Dependency Parsing of Japanese Spoken Monologue Based on Clause Boundaries

Tomohiro Ohno[†a)] **Shigeki Matsubara**[‡] **Hideki Kashioka**[§]
Takehiko Maruyama[♯] and **Yasuyoshi Inagaki**[♮]
[†]Graduate School of Information Science, Nagoya University, Japan
[‡]Information Technology Center, Nagoya University, Japan
[§]ATR Spoken Language Communication Research Laboratories, Japan
[♯]The National Institute for Japanese Language, Japan
[♮]Faculty of Information Science and Technology, Aichi Prefectural University, Japan
[a)]ohno@el.itc.nagoya-u.ac.jp

Abstract

Spoken monologues feature greater sentence length and structural complexity than do spoken dialogues. To achieve high parsing performance for spoken monologues, it could prove effective to simplify the structure by dividing a sentence into suitable language units. This paper proposes a method for dependency parsing of Japanese monologues based on sentence segmentation. In this method, the dependency parsing is executed in two stages: at the clause level and the sentence level. First, the dependencies within a clause are identified by dividing a sentence into clauses and executing stochastic dependency parsing for each clause. Next, the dependencies over clause boundaries are identified stochastically, and the dependency structure of the entire sentence is thus completed. An experiment using a spoken monologue corpus shows this method to be effective for efficient dependency parsing of Japanese monologue sentences.

1 Introduction

Recently, monologue data such as a lecture and commentary by a professional have been considered as human valuable intellectual property and have gathered attention. In applications, such as automatic summarization, machine translation and so on, for using these monologue data as intellectual property effectively and efficiently, it is necessary not only just to accumulate but also to structure the monologue data. However, few attempts have been made to parse spoken monologues. Spontaneously spoken monologues include a lot of grammatically ill-formed linguistic phenomena such as fillers, hesitations and self-repairs. In order to robustly deal with their extra-grammaticality, some techniques for parsing of dialogue sentences have been proposed (Core and Schubert, 1999; Delmonte, 2003; Ohno et al., 2005b). On the other hand, monologues also have the characteristic feature that a sentence is generally longer and structurally more complicated than a sentence in dialogues which have been dealt with by the previous researches. Therefore, for a monologue sentence the parsing time would increase and the parsing accuracy would decrease. It is thought that more effective, high-performance spoken monologue parsing could be achieved by dividing a sentence into suitable language units for simplicity.

This paper proposes a method for dependency parsing of monologue sentences based on sentence segmentation. The method executes dependency parsing in two stages: at the clause level and at the sentence level. First, a dependency relation from one *bunsetsu*[1] to another within a clause is identified by dividing a sentence into clauses based on clause boundary detection and then executing stochastic dependency parsing for each clause. Next, the dependency structure of the entire sentence is completed by identifying the dependencies over clause boundaries stochastically. An experiment on monologue dependency parsing showed that the parsing time can be drasti-

[1] A *bunsetsu* is the linguistic unit in Japanese that 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 in which a *dependent bunsetsu* depends on a *head bunsetsu*. That is, the dependent bunsetsu and the head bunsetsu work as modifier and modifyee, respectively.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 169–176,
Sydney, July 2006.

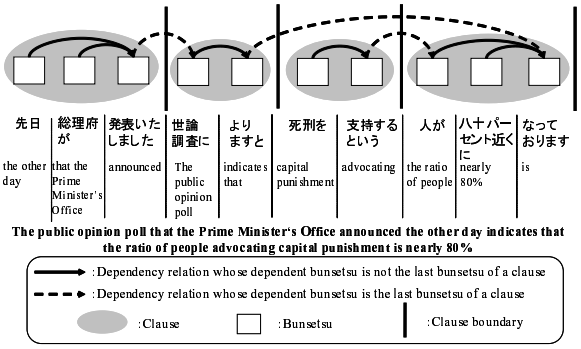

Figure 1: Relation between clause boundary and dependency structure

cally shortened and the parsing accuracy can be increased.

This paper is organized as follows: The next section describes a parsing unit of Japanese monologue. Section 3 presents dependency parsing based on clause boundaries. The parsing experiment and the discussion are reported in Sections 4 and 5, respectively. The related works are described in Section 6.

