HPSG-Style Underspecified Japanese Grammar
with Wide Coverage

MITSUISHI Yutaka†, TORISAWA Kentaro†, TSUJII Jun’ichi‡
†Department of Information Science
Graduate School of Science, University of Tokyo*
‡CCL, UMIST, U.K.

Abstract
This paper describes a wide-coverage Japanese grammar based on HPSG. The aim of this work is to see the coverage and accuracy attainable using an underspecified grammar. Underspecification, allowed in a typed feature structure formalism, enables us to write down a wide-coverage grammar concisely. The grammar we have implemented consists of only 6 ID schemata, 68 lexical entries (assigned to functional words), and 63 lexical entry templates (assigned to parts of speech (POSs)). Furthermore, word-specific constraints such as subcategorization of verbs are not fixed in the grammar. However, this grammar can generate parse trees for 87% of the 10000 sentences in the Japanese EDR corpus. The dependency accuracy is 78% when a parser uses the heuristic that every bunsetsu1 is attached to the nearest possible one.

1 Introduction
Our purpose is to design a practical Japanese grammar based on HPSG (Head-driven Phrase Structure Grammar) (Pollard and Sag, 1994), with wide coverage and reasonable accuracy for syntactic structures of real-world texts. In this paper, “coverage” refers to the percentage of input sentences for which the grammar returns at least one parse tree, and “accuracy” refers to the percentage of bunsetsus which are attached correctly.

To realize wide coverage and reasonable accuracy, the following steps had been taken:
A) At first we prepared a linguistically valid but coarse grammar with wide coverage.
B) We then refined the grammar in regard to accuracy, using practical heuristics which are not linguistically motivated.

As for A), the first grammar we have constructed actually consists of only 68 lexical entries (LEs) for some functional words2, 63 lexical entry templates (LETs) for POSs3, and 6 ID schemata. Nevertheless, the coverage of our grammar was 92% for the Japanese corpus in the EDR Electronic Dictionary (EDR, 1996), mainly due to underspecification, which is allowed in HPSG and does not always require detailed grammar descriptions.

As for B), in order to improve accuracy, the grammar should restrict ambiguity as much as possible. For this purpose, the grammar needs more constraints in itself. To reduce ambiguity, we added additional feature structures which may not be linguistically valid but be empirically correct, as constraints to i) the original LEs and LETs, and ii) the ID schemata.

The rest of this paper describes the architecture of our Japanese grammar (Section 2), refinement of our grammar (Section 3), experimental results (Section 4), and discussion regarding errors (Section 5).

2 Architecture of Japanese Grammar
In this section we describe the architecture of the HPSG-style Japanese grammar we have developed. In the HPSG framework, a grammar consists of (i) immediate dominance schemata (ID schemata), (ii) principles, and (iii) lexical entries (LEs). All of them are represented by typed feature structures (TFSs) (Carpenter, 1992), the fundamental data structures of HPSG. ID schemata, corresponding to rewriting rules in CFG, are significant for constructing syntactic structures. The details of our ID schemata are discussed in Section 2.1. Principles are constraints between mother and daughter feature structures.4 LEs, which compose the lexicon, are detailed constraints on each word. In our grammar, we do not always assign LEs to each word. Instead, we assign lexical entry

1 A bunsetsu is a common unit when syntactic structures in Japanese are discussed.

2 A functional word is assigned one or more LEs.

3 A POS is also assigned one or more LETs.

4 We omit further explanation about principles here due to limited space.
| Schema name               | Explanation                                                                 | Example             |
|--------------------------|-----------------------------------------------------------------------------|---------------------|
| Head-complement schema   | Applied when a predicate subcategorizes a phrase.                           | Kase ga hashiru.  |
|                          |                                                                             | 'He runs.'          |
| Head-relative schema     | Applied when a relative clause modifies a phrase.                           | Aruku hitobito.    |
|                          |                                                                             | 'People who walk.' |
| Head-marker schema       | Applied when a marker like a postposition marks a phrase.                    | Kanajo ga she      |
|                          |                                                                             | 'She ...'           |
| Head-adjacent schema     | Applied when a suffix attaches to a word or a compound word.                | Iku darou go       |
|                          |                                                                             | 'will go ...'       |
| Head-compound schema     | Applied when a compound word is constructed.                                | Shizen Gengo.       |
|                          |                                                                             | 'Natural language.' |
| Head-modifier schema     | Applied when a phrase modifies another or when a coordinate structure is constructed. | Yukkuri tobu.     |
|                          |                                                                             | 'fly slowly.'       |

Table 1: ID schemata in our grammar

templates (LETs) to POSs. The details of our LEs and LETs are discussed in Section 2.2.

2.1 ID Schemata

Our grammar includes the 6 ID schemata shown in Table 1. Although they are similar to the ones used for English in standard HPSG, there is a fundamental difference in the treatment of relative clauses. Our grammar adopts the head-relative schema to treat relative clauses instead of the head-filler schema. More specifically, our grammar does not have SLASH features and does not use traces. Informally speaking, this is because SLASH features and traces are really necessary only when there are more than one verb between the head and the filler (e.g., Sentence (1)). But such sentences are rare in real-world corpora in Japanese. Just using a Head-relative schema makes our grammar simpler and thus less ambiguous.

