Incremental Dependency Parsing of Japanese Spoken Monologue Based on Clause Boundaries

Tomohiro Ohno[†a)], *Shigeki Matsubara*[‡§], *Hideki Kashioka*[§],
Naoto Kato[§] *and Yasuyoshi Inagaki*[♮]

[†]Graduate School of Information Science, Nagoya University, Japan
[‡]Information Technology Center, Nagoya University, Japan
[§]ATR Spoken Language Translation Research Laboratories, Japan
[♮]Faculty of Information Science and Technology, Aichi Prefectural University, Japan
`a)ohno@el.itc.nagoya-u.ac.jp`

10.21437/Interspeech.2005-310

Abstract

In applications of spoken monologue processing such as simultaneous machine interpretation and real-time captions generation, incremental language parsing is strongly required. This paper proposes a technique for incremental dependency parsing of Japanese spoken monologue on a clause-by-clause basis. The technique identifies the clauses based on clause boundaries analysis, analyzes the dependency structures of them, and tries to decide the dependency relations with another clauses, simultaneously with the monologue speech input. The dependency relations are generated at the stage before the input of the entire monologue, and therefore, our technique can be used for language parsing in simultaneous Japanese speech understanding. An experiment using Japanese monologues has shown that our technique had the same degree of the performance as the usual dependency parsing for monologue sentences.

1. Introduction

In applications of spoken monologue processing such as simultaneous machine interpretation and automatic real-time captions generation, incremental language parsing is strongly required. There exist several researches about incremental parsing (e.g. [1, 2, 3]). In these researches, what kind of language unit is defined as a parsing unit becomes a point.

This paper proposes a technique for incremental dependency parsing of spoken Japanese monologue on a clause-by-clause basis. The technique identifies the clauses based on clause boundaries analysis, analyzes the dependency structures of them, and tries to decide the dependency relations with another clauses, simultaneously with the monologue speech input. The dependency relations are generated at the stage before the input of the entire monologue sentence, and therefore, our technique can be used for language parsing in simultaneous Japanese speech understanding. Furthermore, our technique identifies the dependency structure for all *bunsetsus* [1] composing a monologue which is not divided into sentences. This corresponds to the monologue's feature that it is difficult to preliminarily divide it into sentences because there is not sentence breaks clearly in monologues. An experiment using Japanese monologues has shown that our technique had the same degree of the performance as sentence-by-sentence dependency parsing.

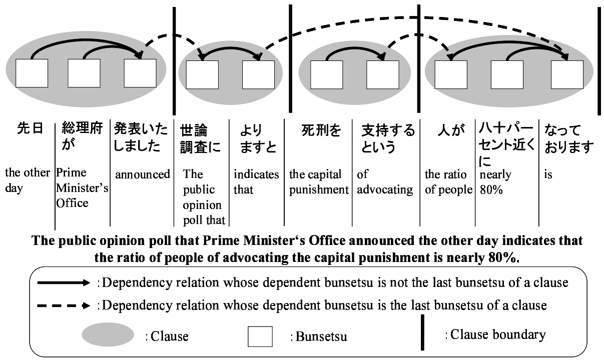

Figure 1: Relation between clause boundary and dependency structure

2. Parsing Unit of Japanese Monologues

In our research, we adopt a clause as a parsing unit and work out the incremental dependency parsing system which can output the dependency structure of a clause simultaneously with the monologue speech input. In Japanese, a clause basically contains one verb phrase. Therefore, a complex sentence or a compound sentence contains one or more clauses. Moreover, since a clause constitutes a syntactically sufficient and semantically meaningful language unit, it can be used as an alternative parsing unit to a sentence. Our proposed method assumes that a monologue is a sequence of one or more clauses, and every bunsetsu in a clause, except the last bunsetsu, depends on another bunsetsu in the same clause. As an example, the dependency structure of a part of a Japanese spoken monologue:

> "先日総理府が発表いたしました世論調査によりますと死刑を支持するという人が八十パーセント近くになっております（The public opinion poll that Prime Minister's Office announced the other day indicates that the ratio of people of advocating the capital punishment is nearly 80%.)"

is presented in Fig. 1. Here, although it is essentially difficult to divide a monologue into clauses on one dimension [4], a monologue can be approximately segmented into clauses by a clause boundary annotation program [5]. In our research, we call the unit sandwiched between two clause boundaries detected by the clause boundary analysis ***clause boundary unit*** and adopt it as an alternative paring unit.

