Construction of Practical Japanese Parsing System Based on Lexical Functional Grammar

Hiroshi Masuichi† and Tomoko Ohkuma†

This paper describes a Japanese parsing system with a linguistically fine-grained grammar based on Lexical-Functional Grammar (LFG). The system is the first Japanese LFG parser with over 97% coverage of real-world text. We evaluated the accuracy of the system by comparing it with standard Japanese dependency parsers. The LFG parser shows roughly equivalent performance in dependency accuracy with standard parsers. It also provides reasonably accurate results of case detection.

Key Words: Lexical-Functional Grammar, Japanese grammar rule, dependency parser

1 Introduction

Deep grammatical analyses of input sentences based on theoretically sound grammar formalisms are essential for the further development of such NLP applications as machine translation, question answering, dialogue understanding, and message extraction. In this paper, we report on the development and performance of a parser for Japanese based on the Lexical-Functional Grammar (LFG) formalism (Kaplan and Bresnan 1982; Dalrymple 2001).

The Japanese LFG grammar used in this parser is being developed in relation to the Parallel Grammar (ParGram) project (Butt, King, Nino, and Segond 1999; Butt, Dyvik, King, Masuichi and Rohrer 2002). In this project, grammars for English, French, German, Japanese, Norwegian, Urdu, and other languages are under-way, sharing various design decisions within the LFG formalism. LFG assumes two levels of syntactic representation for a sentence: a c(onstituent)-structure (a tree) and an f(unctional)-structure (attribute value matrices: AVMs). Within LFG, an f-structure is meant to encode a language-universal level of analysis, allowing for cross-linguistic parallelism.

Our research goal is to construct a practical Japanese LFG parsing system with broad coverage and deep analysis for real-world text. In this paper, we describe the details of the system and

† Research & Technology Group, Fuji Xerox Co., Ltd.
* This article has been partially revised for better understanding of overseas readers.
† Butt et al. (2002) reported how far the parallelism of f-structures can be maintained across languages in the ParGram project. Frank (1999) reported on a machine translation system that takes advantage of the f-structure parallelism.
demonstrate its coverage and accuracy. For the evaluation of accuracy, we compared the outputs of the Japanese LFG parsing system with outputs of standard Bunsetsu\(^2\) dependency parsers for Japanese.

This paper is organized as follows. Section 2 introduces the ParGram project. Section 3 describes the architecture of the Japanese LFG system in the ParGram project. Section 4 presents the Japanese grammar written in the LFG formalism for the system described in Section 3. Section 5.1 demonstrates the coverage of the system, and explains our approach to evaluating the accuracy of the system and Section 5.2 reports experimental results.

2 Parallel Grammar Project

The Japanese LFG grammar is being developed within the ParGram project. In the ParGram project, a biannual meeting is held to provide opportunities to increase consistency as much as possible and avoid contradiction of f-structures among multiple languages. In the meetings we discuss the details of f-structure specifications from naming conventions and usage of attribute-values to construction policies for various sentence structures to improve the standard grammar specifications of ParGram(Butt et al. 1999).

For example, at the level of name specification, the attribute “GEND” was used to indicate the gender information for each noun in all the languages within ParGram. However, in the English and Japanese grammars, modification was made to distinguish “GEND-SEM” from “GEND”. This is because the English and Japanese grammars gave “GEND” to pronouns solely to indicate whether a pronoun refers to a male or female entity, unlike the realization of morphological gender for general nouns in the other languages such as German, French and Norwegian.

(1a) Zyon wa hon o sono tukue no ue ni oku ta.

John TOPIC/SUBJ book OBJ the table of top on put PAST.

(1b) John put a book on the table.