2 Parsing Unit of Japanese Monologues

Our method achieves an efficient parsing by adopting a shorter unit than a sentence as a parsing unit. Since the search range of a dependency relation can be narrowed by dividing a long monologue sentence into small units, we can expect the parsing time to be shortened.

2.1 Clauses and Dependencies

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 sentence is a sequence of one or more clauses, and every bunsetsu in a clause, except the final bunsetsu, depends on another bunsetsu in the same clause. As an example, the dependency structure of the Japanese sentence:

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 the Prime Minister's Office announced the other day indicates that the ratio of people advocating capital punishment is nearly 80%)

is presented in Fig. 1. This sentence consists of four clauses:

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

Each clause forms a dependency structure (solid arrows in Fig. 1), and a dependency relation from the final bunsetsu links the clause with another clause (dotted arrows in Fig. 1).

2.2 Clause Boundary Unit

In adopting a clause as an alternative parsing unit, it is necessary to divide a monologue sentence into clauses as the preprocessing for the following dependency parsing. However, since some kinds of clauses are embedded in main clauses, it is fundamentally difficult to divide a monologue into clauses in one dimension (Kashioka and Maruyama, 2004).

Therefore, by using a clause boundary annotation program (Maruyama et al., 2004), we approximately achieve the clause segmentation of a monologue sentence. This program can identify units corresponding to clauses by detecting the end boundaries of clauses. Furthermore, the program can specify the positions and types of clause boundaries simply from a local morphological analysis. That is, for a sentence morphologically analyzed by ChaSen (Matsumoto et al., 1999), the positions of clause boundaries are identified and clause boundary labels are inserted there. There exist 147 labels such as "compound clause" and "adnominal clause." [2]

In our research, we adopt the unit sandwiched between two clause boundaries detected by clause boundary analysis, were called the ***clause boundary unit***, as an alternative parsing unit. Here, we regard the label name provided for the end boundary of a clause boundary unit as that unit's type.

[2]The labels include a few other constituents that do not strictly represent clause boundaries but can be regarded as being syntactically independent elements, such as "topicalized element," "conjunctives," "interjections," and so on.

Table 1: 200 sentences in "Asu-Wo-Yomu"

sentences	200
clause boundary units	951
bunsetsus	2,430
morphemes	6,017
dependencies over clause boundaries	94

2.3 Relation between Clause Boundary Units and Dependency Structures

To clarify the relation between clause boundary units and dependency structures, we investigated the monologue corpus "Asu-Wo-Yomu [3]." In the investigation, we used 200 sentences for which morphological analysis, bunsetsu segmentation, clause boundary analysis, and dependency parsing were automatically performed and then modified by hand. Here, the specification of the parts-of-speech is in accordance with that of the IPA parts-of-speech used in the ChaSen morphological analyzer (Matsumoto et al., 1999), the rules of the bunsetsu segmentation with those of CSJ (Maekawa et al., 2000), the rules of the clause boundary analysis with those of Maruyama et al. (Maruyama et al., 2004), and the dependency grammar with that of the Kyoto Corpus (Kurohashi and Nagao, 1997).

Table 1 shows the results of analyzing the 200 sentences. Among the 1,479 bunsetsus in the difference set between all bunsetsus (2,430) and the final bunsetsus (951) of clause boundary units, only 94 bunsetsus depend on a bunsetsu located outside the clause boundary unit. This result means that 93.6% (1,385/1,479) of all dependency relations are within a clause boundary unit. Therefore, the results confirmed that the assumption made by our research is valid to some extent.

3 Dependency Parsing Based on Clause Boundaries

In accordance with the assumption described in Section 2, in our method, the transcribed sentence on which morphological analysis, clause boundary detection, and bunsetsu segmentation are performed is considered the input [4]. The dependency parsing is executed based on the following procedures:

1. **Clause-level parsing:** The internal dependency relations of clause boundary units are identified for every clause boundary unit in one sentence.
2. **Sentence-level parsing:** The dependency relations in which the dependent unit is the final bunsetsu of the clause boundary units are identified.

In this paper, we describe a sequence of clause boundary units in a sentence as $C_1 \cdots C_m$, a sequence of bunsetsus in a clause boundary unit C_i as $b^i_1 \cdots b^i_{n_i}$, a dependency relation in which the dependent bunsetsu is a bunsetsu b^i_k as $dep(b^i_k)$, and a dependency structure of a sentence as $\{dep(b^1_1), \cdots, dep(b^m_{n_m-1})\}$.