3. Incremental Dependency Parsing Based on Clause Boundaries

In this section, we describe incremental dependency parsing based on clause boundaries. This method detects a clause

---

[1] A *bunsetsu* is one of the linguistic units in Japanese, and roughly corresponds to a basic phrase in English. A bunsetsu consists of one independent word and more than zero ancillary words. A *dependency* is a modification relation that a *dependent bunsetsu* depends on a *head bunsetsu*. That is, a dependent bunsetsu and a head bunsetsu work as a modifier and a modifyee, respectively.

boundary for speech input as needed and whenever the clause boundary unit is identified, it executes dependency parsing for a sequence of bunsetsus which was provided up to that point. The detection of a clause boundary is parsed by CBAP [5]. In dependency parsing, our method constructs the dependency structure within a clause boundary unit and decides the head bunsetsus of the last bunsetsus of clause boundary units which was previously provided if possible.

In this method, the transcribed sentence for which a morphological analysis, clause boundary detection, and bunsetsu segmentation are provided is considered as an input. In addition, in the above both procedures, our method assumes the following three syntactic constraints:

1. No dependency is directed from right to left.
2. Dependencies don't cross each other.
3. Each bunsetsu, except the last one in a sentence, depends on only one bunsetsu.

These assumptions are usually used for Japanese dependency parsing.

In what follows in the section, we describe the following processing. Note that the concrete algorithm is described in Section 4.

1. The dependency relations of a clause boundary unit inside are identified for every clause boundary unit in a monologue. **(clause-level parsing)**
2. The dependency relations of which the dependent bunsetsus is the last bunsetsus of a clause boundary units in a monologue are identified. **(monologue-level parsing)**

In this paper, we describe a sequence of clause boundary units in a monologue as $C_1, \cdots, C_m$, a sequence of bunsetsus in a clause boundary C_i as $b_1^i, \cdots, b_{n_i}^i$, a dependency relation of which a dependent bunsetsu is a bunsetsu b_k^i as $dep(b_k^i)$, and a dependency structure of a monologue as $\{dep(b_1^1), \cdots, dep(b_{n_m-1}^m)\}$.

3.1. Clause-level Dependency Parsing

Dependency parsing within a clause boundary unit, if a sequence of bunsetsus in an input clause boundary unit C_i is described as $B_i\ (= b_1^i, \cdots, b_n^i)$, identifies the dependency structure $S_i\ (= \{dep(b_1^i), \cdots, dep(b_{n_i-1}^i)\})$, which maximizes the conditional probability $P(S_i|B_i)$. At this level, the head of the last bunsetsu in a clause boundary unit is not identified.

Assuming that each dependency is independent each other, $P(S_i|B_i)$ can be calculated as follows:

$$P(S_i|B_i) \;=\; \prod_{k=1}^{n_i} P(b_k^i \overset{rel}{\rightarrow} b_l^i|B_i), \qquad (1)$$

where $P(b_k^i \overset{rel}{\rightarrow} b_l^i|B_i)$ is the probability that a bunsetsu b_k^i depends on a bunsetsu b_l^i when the sequence of bunsetsus B_i is provided. The structure S_i, which maximizes the conditional probability $P(S_i|B_i)$ is regarded as the dependency structure of B_i and calculated by dynamic programming (DP).

Next, we explain the calculation of $P(b_k^i \overset{rel}{\rightarrow} b_l^i|B_i)$. First, the basic form of independent words in a dependent bunsetsu is represented by h_k^i, its parts-of-speech t_k^i, type of dependency r_k^i, and the basic form of the independent word in a head bunsetsu h_l^i, its parts-of-speech t_l^i. Furthermore, the distance between bunsetsus is described as d_{kl}^{ii}. Here, if a dependent bunsetsu has one or more ancillary words, the type of dependency is the lexicon, part-of-speech and conjugated form of the rightmost ancillary word, and if not so, it is the part-of-speech and conjugated form of the rightmost morpheme [6]. By using the