Figure 1 shows the f-structure generated by the Japanese LFG system for Sentence (1a). The content words Zyon (John) and hon (book) are inserted into PRED (predicate) of SUBJ (subject) and OBJ (object). On the other hand, the case maker ni (on) is inserted into PRED of OBL (oblique). This is derived from the fact that in European languages such as English, content words corresponding to SUBJ and OBJ do not need prepositions, whereas OBL and ADJUNCT

\(^2\) A Bunsetsu is a widely accepted unit of syntactic and phonological phrase structures in Japanese. It consists of at least one content word plus optionally following function words.
are accompanied by prepositions. It is possible to write Japanese grammar that takes PRED of SUBJ and OBJ to be a case maker, or PRED of OBL and ADJUNCT to be a content word. However, we have written the Japanese grammar to generate the f-structure shown in Figure 1 to maintain parallelism with the other languages in ParGram. Figure 2 shows the f-structure generated by the English LFG system for (1b) corresponding to (1a). The structure of Figure 2 is basically the same as Figure 1. On the other hand, the specifications of c-structures, which are language-dependent structures are left to grammar writers of each language in ParGram. Figure 3 and 4 are the c-structures for (1a) and (1b). Their structures are completely different.

3 Japanese LFG system

The ParGram project employs the XLE parser and grammar development platform (Maxwell III and Kaplan 1993). XLE produces packed representations, specifying all possible grammar analyses of the input. Japanese is the first Asian language employing the XLE platform.
Figure 5 shows a diagram of the Japanese LFG system. An input Japanese sentence is segmented and tagged by the ChaSen morphological analyzer (Matsumoto, Kitauchi, Yamashita, Hirano, Matsuda, Takaoka and Asahara 1999). Then the lexical entry for each word in the input is automatically created (Sentence Lexical Entry: SLE). We implemented 40 templates for lexical
entries and SLEs are produced by selecting an appropriate template for each word on the basis of information of the word and the words around it, such as part-of-speech, surface form, and conjugation. Unknown words are currently treated as nouns.

Verb Lexical Entries (VLEs), Adjective Lexical Entries (ALEs), and Adjectival Noun Lexical Entries (ANLEs) were written on the basis of the case frame information in the Japanese IPAL dictionary (IPA 1987) and have been manually enhanced. VLEs consist of 10,387 entries and 41,115 functional annotations for 2,366 verbs. ALEs and ANLEs consist of 947 entries and 2,197 functional annotations for 369 words in total. Core Lexical Entries (CLEs) include entries for basic words such as auxiliary verbs, postpositional particles and so forth, plus syntactically important nouns such as *toki* (time) and *aida* (interval). CLEs consist of 1,252 entries and 1,913 functional annotations for 675 words.

SLEs have the lowest preference; an SLE for a word is overwritten if an entry for the same word already exists in another set of entries. This mechanism is intended to recover from erroneous analyses by ChaSen. For instance, it is impossible to correctly distinguish the Japanese case marker (postpositional particle) *de* from the conjugated form *de* of the auxiliary verb *da* at the level of morphological analysis. Therefore, we ignore the SLE created for *de* regardless of the output of ChaSen. Instead, we wrote CLEs for *de* that represent the behaviors of both the

---

3 This form behaves like a sentential conjunction.
case marker _de_ and the conjugated form _de_ of the auxiliary verb _da_. XLE parses the input, considering both possibilities before selecting the one that matches the syntactic structure of the whole input sentence.

Grammar Rules (GRs) include 2,468 terms in their disjunctive normal forms and 1,223 functional annotations. GRs have been designed to meet the specifications of f-structures mentioned in Section 2. We explain GRs in detail in Section 4. The words in the input are delimited by spaces and passed on to XLE, and XLE outputs c-structures and f-structures.

XLE implements a SKIMMING mechanism (Riezler, Kaplan, Crouch, Maxwell III and Johnson 2002) to increase the coverage of a grammar. XLE goes into the skimming mode when the amount of time or memory used on the input exceeds a specified threshold. In this mode, XLE does a limited amount of work per sub-tree on constituents whose processing is incompletely. This mechanism enables the parser to avoid time-outs and memory-shortage problems and can improve the robustness of the system.