First, our method parses the dependency structure $\{dep(b^i_1), \cdots, dep(b^i_{n_i-1})\}$ within the clause boundary unit whenever a clause boundary unit C_i is inputted. Then, it parses the dependency structure $\{dep(b^1_{n_1}), \cdots, dep(b^{m-1}_{n_{m-1}})\}$, which is a set of dependency relations whose dependent bunsetsu is the final bunsetsu of each clause boundary unit in the input sentence. In addition, in both of the above 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 final one in a sentence, depends on only one bunsetsu.

These constraints are usually used for Japanese dependency parsing.

3.1 Clause-level Dependency Parsing

Dependency parsing within a clause boundary unit, when the sequence of bunsetsus in an input clause boundary unit C_i is described as B_i $(= b^i_1 \cdots b^i_{n_i})$, identifies the dependency structure S_i $(= \{dep(b^i_1), \cdots, dep(b^i_{n_i-1})\})$, which maximizes the conditional probability $P(S_i|B_i)$. At this level, the head bunsetsu of the final bunsetsu $b^i_{n_i}$ of a clause boundary unit is not identified.

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

$$P(S_i|B_i) = \prod_{k=1}^{n_i-1} P(b^i_k \overset{rel}{\rightarrow} b^i_l|B_i), \qquad (1)$$

[3] 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.

[4] It is difficult to preliminarily divide a monologue into sentences because there are no clear sentence breaks in monologues. However, since some methods for detecting sentence boundaries have already been proposed (Huang and Zweig, 2002; Shitaoka et al., 2004), we assume that they can be detected automatically before dependency parsing.

where $P(b_k^i \stackrel{rel}{\rightarrow} b_l^i|B_i)$ 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. Unlike the conventional stochastic sentence-by-sentence dependency parsing method, in our method, B_i is the sequence of bunsetsus that constitutes not a sentence but a clause. 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 $P(b_k^i \stackrel{rel}{\rightarrow} b_l^i|B_i)$. First, the basic form of independent words in a dependent bunsetsu is represented by h_k^i, its parts-of-speech t_k^i, and type of dependency r_k^i, while the basic form of the independent word in a head bunsetsu is represented by h_l^i, and 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. The type of dependency r_k^i is the same attribute used in our stochastic method proposed for robust dependency parsing of spoken language dialogue (Ohno et al., 2005b). Then d_{kl}^{ii} takes 1 or more than 1, that is, a binary value. Incidentally, the above attributes are the same as those used by the conventional stochastic dependency parsing methods (Collins, 1996; Ratnaparkhi, 1997; Fujio and Matsumoto, 1998; Uchimoto et al., 1999; Charniak, 2000; Kudo and Matsumoto, 2002).

Additionally, we prepared the attribute e_l^i to indicate whether b_l^i is the final bunsetsu of a clause boundary unit. Since we can consider a clause boundary unit as a unit corresponding to a simple sentence, we can treat the final bunsetsu of a clause boundary unit as a sentence-end bunsetsu. The attribute that indicates whether a head bunsetsu is a sentence-end bunsetsu has often been used in conventional sentence-by-sentence parsing methods (e.g. Uchimoto et al., 1999).

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

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

Note that F is a co-occurrence frequency function.

In order to resolve the sparse data problems caused by estimating $P(b_k^i \stackrel{rel}{\rightarrow} b_l^i|B_i)$ with formula (2), we adopted the smoothing method described by Fujio and Matsumoto (Fujio and Matsumoto, 1998): if $F(h_k^i, h_l^i, t_k^i, t_l^i, r_k^i, d_{kl}^{ii}, e_l^i)$ in formula (2) is 0, we estimate $P(b_k^i \stackrel{rel}{\rightarrow} b_l^i|B_i)$ by using formula (3).

$$\begin{aligned} & P(b_k^i \stackrel{rel}{\rightarrow} b_l^i|B_i) \qquad (3) \\ \cong\ & P(b_k^i \stackrel{rel}{\rightarrow} b_l^i|t_k^i, t_l^i, r_k^i, d_{kl}^{ii}, e_l^i) \\ =\ & \frac{F(b_k^i \stackrel{rel}{\rightarrow} b_l^i, t_k^i, t_l^i, r_k^i, d_{kl}^{ii}, e_l^i)}{F(t_k^i, t_l^i, r_k^i, d_{kl}^{ii}, e_l^i)} \end{aligned}$$