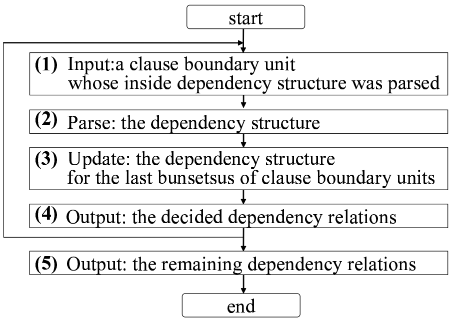

Figure 2: Flow of incremental dependency parsing

above attributes, the conditional probability $P(b_k^i \overset{rel}{\rightarrow} b_l^i|B_i)$ is calculated as follows:

$$\begin{aligned} P(b_k^i \overset{rel}{\rightarrow} b_l^i|B_i) \;&\cong\; P(b_k^i \rightarrow b_l^i|h_k^i, h_l^i, t_k^i, t_l^i, r_k^i, d_{kl}^{ii}) \quad (2)\\ &=\; \frac{F(b_k^i \rightarrow b_l^i, h_k^i, h_l^i, t_k^i, t_l^i, r_k^i, d_{kl}^{ii})}{F(h_k^i, h_l^i, t_k^i, t_l^i, r_k^i, d_{kl}^{ii})}. \end{aligned}$$

Note that F is a cooccurrence frequency function. In order to resolve sparse data problems, we adopted the smoothing method proposed by Collins [7].

3.2. Monologue-level Dependency Parsing

Here, the head bunsetsu of the last bunsetsu of a clause boundary unit is identified. Let $B\ (= B_1, \cdots, B_n)$ be the bunsetsus sequence of one monologue, and S_{last} be a set of dependency relations whose dependent bunsetsu is the last bunsetsu of a clause boundary unit, $\{dep(b_{n_1}^1), \cdots, dep(b_{n_{m-1}}^{m-1})\}$, then S_{last}, which makes $P(S_{last}|B)$ the maximum, is calculated. The $P(S_{last}|B)$ can be calculated as follows:

$$P(S_{last}|B) \;=\; \prod_{i=1}^{m-1} P(b_{n_i}^i \overset{rel}{\rightarrow} b_l^j|B), \qquad (3)$$

where $P(b_{n_i}^i \overset{rel}{\rightarrow} b_l^j|B)$ is the probability that a bunsetsu $b_{n_i}^i$ depends on a bunsetsu b_l^j when the sequence of bunsetsus B of a monologue is provided. It is calculated as in Eq. (2). The parameter S_{last}, which maximizes the conditional probability $P(S_{last}|B)$, is regarded as the dependency structure of B and is calculated by DP.

4. Algorithm of Incremental Parsing

In monologue-level dependency parsing, since it is not clear when their head bunsetsus are provided, the timing on which the dependency relation is decided is important. In our research, by taking into consideration that a dependency relation which crosses over a sentence boundary does not exist and that the length between dependent bunsetsu and head one is not long, we thought of deciding the head bunsetsu when the analysis advances to some degree after the last bunsetsu of a clause boundary unit is provided. Concretely speaking, in our method, whenever a clause boundary unit is provided, the maximum likelihood dependency structure of that point is parsed by the technique described in Section 3.2 and if a dependency relation for the last bunsetsu of a clause boundary unit does not change during a fixed input time (hereinafter referred to as a fixed value), the dependency relation is decided as having the head bunsetsu.

4.1. Algorithm

Figure 2 shows the flow of incremental dependency parsing for the last bunsetsus of clause boundary units. This algorithm executes incremental parsing by updating the dependency structure $D = \{(dep(b_{n_j}^j), k)|\ 1 \leq j \leq i-1\}$ for the last bunsetsus $b_{n_1}^1, \cdots, b_{n_{i-1}}^{i-1}$ of clause boundary units $C_1, \cdots, C_{i-1}$ which were already provided whenever a new clause boundary unit

