Dependency Parsing of Japanese Spoken Monologue
Based on Clause-Starts Detection

Tomohiro Ohno\textsuperscript{1}, Shigeki Matsubara\textsuperscript{2,3}, Hideki Kashioka\textsuperscript{3,4}, Yasuyoshi Inagaki\textsuperscript{5}

\textsuperscript{1}Graduate School of International Development, Nagoya University, Japan
\textsuperscript{2}Information Technology Center, Nagoya University, Japan
\textsuperscript{3}National Institute of Information and Communications Technology, Japan
\textsuperscript{4}ATR Spoken Language Communication Research Laboratories, Japan
\textsuperscript{5}Faculty of Engineering, Toyohashi University of Technology, Japan

Abstract

A dependency parsing method based on sentence segmentation into clauses has been proposed and confirmed to be effective. In this method, dependency parsing is executed in two stages: at the clause level and the sentence level. However, since a sentence cannot be segmented into complete clauses, in the past research, a unit sandwiched between two clause-end boundaries (clause boundary unit) was adopted as an approximate unit of the complete clause. There is the problem that the dependency structure of the clause boundary unit is not necessarily closed. This paper proposes a method for dependency parsing based on sentence segmentation into units which correspond to clauses and whose dependency structure is completely closed (clause fragment). Our method identifies such the unit by re-dividing a clause boundary unit at clause-start boundaries. As the results of the experiment show, we confirmed the improvement of the dependency parsing accuracy by utilizing the clause fragment as a parsing unit.

Index Terms: clause boundary analysis, parsing, dependency structure, spoken language, corpus

1. Introduction

To achieve efficient access or effective reuse of such spoken monologue data as professional lectures and commentaries, it is desirable for monologue speech to be structured and then accumulated. As one of the elemental technologies, high-performance parsing is necessary. Monologues are generally characterized by longer sentences. Generally speaking, the longer the sentence is, the more complex its dependency structure becomes. Therefore, in monologue sentences, parsing time will increase and parsing accuracy will decrease. These problems are caused in not only during parsing but also during various language processing such as machine translation or automatic summarization. To solve these problems, some methods which execute the processing after dividing a sentence into units which correspond to clauses have been proposed and their effectiveness has been confirmed\cite{1, 2, 3, 4, 5}.

This paper proposes a method for dependency parsing of monologue sentences based on sentence segmentation. In our method, we divide a sentence into the units which correspond to clauses and in which the dependency structure is closed. Then, we use the divided unit as the parsing unit. Since, in the divided unit, the dependency structure is necessarily closed, dependency parsing can be executed in the unit, and problems such as the increase of parsing time and the decrease of parsing accuracy can be alleviated. In our research, in order to detect such units, we focus attention on the clause-starts and clause-ends.

A lot of research exits regarding detecting clause-starts or clause-ends in English. In this research, detecting methods based on the rules made by hands\cite{1, 2} and the detecting methods based on machine learning\cite{6, 7} were proposed and achieved high accuracy. On the other hand, in Japanese, since predicates are located at the end of a clause, the clause-ends can be detected with considerable accuracy by taking into account the inflected forms of a predicate, connection-particle and so on\cite{8}. In contrast, detecting Japanese clause-starts is not easy because a specific word does not appear at the start of a clause, like a relative pronoun in English. There is little research on the detection of Japanese clause-starts. Therefore, most conventional research, which uses the segmentation of a Japanese sentence, divides a sentence by only clause-ends\cite{3, 4, 5}. However, in the unit divided by only clause-ends, the dependency structure is not closed just in case embedded clauses exist. Our method divides a sentence into a unit, which corresponds to a clause and in which the dependency structure is closed, by detecting not only clause-ends but also clause-starts. Furthermore, our method executes dependency parsing for each of the divided units. An experiment on monologue dependency parsing showed that compared to the conventional method which divides a sentence based only on clause-ends, the dependency parsing accuracy is improved by our method.

2. Parsing unit of Japanese spoken monologue sentences

The parsing unit of our research corresponds to a clause and is the unit in which the dependency structure is closed.

2.1. Parsing unit based only on clause-ends

Since a clause is shorter than a sentence and is the unit in which the dependency structure is closed, it is considered to be the language unit which can be utilized as a parsing unit of dependency parsing. However, when an embedded clause exists, we cannot divide a sentence into clauses one-dimensionally.