4 Japanese LFG grammar

4.1 Basic Rules

Some of the most noticeable syntactic characteristics of Japanese sentences are relatively free word order, frequent rampant pro-drop, and extensive use of complex predication. To account for the free word order, we adopted (2) as the most basic rule of our grammar.

\[ S \rightarrow PP^* V \]
\[ (↑ GF) = ↓↑ = ↓ \]

For each sentence in Japanese, native speaker judgements tend to converge on an “optimal” word order (Shibatani 1990). For instance, (3a) is an intuitively strange Japanese sentence, while (3b) is felt to be more natural. It is possible to write a grammar that does not allow sentences such as (3a). Ohtani, Miyata and Matsumoto (2000) proposed a Japanese Head-driven Phrase Structure Grammar that is very sensitive to the word order. We, however, adopt (2) to achieve broader coverage because sentences such as (3a) frequently appear in real-world text.

(3a) Singaporu e Zyon ga Tokyo kara iku ta.
    Singapore to John SUBJ Tokyo from go PAST.
    John went from Tokyo to Singapore.

(3b) Zyon ga Tokyo kara singaporu e iku ta.
    John SUBJ Tokyo from Tokyo to go PAST.
    John went from Tokyo to Singapore.
We use the following type of lexical entry to handle pro-drop. (4) is the (simplified) entry for the verb *yomu* (read).

(4) \[\text{yomu} \quad \text{V} \quad (\text{PRED}) = \text{‘yomu(↑}\text{SUBJ})(↑}\text{OBJ)’}\]
\[\text{ @(PD SUBJ)}\]
\[\text{ @(PD OBJ)}\]
\[\text{PD(GF)} = \text{@(DEFAULT (↑}\text{GF PRED) ‘pro’)}\]
\[\text{(↑}\text{GF PRON-TYPE) null)}\]
\[\text{DEFAULT(ATTRIBUTE1 VALUE1 ATTRIBUTE2 VALUE2) =}\]
\[\text{ATTRIBUTE1 = VALUE1}\]
\[\text{ATTRIBUTE2 = VALUE2}\]
\[\text{ProDrop: OT}\]

“PD” and “DEFAULT” are macro definitions, and “@” indicates a macro call. “ProDrop: OT” indicates that the Optimality Theory (Bresnan, 2000) mark “ProDrop” is added. We set the preference of “ProDrop” at the lowest level. Therefore, “@(PD SUBJ)” and “@(PD OBJ)” work only if no constituent that can be subcategorized for by the verb *yomu* exists in the input.

Because pro-drop frequently occurs in Japanese, lexical entries such as (4) are important for achieving broad coverage. (Frank, King, Kuhn and Maxwell III 2001)

We adopted relatively loosely constrained mono-clausal analyses for verbs to accommodate various types of complex predications. However, we employed multi-clausal analyses when it was reasonable to consider that a verb includes multiple predicate-argument relations (PARs). This treatment is essential for such NLP applications as question answering, dialogue understanding, and machine translation. We describe the multi-clausal analyses in 4.2.

### 4.2 Predicate-Argument Relations and Cases

Japanese case markers (e.g., *ga, o*) are frequently omitted when a particular particle, such as a topic particle (e.g., *wa, mo, or koso*) or a focus particle (e.g., *made, bakari, or sae*), is added to a noun phrase. Moreover, a case marker can denote a different case than it typically does, in particular syntactic constructions. In addition to these problems, the problem of rampant pro-drop makes it a difficult task to capture the correct PARs and detect correct cases in Japanese. Our grammar pays maximum attention to this task.

(5a) \[\text{Zyon ga yomu ta hon}\]
\[\text{John SUBJ read PAST book}\]
\[\text{John read the book.}\]
(5b) Zyon no yomu ta hon
    John SUBJ read PAST book
The book John read
(5c) Zyon no hon o yomu ta.
    John GEN book OBJ read PAST
    (Someone) read John’s book.
(5d) Zyon ga hon o yomu ta.
    John SUBJ book OBJ read PAST
    John read the book.
(5e) Zyon wa hon o yomu ta.
    John TOPIC/SUBJ book OBJ read PAST
    John read the book.
(5f) Zyon wa yomu ta hon o nagesuteru ta.
    John TOPIC/SUBJ read PAST book OBJ throw away PAST
    John threw away the book (someone) read.