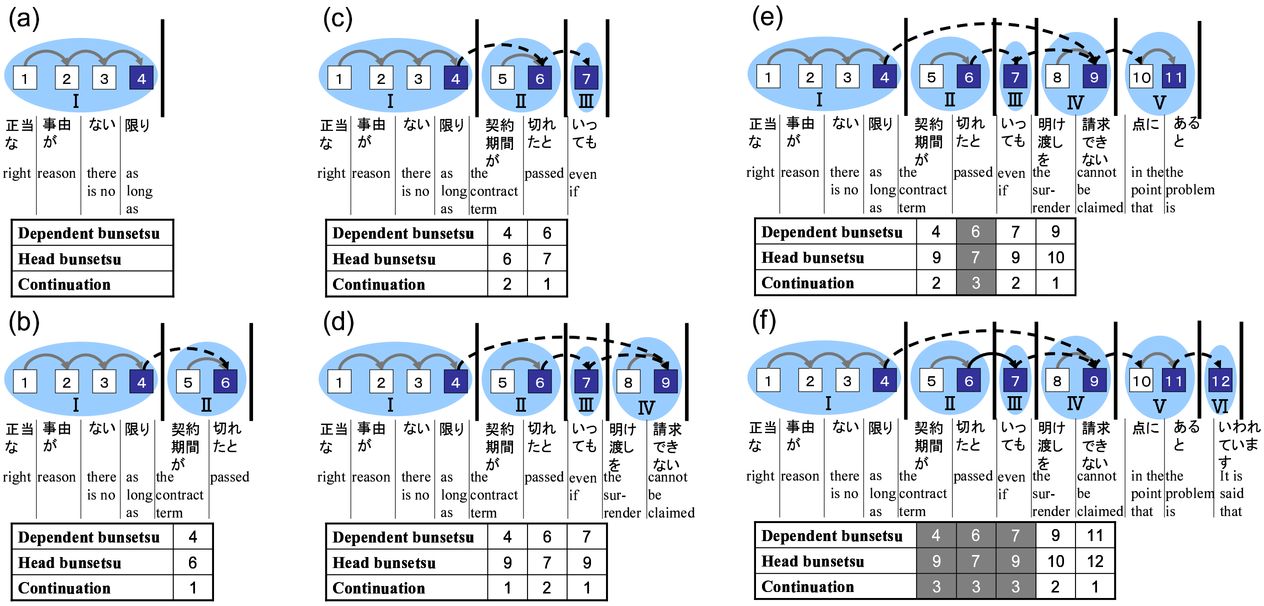

Figure 3: Example of incremental dependency parsing (in case that fixed value is 3)

C_i is provided. k is a number called *continuation*, that is, the number of times into which $dep(b^j_{n_j})$ does not change. The following indicates the algorithm of dependency parsing. Here, we describe the fixed value as σ.

(1) The clause boundary unit C_i, whose inside dependency structure was decided, is provided.

(2) The dependency relations containing a dependent bunsetsu whose head bunsetsu is not identified and which is the last bunsetsu of a clause boundary unit are parsed by the method described in Section 3.2.

(3) Based on the dependency relations $dep(b^j_{n_j})$ $(1 \leq j \leq i-1)$ which were generated in (2), the dependency structure D for the last bunsetsus is updated. Here, if $dep(b^j_{n_j})$ does not change, *continuation* k is updated into $k+1$, and if not so, it is updated into 1.

(4) Assuming that the dependency relations $(dep(b^j_{n_j}), k) \in D$ which satisfy $k = \sigma$ are reliable enough to be decided, the dependency relations are generated.

(5) After all clause boundary units were provided, the dependency relations which are undecided, that is, satisfy $k < \sigma$ in $(dep(b^j_{n_j}), k) \in D$ are generated.

In addition, our method parse the bunsetsus corresponding to the sentence end as not depending on any bunsetsu. Therefore, the probability that a bunsetsu does not have a head bunsetsu is also calculated in dependency parsing for the last bunsetsus of clause boundary units. Concretely, the probability of a bunsetsu not having a head bunsetsu can also be calculated in formula (2) by considering that such a bunsetsu depends on itself (i.e. $b^i_{n_i} = b^j_l$).

