- Karin Harbusch, Gerard Kempen, Camiel van Breugel, and Ulrich Koch. 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 (INLG2006)*, pages 9–11.
- Geert-Jan M. Kruijff, Ivana Kruijff-Korbayová, John Bateman, and Elke Teich. 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 (ENLG2001)*, pages 74–83.
- Nihongo Kijutsu Bunpo Kenkyukai, editor, 2009. *Gendai nihongo bunpo 7 (Contemporary Japanese Grammar 7)*, pages 165–182. Kuroshio Shuppan. (In Japanese).
- Eric Ringger, Michael Gamon, Robert C. Moore, David Rojas, Martine Smets, and Simon Corston-Oliver. 2004. Linguistically informed statistical models of constituent structure for ordering in sentence realization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING2004)*, pages 673–679.
- James Shaw and Vasileios Hatzivassiloglou. 1999. Ordering among premodifiers. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*, pages 135–143.
- Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 1999. Japanese dependency structure analysis based on maximum entropy models. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL '99)*, pages 196–203.
- Kiyotaka Uchimoto, Masaki Murata, Qing Ma, Satoshi Sekine, and Hitoshi Isahara. 2000. Word order acquisition from corpora. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING2000)*, volume 2, pages 871–877.
- Hiroshi Yokobayashi, Akira Saganuma, and Rin-ichiro Taniguchi. 2004. Generating candidates for rewriting based on an indicator of complex dependency and its application to a writing tool. *Journal of Information Processing Society of Japan*, 45(5):1451–1459. (In Japanese).
- Kazushi Yoshida, Tomohiro Ohno, Yoshihide Kato, and Shigeki Matsubara. 2014. Japanese word re-ordering integrated with dependency parsing. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING2014)*, pages 1186–1196.