For instance, *no* can be used as a SUBJ marker as seen in (5b) instead of *ga* as in (5a). However, *no* in (5c) cannot be interpreted as a SUBJ marker. *Wa* in (5e) triggers the omission of the SUBJ marker *ga* in (5d). On the other hand, *wa* in (5f) also triggers the omission of the SUBJ marker but “John” is the SUBJ of “throw away” and the SUBJ of “read” is dropped, that is, “Zyon wa” modifies *nagesuteru* rather than *yomu*.

The generalized rules for these grammatical phenomena are as follows: (I) *no* can be a SUBJ marker only in a relative clause, and (II) *wa* cannot cause a topicalization and an omission of a SUBJ marker in a relative clause. Although these rules have been widely discussed, no Japanese syntactic parser with formal rules for treating (I) and (II) has been reported. We can write the following simple rule (6) based on the LFG formalism, which represents (I) and (II). “PPsubj-no” in (6) represents a postpositional phrase with *no*, and “PPsubj” represents a postpositional phrase with a SUBJ marker other than *no*.

(6) \[ Srel \rightarrow \{ \text{PPsubj} | \text{PPsubj-no} \} \quad \text{PP* V} \]
\[
    (\uparrow \text{SUBJ}=\downarrow) \quad \uparrow \text{GF} = \downarrow \uparrow = \downarrow
\]
\[
    (\downarrow \text{TOPICALIZATION-FORM}) \neq \text{‘wa’}
\]
Another type of construction that relates to PAR analysis and case detection is a verb (a bunsetsu) which includes multiple PARs.
(7) kare ga 9gatu ni Tokyo de yuki ni huru rareru ta
he SUBJ September in Tokyo in snow by fall PASSIVE PAST
*He was fallen by snow in Tokyo in September.

(8) zyoo ga Sirayukihime ni ringo o taberu saseru ta
queen SUBJ Snow-White by apple OBJ eat let PAST
The queen made Snow White eat an apple. (The queen tempted Snow White to eat an apple.)

Sentence (7) is an example of the indirect passive (Masuoka, Nitta, Gunji and Kinsui 1997). In this case, the intransitive verb *huru* (fall) is passivized. Sentence (8) is an example of a causative sentence. We adopted multi-clausal analyses for indirect passives and causative sentences, by regarding auxiliary verbs (fragments of bunsetsus) that cause passives (e.g., reru and rareru), causatives (e.g., seru, saseru, (te-)morau, and (te-)jitadaku) as having PREDs. These analyses enable us to capture the PARs "huru-yuki(SUBJ)" in (7) and "taberu-Sirayukihime(SUBJ)-ringo(OBJ)" in (8) as well as the main PARs "reru-kare(SUBJ)-yuki(OBL)-huru(XCOMP)" in (7) and "saseru-zyoo(SUBJ)-Sirayukihime(OBL)-taberu(XCOMP)" in (8). Figure 6 shows the f-structure generated by the LFG system for (7); Figure 8 shows the f-structure for (8). The standard Bunsetsu dependency parsers for Japanese cannot capture “snow falls in Tokyo in September” or “Snow White eats an apple” because they consider a Bunsetsu as the unit of analysis.