Therefore, as described in Section 1, in Japanese, sentence segmentation based on clause-ends has been performed as a preprocessing of various kinds of language processing. One of the studies has proposed a method which detects the clause-ends cyclopaedically by using a clause boundary annotation program (CBAp)\cite{8} and then executes dependency parsing based on the divided units\cite{3}. In this research, the unit sandwiched between two clause-ends is called the clause boundary unit and adopted as an alternative unit of sentence parsing. However, there is
the problem that the dependency structure of a clause boundary unit is not closed in it when an embedded clause exists. As an example, the dependency structure of the following Japanese sentence:

Idenshi-wo kaidoku-suru-toyō kenkyu-ga minkan-kigyo-mo sinka-shite hageshi kyoso-no naka-de ima susume-rare-te-i-masu (The research is now advanced in intense competition) (The research to decode a gene, in which private enterprises also participate, is now advanced in intense competition) is presented in Fig. 1. In this figure, the dependency relation between the bunsetsu “kenkyu-ga (the research)” and the bunsetsu “susume-rare-te-i-masu (is advanced)” is not closed in the clause boundary unit “kenkyu-ga minkan-kigyo-mo sinka-shite (the research, in which private enterprises also participate)”.

This is because a bunsetsu “kenkyu-ga (the research)” which intrinsically constitutes a clause “kenkyu-ga hageshi kyoso-no naka-de ima susume-rare-te-i-masu (the research is now advanced intense competition)” constitutes the same clause boundary unit as that of the embedded clause “minkan-kigyo-mo sinka-shite (in which private enterprises also participate)”.

### 2.2. Parsing unit based on clause-starts and clause-ends

In our research, we define the approximate unit of a clause, which can be divided into one-dimensionally and in which the dependency structure is closed, for use as the parsing unit of dependency parsing. Our research describes the unit sandwiched between clause-starts or clause-ends as a **clause fragment** and adopts it as an alternative parsing unit. Figure 2 shows an example of clause fragments. As shown in this figure, even if an embedded clause exists, the dependency structure of each clause fragment is inevitably closed within it. This is because by using not only clause-ends but also clause-starts we can identify the range of the embedded clause “minkan-kigyo-mo sinka-shite (in which private enterprises also participate)” that is, divide the clause “kenkyu-ga hageshi kyoso-no naka-de ima susume-rare-te-i-masu (the research is now advanced in intense competition)” into the front fragment “kenkyu-ga (the research)” and the rear fragment “hageshi kyoso-no naka-de ima susume-rare-te-i-masu (is now advanced in intense competition)”.

Here, our proposed method assumes that a sentence is a sequence of one or more clause fragments, and every bunsetsu in a clause fragment, except the final bunsetsu, depends on another bunsetsu in the same clause fragment.

3. **Identification of clause fragments**

To use the clause fragment as a parsing unit, it is necessary to divide a sentence into clause fragments as the preprocessing of dependency parsing. In this section, we describe a method for identifying of clause fragments. In this method, the transcribed sentence, on which morphological analysis and bunsetsu segmentation are performed, is considered the input. By the following procedure, our method identifies clause fragments.

1. **Identification of clause boundary units**: The clause-ends in an input sentence are detected by using CBAP[8] and then the clause boundary units are identified.

2. **Division of clause boundary units**: The modifier bunsetsus of dependency relations which are not closed in a clause boundary unit are detected and then the clause

---

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 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.

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

3 *CBAP*[8] can also specify the 147 types of clause boundary units such as “Compound,” “Adnominal” and “Topicalized element wa.”
In what follows, we describe the detection of the modifier bunsetsu of the dependency relation which is not closed in a clause boundary unit.

3.1. Detection of the dependency relations which are not closed in a clause boundary unit

In the algorithm, the sequence of bunsetsus in a sentence is provided as the input. The algorithm repeatedly determines whether each bunsetsu in a clause boundary unit (except the final bunsetsu) or is or is not the modifier bunsetsu of the dependency relation which is not closed in the clause boundary unit. The determination is based on the maximum entropy method. The probability that a bunsetsu is a modifier bunsetsu of a dependency relation which is not closed in a clause boundary unit in a context is estimated and then if the probability value is more than a threshold value \( \alpha \), the bunsetsu is judged as a modifier bunsetsu of a dependency relation which is not closed in a clause boundary unit.

We explain the features utilized in the detecting model based on the maximum entropy method. Our research used the following features on the following four language units based on the results of the analysis of Section 2.3. Figure 3 shows the four language units based on the maximum entropy method.