3.2 Sentence-level Dependency Parsing

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

$$P(S_{fin}|B) \ = \ \prod_{i=1}^{m-1} P(b_{n_i}^i \stackrel{rel}{\rightarrow} b_l^j|B), \qquad (4)$$

where $P(b_{n_i}^i \stackrel{rel}{\rightarrow} b_l^j|B)$ is the probability that a bunsetsu $b_{n_i}^i$ depends on a bunsetsu b_l^j when the sequence of the sentence's bunsetsus, B, is provided. Our method parses by giving consideration to the dependency structures in each clause boundary unit, which were previously parsed. That is, the method does not consider all bunsetsus located on the right-hand side as candidates for a head bunsetsu but calculates only dependency relations within each clause boundary unit that do not cross any other relation in previously parsed dependency structures. In the case of Fig. 1, the method calculates by assuming that only three bunsetsus "人が (the ratio of people)," or "なっております (is)" can be the head bunsetsu of the bunsetsu "指示するという (advocating)."

In addition, $P(b_{n_i}^i \stackrel{rel}{\rightarrow} b_l^j|B)$ is calculated as in Eq. (5). Equation (5) uses all of the attributes used in Eq. (2), in addition to the attribute s_l^j, which indicates whether the head bunsetsu of b_l^j is the final bunsetsu of a sentence. Here, we take into

Table 2: Size of experimental data set (Asu-Wo-Yomu)

	test data	learning data
programs	8	95
sentences	500	5,532
clause boundary units	2,237	26,318
bunsetsus	5,298	65,821
morphemes	13,342	165,129

Note that the commentator of each program is different.

Table 3: Experimental results on parsing time

	our method	conv. method
average time (msec)	10.9	51.9

programming language: LISP
computer used: Pentium4 2.4 GHz, Linux

account the analysis result that about 70% of the final bunsetsus of clause boundary units depend on the final bunsetsu of other clause boundary units [5] and also use the attribute e_l^j at this phase.

$$
\begin{aligned}
& P(b_{n_i}^i \stackrel{rel}{\rightarrow} b_l^j | B) \qquad (5) \\
& \cong P(b_{n_i}^i \stackrel{rel}{\rightarrow} b_l^j | h_{n_i}^i, h_l^j, t_{n_i}^i, t_l^j, r_{n_i}^i, d_{n_i l}^{ij}, e_l^j, s_l^j) \\
& = \frac{F(b_{n_i}^i \stackrel{rel}{\rightarrow} b_l^j, h_{n_i}^i, h_l^j, t_{n_i}^i, t_l^j, r_{n_i}^i, d_{n_i l}^{ij}, e_l^j, s_l^j)}{F(h_{n_i}^i, h_l^j, t_{n_i}^i, t_l^j, r_{n_i}^i, d_{n_i l}^{ij}, e_l^j, s_l^j)}
\end{aligned}
$$

4 Parsing Experiment

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

4.1 Outline of Experiment

We used the spoken monologue corpus " Asu-Wo-Yomu, " annotated with information on morphological analysis, clause boundary detection, bunsetsu segmentation, and dependency analysis[6]. Table 2 shows the data used for the experiment. We used 500 sentences as the test data. Although our method assumes that a dependency relation does not cross clause boundaries, there were 152 dependency relations that contradicted this assumption. This means that the dependency accuracy of our method is not over 96.8% (4,646/4,798). On the other hand, we used 5,532 sentences as the learning data.

To carry out comparative evaluation of our method's effectiveness, we executed parsing for the above-mentioned data by the following two methods and obtained, respectively, the parsing time and parsing accuracy.

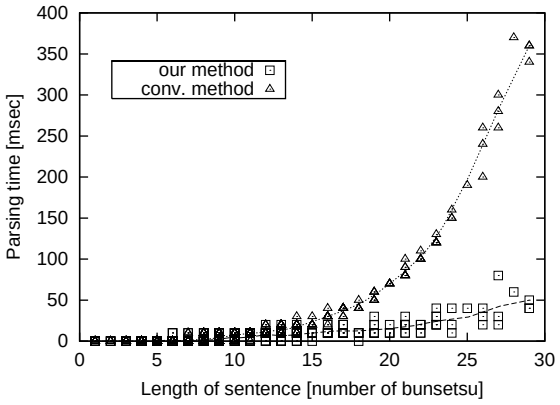

Figure 2: Relation between sentence length and parsing time

- **Our method:** First, our method provides clause boundaries for a sequence of bunsetsus of an input sentence and identifies all clause boundary units in a sentence by performing clause boundary analysis (CBAP) (Maruyama et al., 2004). After that, our method executes the dependency parsing described in Section 3.