Several grammatical issues involve PAR analysis and case detection. Table 1 shows a list of

---

4 Within LFG, although the analysis for Japanese causative sentences adopted here has already been proposed (Sells, 1985), the debate has not yet been settled as to whether Japanese causative structures are multi-clausal or mono-clausal (Matsumoto, 1996; Yokota, 2001).
Table 1  Examples of Japanese grammatical issues related to PAR and case detection

<table>
<thead>
<tr>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A case marker shift caused by potential verbs</td>
</tr>
<tr>
<td>A case marker shift caused by te-aru sentences</td>
</tr>
<tr>
<td>Behavior of particular nouns that a verb in a relative clause</td>
</tr>
<tr>
<td>subcategorize for</td>
</tr>
<tr>
<td>Case disambiguation in causative-and-passive sentences</td>
</tr>
<tr>
<td>Case disambiguation in coordinated sentences with topic particles</td>
</tr>
<tr>
<td>Case disambiguation in relative and embedded clauses with topic</td>
</tr>
<tr>
<td>particles</td>
</tr>
<tr>
<td>Case marker shifts caused by relative and embedded clauses</td>
</tr>
</tbody>
</table>

examples of the grammatical issues for which we have already written LFG rules.

4.3 Robustness Techniques

As mentioned in 4.1, we use Optimality Theory (OT) marks to delete dispreferred parses. We also use OT marks for efficiency. When ranked OT constraints are divided into groups by relative ranking, XLE processes the input in multiple passes. The core grammar consisting of the rules with OT marks in the highest ranked group is used for the first pass. If a valid parse is found, then XLE will stop. In contrast, XLE will process the input again with the core grammar plus the rules with OT marks in the second-highest-ranked group for the second pass. This multiple-pass-parsing mechanism is useful for writing rules for rare grammatical phenomena without increasing unintended ambiguity.

(9a) aruku koto ga itiban da
walk NOMINALIZER SUBJ best be
To walk is best.

(9b) *aruku ga itiban da
walk SUBJ best be

(9c) makeru ga kati da
lose SUBJ victory be
To lose is a victory.

For instance, the SUBJ marker *ga ordinarily does not follow the canonical form of verbs as shown in (9a) and (9b). However, it exceptionally occurs in some idiomatic expressions as shown in (9c). We divided the 42 OT marks in our grammar into four groups, and we put the OT mark for the rule for sentences such as (9c) in the lowest-ranked preference group. The OT marks in the lowest-ranked-preference group are added to the rules for rare idiomatic expressions or
Masuichi and Ohkuma

Construction of Practical Japanese Parsing System Based on LFG

colloquial expressions so that those rules do not affect the core grammar.

When the STANDARD grammar, which consists of all the rules described above, does not produce a complete parse, a FRAGMENT grammar (Riezler et al. 2002) that we wrote for Japanese is used. This grammar parses the input as a sequence of well-formed chunks. These chunks have both c-structures and f-structures. The set of fragment parses is then chosen on the basis of a fewest-chunk method. The ungrammatical constituents in Japanese sentences that the STANDARD grammar does not cover tend to appear in sentence-final position. Therefore, the fragment grammar is likely to output a meaningful chunk for the major part of the input sentence. Examples of fragment c-structures and f-structures are shown in Figure 7.

In this case, the adverb *tinaminsi* (for your information) in sentence-final position is regarded as an ungrammatical constituent; the STANDARD grammar assumes that a Japanese adverb cannot modify a verb to its left.\(^5\) Figure 7 shows that the whole sentence except *tinaminsi* is parsed correctly.

---

\(^5\) It is one of the principles of Japanese syntax that a bunsetsu should modify another bunsetsu to the right of it.
correctly analyzed.

5 Experimental Evaluation

5.1 Coverage

We prepared 3 types of Japanese text for evaluating the coverage of the Japanese LFG system:

(A) 10,000 sentences from the Japanese EDR corpus (EDR 1996), mainly composed of newspaper text
(B) 460 sentences from a copier manual (Fuji Xerox 2000)
(C) 9,637 sentences of eCRM text

The coverage in this paper refers to the percentage of the sentences for which the system returns at least one f-structure. All three text types consist of randomly selected unseen sentences. Most sentences in (A) and (B) are grammatical. On the other hand, (C) includes many ungrammatical and colloquial sentences, because (C) consists of transcriptions of telephone calls from customers to a customer service center.