(1) Taro ga aisuru to iu onna.
  -SUBJ love -QUOTE say woman
  'The woman who Taro says that he loves.'

2.2 Lexical Entries (LEs) and Lexical Entry Templates (LETs)

Basically, we assign LETs to POSs. For example, common nouns are assigned one LET, which has general constraints that they can be complements of predicates, that they can be a compound noun with other common nouns, and so on. However, we assign LEs to some single functional words which behave in a special way. For example, the verb 'suru' can be adjacent to some nouns unlike other ordinary verbs. The solution we have adopted is that we assign a special LE to the verb 'suru'.

Our lexicon consists of 68 LEs for some functional words, and 63 LETs for POSs. A functional word is assigned one or more LEs, and a POS is also assigned one or more LETs.

3 Refinement of our Grammar

Our goal in this section is to improve accuracy without losing coverage. Constraints to improve accuracy can also be represented by TFSs and be added to the original grammar components such as ID schemata, LEs, and LETs.

The basic idea to improve accuracy is that including descriptions for rare linguistic phenomena might make it more difficult for our system to choose the right analyses. Thus, we abandon some rare linguistic phenomena. This approach is not always linguistically valid but at least is practical for real-world corpora.

In this section, we consider some frequent linguistic phenomena, and explain how we discarded the treatment of rare linguistic phenomena in favor of frequent ones, regarding three components: (i) the postposition 'wa', (ii) relative clauses and commas and (iii) nominal suffixes representing time. The way how we abandon the treatment of rare linguistic phenomena is by introducing additional constraints in feature structures. Regarding (i) and (ii), we introduce 'pseudo-principles', which are unified with ID schemata in the same way principles are unified. Regarding (iii), we add some feature structures to LEs/LETs.

3.1 Postposition 'Wa'

The main usage of the postposition 'wa' is divided into the following two patterns:

- If two PPs with the postposition 'wa' appear consecutively, we treat the first PP as

---

5 These patterns are almost similar to the ones in (Kurohashi and Nagao, 1994).

---

877
a complement of a predicate just before the second PP.

- Otherwise, PP with the postposition ‘wa’ is treated as the complement of the last predicate in the sentence.

Sentences (2) and (3) are examples for these patterns, respectively. The parse tree for Sentence (2) corresponds to Figure 1(a), but not to Figure 1(b), and the parse tree for Sentence (3) corresponds to Figure 1(c), but not to Figure 1(d).

(2) Taro wa iku ga Jiro wa ika nai.

‘Though Taro goes, Jiro does not go.’

(3) Tokai wa hito ga ookute sawagashii.

‘A city is noisy because there are many people.’

Although there are exceptions to the above patterns (e.g., Sentence (4)), they are rarely observed in real-world corpora. Thus, we abandon their treatment.

(4) Ude wa nai ga, konjo ga aru.

‘Though he does not have ability, he has guts.’

To deal with the characteristic of ‘wa’, we introduced the WA feature and the P_WA feature. Both of them are binary features as follows:

| Feature | Value | Meaning |
|---------|-------|---------|
| WA      | +/-   | The phrase contains a/no ‘wa’. |
| P_WA    | +/-   | The PP is/isn’t marked by ‘wa’. |

We then introduced a ‘pseudo-principle’ for ‘wa’ in a disjunctive form as below:

(A) When applying head-complement schema, also apply:

\[ \text{wa \_hc}_{(-, -, -)} \Rightarrow \text{wa \_hc}_{(+, +, +)} \]

(B) When applying head-modifier schema, also apply:

\[ \text{wa \_hm}_{(-, +)} \Rightarrow \text{wa \_hm}_{(+, +, +)} \]

This treatment prunes the parse trees like those in Figure 1(b, d) as follows:

- Figure 1(b)
  1) At (\#), the head-complement schema should be applied, and (A) of the ‘pseudo-principle’ should also be applied.
  2) Since the phrase ‘iku kedo ashita wa ika nai’ contains a ‘wa’, \([2]\) is +.
  3) Since the PP ‘Kyou wa’ is marked by ‘wa’, \([3]\) is +.
  4) \(\text{wa \_hc}_{(-, -, -)} \Rightarrow \text{wa \_hc}_{(+, +, +)} \) fails.

- Figure 1(d)
  1) At (\#), the head-modifier schema should be applied, and (B) of the ‘pseudo-principle’ should also be applied.
  2) Since the phrase ‘Tokai wa hito ga ookute’ contains a ‘wa’, \([1]\) is +.
  3) Since the phrase ‘sawagashii’ contains no ‘wa’, \([2]\) is –.
  4) \(\text{wa \_hm}_{(-, +)} \Rightarrow \text{wa \_hm}_{(+, +, +)} \) fails.

3.2 Relative Clauses and Commas

Relative clauses have a tendency to contain no commas. In Sentence (5), the PP ‘Nippon de,’ is a complement of the main verb ‘atta’, not a complement of ‘umareta’ in the relative clause (Figure 3(a)), though ‘Nippon de’ is preferred to ‘umareta’ if the comma after ‘de’ does not exist (Figure 3(b)). We, therefore, abandon the treatment of relative clauses containing a
(5) Nippon de, saikin umareta akachan.