- **Conventional method:** This method parses a sentence at one time without dividing it into clause boundary units. Here, the probability that a bunsetsu depends on another bunsetsu, when the sequence of bunsetsus of a sentence is provided, is calculated as in Eq. (5), where the attribute e was eliminated. This conventional method has been implemented by us based on the previous research (Fujio and Matsumoto, 1998).

4.2 Experimental Results

The parsing times of both methods are shown in Table 3. The parsing speed of our method improves by about 5 times on average in comparison with the conventional method. Here, the parsing time of our method includes the time taken not only for the dependency parsing but also for the clause boundary analysis. The average time taken for clause boundary analysis was about 1.2 millisecond per sentence. Therefore, the time cost of performing clause boundary analysis as a preprocessing of dependency parsing can be considered small enough to disregard. Figure 2 shows the relation between sentence length and parsing time

[5] We analyzed the 200 sentences described in Section 2.3 and confirmed 70.6% (522/751) of the final bunsetsus of clause boundary units depended on the final bunsetsu of other clause boundary units.

[6] Here, the specifications of these annotations are in accordance with those described in Section 2.3.

Table 4: Experimental results on parsing accuracy

	our method	conv. method
bunsetsu within a clause boundary unit (except final bunsetsu)	88.2% (2,701/3,061)	84.7% (2,592/3,061)
final bunsetsu of a clause boundary unit	65.6% (1,140/1,737)	63.3% (1,100/1,737)
total	80.1% (3,841/4,798)	76.9% (3,692/4,798)

Table 5: Experimental results on clause boundary analysis (CBAP)

recall	95.7% (2,140/2,237)
precision	96.9% (2,140/2,209)

for both methods, and it is clear from this figure that the parsing time of the conventional method begins to rapidly increase when the length of a sentence becomes 12 or more bunsetsus. In contrast, our method changes little in relation to parsing time. Here, since the sentences used in the experiment are composed of 11.8 bunsetsus on average, this result shows that our method is suitable for improving the parsing time of a monologue sentence whose length is longer than the average.

Table 4 shows the parsing accuracy of both methods. The first line of Table 4 shows the parsing accuracy for all bunsetsus within clause boundary units except the final bunsetsus of the clause boundary units. The second line shows the parsing accuracy for the final bunsetsus of all clause boundary units except the sentence-end bunsetsus. We confirmed that our method could analyze with a higher accuracy than the conventional method. Here, Table 5 shows the accuracy of the clause boundary analysis executed by CBAP. Since the precision and recall is high, we can assume that the clause boundary analysis exerts almost no harmful influence on the following dependency parsing.

As mentioned above, it is clear that our method is more effective than the conventional method in shortening parsing time and increasing parsing accuracy.

5 Discussions

Our method assumes that dependency relations within a clause boundary unit do not cross clause boundaries. Due to this assumption, the method cannot correctly parse the dependency relations over clause boundaries. However, the experimental results indicated that the accuracy of our method was higher than that of the conventional method.

In this section, we first discuss the effect of our method on parsing accuracy, separately for bunsetsus within clause boundary units (except the final bunsetsus) and the final bunsetsus of clause boundary units. Next, we discuss the problem of our method's inability to parse dependency relations over clause boundaries.

Table 6: Comparison of parsing accuracy between conventional method and our method (for bunsetsu within a clause boundary unit except final bunsetsu)

conv. method \ our method	correct	incorrect	total
correct	2,499	93	2,592
incorrect	202	267	469
total	2,701	360	3,061

5.1 Parsing Accuracy for Bunsetsu within a Clause Boundary Unit (except final bunsetsu)

Table 6 compares parsing accuracies for bunsetsus within clause boundary units (except the final bunsetsus) between the conventional method and our method. There are 3,061 bunsetsus within clause boundary units except the final bunsetsu, among which 2,499 were correctly parsed by both methods. There were 202 dependency relations correctly parsed by our method but incorrectly parsed by the conventional method. This means that our method can narrow down the candidates for a head bunsetsu.