1) **Focused bunsetsu** (This is the bunsetsu which is currently being judged as to whether or not it is a modifier bunsetsu of a dependency relation which is not closed in a clause boundary unit.)
   - basic form and part-of-speech of the rightmost independent word
   - token and part-of-speech of the rightmost morpheme
   - token and part-of-speech of the rightmost particle
   - whether or not there exists a pause right after this bunsetsu

2) **Focused clause boundary unit** (This is the clause boundary unit which contains the focused bunsetsu.)
   - type
   - whether or not there exists a bunsetsu which has the same case as the focused bunsetsu

3) **Bunsetsu right after the focused clause boundary unit**
   - whether or not this bunsetsu has the same basic form of the rightmost independent word as the focused bunsetsu.
   (This feature is used only when the type of the focused clause boundary unit is “Adnominal.”)

4) **Bunsetsu which is a predicate right after the focused clause boundary unit**
   - whether or not the dependency probability\(^4\) between this bunsetsu and the focused bunsetsu is higher than that of any other bunsetsu in the focused clause boundary unit.
   (This feature is used only when the type of the focused clause boundary unit is “Topicalized element wa,” “Adnominal” or “Compound koto.”)

4. Two-stage dependency parsing

In this method, the transcribed sentence, on which morphological analysis, bunsetsu segmentation, and clause fragments identification are performed, is considered the input. Dependency parsing is executed based on the following procedures:

1. **Clause-level parsing:** The internal dependency relations of clause fragments are identified for every clause fragment in one sentence.

2. **Sentence-level parsing:** Dependency relations are identified in which the modifier bunsetsu is the final bunsetsu of the clause fragment.

In this paper, we describe a sequence of clause fragments in a sentence as \( C_1 \cdots C_m \), a sequence of bunsetsu in a clause fragment \( C_i \) as \( b_1 \cdots b_{n_i} \), a dependency relation in which the modifier bunsetsu is bunsetsu \( b_i \) as \( \text{dep}(b_i) \), and a dependency structure of the sentence as \( \{ \text{dep}(b_1), \cdots, \text{dep}(b_{n_i-1}) \} \).

4.1. Clause-level dependency parsing

When the sequence of bunsetsus in input clause fragment \( C_i \) is described as \( B_i = b_1^i \cdots b_{n_i}^i \), clause-level dependency parsing identifies dependency structure \( S_i = \{ \text{dep}(b_1), \cdots, \text{dep}(b_{n_i-1}) \} \), which maximizes conditional probability \( P(S_i|B_i) \). At this level, the modified bunsetsu of final bunsetsu \( b_{n_i}^i \) of a clause fragment is not identified.

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

\[
P(S_i|B_i) = \prod_{k=1}^{n_i-1} P(b_k^{\rightarrow} \leftarrow b_{k+1}^i | B_i),
\]

where \( P(b_k^{\rightarrow} \leftarrow b_{k+1}^i | B_i) \) is the probability that bunsetsu \( b_{k+1}^i \) depends on bunsetsu \( b_k^i \) when the sequence of bunsetsus \( B_i \) is provided. Structure \( S_i \), which maximizes conditional probability \( P(S_i|B_i) \), is regarded as the dependency structure of \( B_i \) and is calculated by dynamic programming (DP). \( P(b_k^{\rightarrow} \leftarrow b_{k+1}^i | B_i) \) is estimated by using the Uchimoto’s dependency probability model based on maximum entropy method\(^9\). We use almost the same features of the Uchimoto\(^9\) but eliminate some features such as punctuation or parenthesis because our method is designed for spoken language.

4.2. Sentence-level dependency parsing

Here, the modified bunsetsu of the final bunsetsu of a clause fragment is identified. Let \( B = b_1^i \cdots b_{n_i}^m \) be the sequence of bunsetsu of one sentence and \( S_{fin} \) be a set of dependency relations whose modifier bunsetsu is the final bunsetsu of a clause fragment, \( \{ \text{dep}(b_{n_i}), \cdots, \text{dep}(b_{n_{fin}-1}) \} \); then \( S_{fin} \), which makes \( P(S_{fin}|B) \) the maximum, is calculated by DP.

\[
P(S_{fin}|B) = \prod_{i=1}^{m} P(b_{n_i}^{\rightarrow} \leftarrow b_1^i | B).
\]

\( P(b_{n_i}^{\rightarrow} \leftarrow b_1^i | B) \) is calculated based on maximum entropy method similarly to Section 4.1.