Japan -LOC recently be-born-PAST baby

ni atta.

-GOAL meet-PAST

‘In Japan I met a baby who was born recently.’

To treat such a tendency of relative clauses.
we first introduced the **TOUTEN** feature\(^7\). The **TOUTEN** feature is a binary feature which takes 
+/- if the phrase contains a/no comma. We
then introduced a ‘pseudo-principle’ for relative
clauses as follows:

(A) When applying head-relative schema, also
apply:

\[
\text{MARK} \{ \text{SYN} \{ \text{LOCAL} \{ \text{N-SUFFIX} - \}}
\]

(B) When applying other ID schemata, this
pseudo-principle has no effect.

This is to make sure that parse trees for relative
clauses with a comma cannot be produced.

### 3.3 Nominal Suffixes Representing
Time and Commas

Noun phrases (NPs) with nominal suffixes such
as nen (year), gatsu (month), and ji (hour) repre-
sent information about time. Such NPs are sometimes used adverbially, rather than nomi-
nally. Especially NPs with such a nominal suffix
and comma are often used adverbially (Sentence
(6) & Figure 4(a) ), while general NPs with a
comma are used in coordinate structures (Sen-
tence (7) & Figure 4(b) ).

(6) 1995 nen, jishin ga okita.

year earthquake -SUBJ occur-PAST

An earthquake occurred in 1995.

\(^7\)A **touten** stands for a comma in Japanese.

---

4 Experiments

We implemented our parser and grammar in
LiLFeS (Makino et al., 1998)\(^8\), a feature-
structure description language developed by our

\(^8\)LiLFeS will soon be published on its homepage,
http://www.is.s.u-tokyo.ac.jp/~mak/lilfes/
Table 2: Experimental results for 10000 sentences from the Japanese EDR Corpus: (a–e) are grammars respectively corresponding to Section 2 (a), Section 2 + Subsection 3.1 (b), Section 2 + Subsection 3.2 (c), Section 2 + Subsection 3.3 (d), and Section 2 + Section 3 (e).

|     | Coverage | Partial Accuracy | Total Accuracy |
|-----|----------|------------------|---------------|
| (a) | 91.87%   | 74.20%           | 72.61%        |
| (b) | 88.37%   | 77.50%           | 74.65%        |
| (c) | 90.75%   | 74.98%           | 73.11%        |
| (d) | 91.87%   | 74.41%           | 72.80%        |
| (e) | 87.37%   | 77.17%           | 74.65%        |

When calculating **total accuracy**, the dependencies for unparsable sentences are predicted so that every *bunsetsu* is attached to the nearest *bunsetsu*. In other words, total accuracy can be regarded as a weighted average of partial accuracy and baseline accuracy.

Table 2 lists the results of our experiments. Comparison of the results between (a) and (b–d) shows that all the three constraints improve partial accuracy and total accuracy with little coverage loss. And grammar (e) using the combination of the three constraints still works with no side effect.

We also measured average parsing time per sentence for the original grammar (a) and the fully augmented grammar (e). The parser we adopted is a naïve CKY-style parser. Table 3 gives the average parsing time per sentence for those 2 grammars. Pseudo-principles and further constraints on LEs/LETs also make parsing more time-efficient. Even though they are sometimes considered to be slow in practical application because of their heavy feature structures, actually we found them to improve speed.

Table 3: The average parsing time per sentence

|     | Average parsing time per sentence |
|-----|----------------------------------|
| (a) | 1277 (msec)                     |
| (e) | 838 (msec)                      |

in another way, a phrase with a comma, sometimes, is attached to a phrase farther than the nearest possible phrase. In (Kurohashi and Nagao, 1994), the parser always attaches a phrase with a comma to the second nearest possible phrase. We need to introduce such a constraint into our grammar.

Though the grammar (e) had the pseudo-principle prohibiting relative clauses containing commas, there were still 6 relative clauses containing commas. This can be fixed by investigating the nature of relative clauses.

6 Conclusion and Future Work

We have introduced an underspecified Japanese grammar using the HPSG framework. The techniques for improving accuracy were easy to include into our grammar due to the HPSG framework. Experimental results have shown that our grammar has wide coverage with reasonable accuracy.

Though the pseudo-principles and further constraints on LEs/LETs that we have introduced contribute to accuracy, they are too strong and therefore cause some coverage loss. One way we could prevent coverage loss is by introducing preferences for feature structures.

References

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press.

EDR (Japan Electronic Dictionary Research Institute, Ltd.). 1996. EDR electronic dictionary version 1.5 technical guide.

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

Takaki Makino, Minoru Yoshida, Kentaro Torisawa, and Tsujii Jun’ichi. 1998. LilFeS – towards a practical HPSG parser. In *COLING-ACL ‘98*, August.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press.

Kentaro Torisawa and Jun’ichi Tsujii. 1996. Computing phrasal-signs in HPSG prior to parsing. In *COLING-96*, pages 949–955, August.