In contrast, 93 dependency relations were correctly parsed solely by the conventional method. Among these, 46 were dependency relations over clause boundaries, which cannot in principle be parsed by our method. This means that our method can correctly parse almost all of the dependency relations that the conventional method can correctly parse except for dependency relations over clause boundaries.

5.2 Parsing Accuracy for Final Bunsetsu of a Clause Boundary Unit

We can see from Table 4 that the parsing accuracy for the final bunsetsus of clause boundary units by both methods is much worse than that for bunsetsus within the clause boundary units (except the final bunsetsus). This means that it is difficult

Table 7: Comparison of parsing accuracy between conventional method and our method (for final bunsetsu of a clause boundary unit)

conv. method \ our method	correct	incorrect	total
correct	1037	63	1,100
incorrect	103	534	637
total	1,140	597	1,737

Table 8: Parsing accuracy for dependency relations over clause boundaries

	our method	conv. method
recall	1.3% (2/152)	30.3% (46/152)
precision	11.8% (2/ 17)	25.3% (46/182)

to identify dependency relations whose dependent bunsetsu is the final one of a clause boundary unit.

Table 7 compares how the two methods parse the dependency relations when the dependent bunsetsu is the final bunsetsu of a clause boundary unit. There are 1,737 dependency relations whose dependent bunsetsu is the final bunsetsu of a clause boundary unit, among which 1,037 were correctly parsed by both methods. The number of dependency relations correctly parsed only by our method was 103. This number is higher than that of dependency relations correctly parsed by only the conventional method. This result might be attributed to our method's effect; that is, our method narrows down the candidates internally for a head bunsetsu based on the first-parsed dependency structure for clause boundary units.

5.3 Dependency Relations over Clause Boundaries

Table 8 shows the accuracy of both methods for parsing dependency relations over clause boundaries. Since our method parses based on the assumption that those dependency relations do not exist, it cannot correctly parse anything. Although, from the experimental results, our method could identify two dependency relations over clause boundaries, these were identified only because dependency parsing for some sentences was performed based on wrong clause boundaries that were provided by clause boundary analysis. On the other hand, the conventional method correctly parsed 46 dependency relations among 152 that crossed a clause boundary in the test data. Since the conventional method could correctly parse only 30.3% of those dependency relations, we can see that it is in principle difficult to identify the dependency relations.

6 Related Works

Since monologue sentences tend to be long and have complex structures, it is important to consider the features. Although there have been very few studies on parsing monologue sentences, some studies on parsing written language have dealt with long-sentence parsing. To resolve the syntactic ambiguity of a long sentence, some of them have focused attention on the "clause."

First, there are the studies that focused attention on compound clauses (Agarwal and Boggess, 1992; Kurohashi and Nagao, 1994). These tried to improve the parsing accuracy of long sentences by identifying the boundaries of coordinate structures. Next, other research efforts utilized the three categories into which various types of subordinate clauses are hierarchically classified based on the "scope-embedding preference" of Japanese subordinate clauses (Shirai et al., 1995; Utsuro et al., 2000). Furthermore, Kim et al. (Kim and Lee, 2004) divided a sentence into "S(ubject)-clauses," which were defined as a group of words containing several predicates and their common subject. The above studies have attempted to reduce the parsing ambiguity between specific types of clauses in order to improve the parsing accuracy of an entire sentence.

On the other hand, our method utilizes all types of clauses without limiting them to specific types of clauses. To improve the accuracy of long-sentence parsing, we thought that it would be more effective to cyclopaedically divide a sentence into all types of clauses and then parse the local dependency structure of each clause. Moreover, since our method can perform dependency parsing clause-by-clause, we can reasonably expect our method to be applicable to incremental parsing (Ohno et al., 2005a).

7 Conclusions

In this paper, we proposed a technique for dependency parsing of monologue sentences based on clause-boundary detection. The method can achieve more effective, high-performance spoken monologue parsing by dividing a sentence into clauses, which are considered as suitable language units for simplicity. To evaluate the effectiveness of our method for Japanese spoken monologue, we conducted an experiment on dependency parsing of the spoken monologue sentences recorded in the "Asu-Wo-Yomu." From the experimental re-

sults, we confirmed that our method shortened the parsing time and increased the parsing accuracy compared with the conventional method, which parses a sentence without dividing it into clauses.