5. Parsing experiment

To evaluate the effectiveness of our method for Japanese spoken monologues, we conducted an experiment on dependency parsing.
Table 1: Experimental Data (Asu-Wo-Yomu)

<table>
<thead>
<tr>
<th></th>
<th>test data</th>
<th>learning data 1</th>
<th>learning data 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentence</td>
<td>500</td>
<td>5,532</td>
<td>2,274</td>
</tr>
<tr>
<td>bunsetsu</td>
<td>5,298</td>
<td>65,762</td>
<td>27,027</td>
</tr>
<tr>
<td>morpheme</td>
<td>13,342</td>
<td>165,173</td>
<td>67,183</td>
</tr>
</tbody>
</table>

learning data 1: learning data in dependency parsing
learning data 2: learning data in detecting clause-starts

Table 2: Dependency accuracy

<table>
<thead>
<tr>
<th></th>
<th>our method</th>
<th>conv. method</th>
</tr>
</thead>
<tbody>
<tr>
<td>recall</td>
<td>28.9% (44/152)</td>
<td>1.3% (2/152)</td>
</tr>
<tr>
<td>precision</td>
<td>53.7% (74/142)</td>
<td>11.8% (2/17)</td>
</tr>
</tbody>
</table>

5.1. Outline of experiment

We used the spoken monologue corpus “Asu-Wo-Yomu” described in Section 2.3. Table 1 shows the data used for the experiment. As test data, we used 500 sentences, which were disjointed with 315 sentences in Section 2.3. Here, there exist 152 dependency relations which are not closed in the clause boundary units in the test data. On the other hand, we used 5,532 sentences as learning data in dependency parsing. Among the 5,532 sentences, 2,274 sentences, which provide time information, are used as learning data in detecting clause-starts.

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

- **Our method**: dependency parsing based on clause fragments
- **Conventional method**: dependency parsing based on clause boundary units

Here, we used the maximum entropy method tool[10] with the default options.

5.2. Experimental results

The parsing times of our method (92.56 msec/sentence) was almost the same as that of the conventional method (87.66 msec/sentence). Therefore, the time cost of detecting clause-starts as dependency parsing preprocessing can be considered small enough to disregard.

Table 2 shows the dependency accuracy of each method. The accuracy of our method increased slightly by 0.8%, compared with that of the conventional method. Here, there are only 152 dependency relations which are not closed in a clause boundary unit in the test data. Even if all the dependency relations are completely identified, the dependency accuracy is increased only by 3.2% (152/4,798). For this reason, the results show the effectiveness of our method.

Table 3 shows the parsing accuracy of dependency relations which are not closed in a clause boundary unit. Our method could identify some dependency relations which are not closed in a clause boundary unit by executing dependency parsing for each clause fragment. Next, the result of detecting dependency relations which are not closed in a clause boundary unit is shown in Table 4. The recall and precision are not so high. Detecting the clause fragments more properly is desired\(^\text{5}\). However, even if the accuracy of detecting clause fragments is not as high as hoped for, the dependency accuracy of our method improved compared with the conventional method. This reason is considered to be because the mistake is absorbed in dependency parsing, even if the detection of clause fragments is somewhat inaccurate.

\(^\text{5}\)On the assumption that the clause fragments are perfectly detected, the dependency accuracy of our method was 86.5% (4,152/4,798).

Table 3: Results of dependency parsing for dependency relations which are not closed in a clause boundary unit

<table>
<thead>
<tr>
<th></th>
<th>our method</th>
<th>conv. method</th>
</tr>
</thead>
<tbody>
<tr>
<td>recall</td>
<td>28.9% (44/152)</td>
<td>1.3% (2/152)</td>
</tr>
<tr>
<td>precision</td>
<td>53.7% (74/142)</td>
<td>11.8% (2/17)</td>
</tr>
</tbody>
</table>

Table 4: Results of detecting dependency relations which are not closed in a clause boundary unit

<table>
<thead>
<tr>
<th></th>
<th>recall</th>
<th>precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>48.7% (74/152)</td>
<td>58.7% (74/126)</td>
</tr>
</tbody>
</table>

As mentioned above, we confirmed our method can improve the parsing accuracy of the conventional method when maintaining the same parsing time.

6. Conclusions

This paper proposed a method for dependency parsing of monologue sentences based on clause fragments. From the experimental results, we confirmed that our method maintained parsing time and increased parsing accuracy compared with the conventional method, which parses a sentence based on clause boundary units. Future research will include thorough investigations of the relations between acoustic information and the dependency relations which are not closed in a clause boundary unit to detect them more correctly.

7. Acknowledgements

This research was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 20300058) of JSPS and by the Research Foundation for the Electrotechnology of Chubu.

8. References