4.2. Parsing Example

Figure 3 shows the process analyzing the head bunsetsus of the last bunsetsus of clause boundary units in

"正当な事由がない限り契約期間が切れたといっても明け渡しを請求できない点にあるといわれています (It is said that the problem is in the point that the surrender cannot be claimed even if the contract term passed as long as there is no right reason),"

which is a part of a monologue. This figure consists of 6 processes (a)–(f), which respectively show the dependency structure in the top figure and the dependency relations for the last bunsetsus of clause boundary units in the bottom table. $dep(b^j_{n_j})$ and k of $(dep(b^j_{n_j}), k) \in D$ respectively correspond to "Dependent bunsetsu and Head bunsetsu" and "Continuation" in the table. Here, we explain the process based on the assumption that the fixed value is 3.

Table 1: Size of the experimental data (Asu-Wo-Yomu)

	Test data	Learning data
Program	7	95
Sentence	470	5,532
Clause	2,140	26,318
Bunsetsu	5,054	65,821
Morpheme	12,753	165,129

(a) and **(b)** respectively shows the state on which the first clause boundary unit **I** was provided and the state on which the next clause boundary unit **II** was provided and the dependency structure $\{dep(\text{限り (as long as)})\}$ was parsed. $dep(\text{限り (as long as)})$ corresponds to the dashed arrow between the dependent bunsetsu "限り (as long as)" and the head bunsetsu "切れたと (passed)" in the top figure. 1 is recorded to the continuation in the bottom table. Similarly, **(c)** and **(d)** respectively shows the state on which each maximum likelihood dependency structure $\{dep(\text{限り (as long as)}), dep(\text{切れたと (passed)})\}$, $\{dep(\text{限り (as long as)}), dep(\text{切れたと (passed)}), dep(\text{いっても (even if)})\}$ was parsed when the clause boundary unit **III**, **IV** respectively was provided.

(e) shows the state on which the new clause boundary unit **V** was provided and the maximum likelihood dependency structure $\{dep(\text{限り (as long as)}), dep(\text{切れたと (passed)}), dep(\text{いっても (even if)}), dep(\text{請求できない (cannot be claimed)})\}$ was parsed. In this time, since the continuation of the dependency relation $dep(\text{切れたと (passed)})$ reaches to 3, the dependency relation is decided and generated. **(f)** shows the state similar to **(e)**.

5. Parsing Experiment

To evaluate the effectiveness of our method for incremental dependency parsing of Japanese spoken monologue, we conducted an experiment.

Table 2: Dependency accuracy for each fixed value

Fixed value	Last bunsetsu of a clause boundary unit	Total
1	57.6% (1,228/2,133)	74.9% (3,778/5,047)
2	**60.8%**(1,296/2,133)	**76.2%**(3,847/5,047)
3	**60.8%**(1,296/2,133)	**76.2%**(3,847/5,047)
4	60.4% (1,289/2,133)	76.1% (3,840/5,047)
5	59.8% (1,276/2,133)	75.8% (3,827/5,047)
6	59.4% (1,268/2,133)	75.7% (3,819/5,047)
7	58.8% (1,254/2,133)	75.4% (3,805/5,047)
8	58.6% (1,251/2,133)	75.4% (3,803/5,047)
9	58.7% (1,253/2,133)	75.4% (3,805/5,047)
10	58.4% (1,245/2,133)	75.2% (3,797/5,047)
11	57.6% (1,229/2,133)	74.9% (3,780/5,047)
12	57.9% (1,235/2,133)	75.0% (3,786/5,047)

Table 3: Experimental result of clause boundary analysis

Recall	97.6% (2,088/2,140)
Precision	99.1% (2,088/2,106)

5.1. Outline of Experiment

We used the spoken monologue corpus "Asu-Wo-Yomu[2]." Table 1 shows the data used in the experiment. We used 7 programs (470 sentences) as the test data, to which information about a morphological analysis and bunsetsu segmentation is annotated. On the other hand, we used 95 programs (5,532 sentences), as the learning data, to which information about a morphological analysis, clause boundary detection, bunsetsu segmentation and dependency analysis is annotated.

We executed parsing by using the above-mentioned data, and evaluated the dependency accuracy and parsing time. We implemented our parsing method on Linux PC (Pentium4 2.4 GHz, with main memory 2 GB) using GNU Common Lisp. In addition, we performed the parsing experiment 12 times, by changing the fixed value described in Section 4.1 from 1 to 12.