Future research will include making a thorough investigation into the relation between dependency type and the type of clause boundary unit. After that, we plan to investigate techniques for identifying the dependency relations over clause boundaries. Furthermore, as the experiment described in this paper has shown the effectiveness of our technique for dependency parsing of long sentences in spoken monologues, so our technique can be expected to be effective in written language also. Therefore, we want to examine the effectiveness by conducting the parsing experiment of long sentences in written language such as newspaper articles.

8 Acknowledgements

This research was supported in part by a contract with the Strategic Information and Communications R&D Promotion Programme, Ministry of Internal Affairs and Communications and the Grand-in-Aid for Young Scientists of JSPS. The first author is partially supported by JSPS Research Fellowships for Young Scientists.

References

R. Agarwal and L. Boggess. 1992. A simple but useful approach to conjunct indentification. In *Proc. of 30th ACL*, pages 15–21.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of 1st NAACL*, pages 132–139.

M. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proc. of 34th ACL*, pages 184–191.

Mark G. Core and Lenhart K. Schubert. 1999. A syntactic framework for speech repairs and other disruptions. In *Proc. of 37th ACL*, pages 413–420.

R. Delmonte. 2003. Parsing spontaneous speech. In *Proc. of 8th EUROSPEECH*, pages 1999–2004.

M. Fujio and Y. Matsumoto. 1998. Japanese dependency structure analysis based on lexicalized statistics. In *Proc. of 3rd EMNLP*, pages 87–96.

J. Huang and G. Zweig. 2002. Maximum entropy model for punctuation annotation from speech. In *Proc. of 7th ICSLP*, pages 917–920.

H. Kashioka and T. Maruyama. 2004. Segmentation of semantic unit in Japanese monologue. In *Proc. of ICSLT-O-COCOSDA 2004*, pages 87–92.

M. Kim and J. Lee. 2004. Syntactic analysis of long sentences based on s-clauses. In *Proc. of 1st IJCNLP*, pages 420–427.

T. Kudo and Y. Matsumoto. 2002. Japanese dependency analyisis using cascaded chunking. In *Proc. of 6th CoNLL*, pages 63–69.

S. Kurohashi and M. Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

S. Kurohashi and M. Nagao. 1997. Building a Japanese parsed corpus while improving the parsing system. In *Proc. of 4th NLPRS*, pages 451–456.

K. Maekawa, H. Koiso, S. Furui, and H. Isahara. 2000. Spontaneous speech corpus of Japanese. In *Proc. of 2nd LREC*, pages 947–952.

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

Y. Matsumoto, A. Kitauchi, T. Yamashita, and Y. Hirano, 1999. *Japanese Morphological Analysis System ChaSen version 2.0 Manual*. NAIST Technical Report, NAIST-IS-TR99009.

T. Ohno, S. Matsubara, H. Kashioka, N. Kato, and Y. Inagaki. 2005a. Incremental dependency parsing of Japanese spoken monologue based on clause boundaries. In *Proc. of 9th EUROSPEECH*, pages 3449–3452.

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

A. Ratnaparkhi. 1997. A liner observed time statistical parser based on maximum entropy models. In *Proc. of 2nd EMNLP*, pages 1–10.

S. Shirai, S. Ikehara, A. Yokoo, and J. Kimura. 1995. A new dependency analysis method based on semantically embedded sentence structures and its performance on Japanese subordinate clause. *Journal of Information Processing Society of Japan*, 36(10):2353–2361. (In Japanese).

K. Shitaoka, K. Uchimoto, T. Kawahara, and H. Isahara. 2004. Dependency structure analysis and sentence boundary detection in spontaneous Japanese. In *Proc. of 20th COLING*, pages 1107–1113.

K. Uchimoto, S. Sekine, and K. Isahara. 1999. Japanese dependency structure analysis based on maximum entropy models. In *Proc. of 9th EACL*, pages 196–203.

T. Utsuro, S. Nishiokayama, M. Fujio, and Y. Matsumoto. 2000. Analyzing dependencies of Japanese subordinate clauses based on statistics of scope embedding preference. In *Proc. of 6th ANLP*, pages 110–117.