5.2. Experimental Result

Table 2 shows the dependency accuracy for the last bunsetsus of all clause boundary units and for all bunsetsus by each fixed value. When the fixed value was 2 or 3, the dependency accuracy for the last bunsetsus of all clause boundary units was highest and the total dependency accuracy became 76.2%. Here, the dependency accuracy for all bunsetsus except the last bunsetsu within clause boundary units was 87.5%. Table 3 shows the accuracy of the clause boundary analysis, executed by CBAP. The numerator value of this table, 2,088, is the number of the clause boundary whose the position was corresponding to the correct answer. Since the recall and precision is high, the result parsed by CBAP doesn't influence the following dependency parsing so much.

The relation between the fixed value and the parsing time is shown in Fig. 4. The parsing time increases as a fixed value grows. When the fixed value was 3, the parsing time was the shortest and was 1.8 second/program.Here, the parsing time contains the time taken on not only the dependency parsing but also the clause boundary analysis. The average time taken on the clause boundary analysis was about 0.3 second/program.

From the result, in this experiment, it can be understood that our method has the highest performance when a fixed value is 3. By considering of the performance, we can confirm that our technique has the same performance as sentence-by-sentence dependency parsing method[3] implemented for comparison.

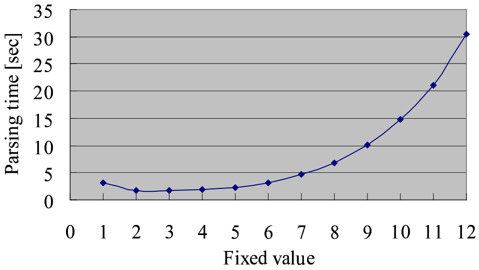

Figure 4: Relation of fixed value and parsing time

6. Conclusions

In this paper, we have proposed a technique for incremental dependency parsing of Japanese spoken monologue on a clause-by-clause basis. Our technique identifies dependency relations incrementally for a monologue which is not divided into sentences. Moreover, our technique parses a monologue by considering that the bunsetsus corresponding to sentence end do not depend on any bunsetsu. To evaluate the effectiveness of our method for incremental dependency parsing of Japanese spoken monologue, we conducted an experiment. As the result of the experiment. we have confirmed that our technique had the same degree of the performance as our past dependency parsing for monologue sentences. Future research will involve improving the accuracy of incremental dependency parsing by using information of pauses effectively.

Acknowledgements The research reported here was supported in part by a contract with the Telecommunications Advancement Organization of Japan entitled, "A study of speech dialogue translation technology based on a large corpus."

7. References

[1] Kato, Y., Matsubara, S., Toyama, K. and Inagaki, Y., "Incremental dependency parsing based on headed context-free grammar," *Systems and Computers in Japan*, 36(2):63–77, 2005.

[2] Nivre, J., "Incrementality in deterministic dependency parsing," *Proc. of the ACL-2004 Workshop Incremental Parsing*, pp.697–708, 2004.

[3] Roark, B., "Probabilistic top-down parsing and language modeling," *Computational Linguistics*, 27(2):249–276, 2001.

[4] Kashioka, T. and Maruyama, T., "Segmentation of semantic unit in Japanese monologue," *Proc. of ICSLT-O-COCOSDA*, pp.87–92, 2004.

[5] Maruyama, T., Kashioka, T. and Tanaka, H., "Development and evaluation of Japanese clause boundaries annotation program," *Journal of Natural Language Processing*, 11(3):39–68, 2004. (In Japanese)

[6] Ohno, T., Matsubara, S., Kawaguchi, N. and Inagaki, Y., "Robust dependency parsing of spontaneous Japanese spoken language," *IEICE Transactions on Information and Systems*, E88-D(3):545–552, 2005.

[7] Collins, M., "A New Statistical Parser Based on Bigram Lexical Dependencies," *Proc. of the 34th ACL*, pp. 184–191, 1996.

---

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

[3] This method executes dependency parsing by setting a sentence to the parsing unit. Moreover, this parsing is not incremental. In the result which this method parsed the above-mentioned experimental data, the dependency accuracy was 79.0% and the parsing time was about 2.1 second/program.