Table 2 lists the coverage results. The total coverage for (C) is almost the same as that for (A) and (B). On the other hand, the STANDARD coverage for (C) is lower than for (A) and (B).

This difference is due to the large number of ungrammatical sentences in (C), which implies that the ratio of sentences that could be covered by the STANDARD grammar was low; the FRAGMENT grammar covered the ungrammatical sentences. The main reason for failure of analysis in all three types was time-outs caused by the sentences that even the skimming mode

<table>
<thead>
<tr>
<th></th>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Num. of sentences</strong></td>
<td>10,000</td>
<td>460</td>
<td>9,637</td>
</tr>
<tr>
<td><strong>Ave. num. of words in a sentences</strong></td>
<td>22.6</td>
<td>21.3</td>
<td>16.3</td>
</tr>
<tr>
<td><strong>Total coverage(%)</strong></td>
<td>97.3</td>
<td>98.7</td>
<td>97.9</td>
</tr>
<tr>
<td>STANDARD</td>
<td>91.6</td>
<td>94.4</td>
<td>87.0</td>
</tr>
<tr>
<td>FRAGMENT</td>
<td>5.7</td>
<td>4.3</td>
<td>10.9</td>
</tr>
<tr>
<td>Skimmed STANDARD</td>
<td>4.7</td>
<td>2.0</td>
<td>1.9</td>
</tr>
<tr>
<td>Skimmed FRAGMENT</td>
<td>1.2</td>
<td>0.7</td>
<td>0.9</td>
</tr>
<tr>
<td><strong>Ave. time. to analyze a sentence without Skimming$^6$ (sec)</strong></td>
<td>1.9</td>
<td>1.1</td>
<td>4.8</td>
</tr>
<tr>
<td><strong>Ave. num. of STANDARD results</strong></td>
<td>14.6</td>
<td>10.1</td>
<td>46.2</td>
</tr>
</tbody>
</table>

Table 2 Results of the coverage test

---

$^6$ 2.8 GHz CPU/2 GB memory
Table 2 shows that the system has over 97% coverage in total and 87% coverage with the STANDARD grammar, even for (C).

5.2 Accuracy

5.2.1 Approach

Outputs of standard Japanese parsers such as KNP (Kurohashi and Nagao 1994) and CaboCha (Kudo and Matsumoto 2002) are trees expressing Bunsetsu dependencies (Bunsetsu Trees). It is quite natural and reasonable to compare a Bunsetsu Tree with an f-structure, because a Bunsetsu Tree is essentially a simplified f-structure. We can convert an f-structure into a dependency tree by regarding a PRED in the f-structure as a node in the tree, and the outside-inside relation of the nested AVMs of the f-structure as the parent-child relation between the nodes (Pred Tree).

A problem is that one Bunsetsu unit can correspond to more than one PRED. For instance, (10a) has one Bunsetsu, but the f-structure for (10a) includes two PREDs as described in 4.2. (10b) has also one Bunsetsu but its f-structure includes two PREDs: one for the verb kaku (write) and the other for the auxiliary verb nai (not). A compound noun is also regarded as one Bunsetsu as in (10c), but its f-structure can include multiple PREDs corresponding to the nouns that form the compound noun.

Another problem is that a Bunsetsu is a language-dependent and phrase-structural constituent. As previously noted, an f-structure is meant to encode a more language universal level of analysis. On the other hand, a c-structure encodes language-particular differences in syntactic structures and constituency. Thus, information about Bunsetsus should appear in c-structures, but not in f-structures. This means that we cannot detect which part of an f-structure corresponds to a Bunsetsu. However, it is easy in general to list the grammatical categories in c-structures for Japanese, which correspond to Bunsetsus. We created a Bunsetsu Tree from a c-structure and an f-structure as follows:

1. List the grammatical categories corresponding to Bunsetsus.
2. Specify the nodes in a c-structure that correspond to Bunsetsus, referring to the category list in 1.
3. Specify the f-structure AVMs, which correspond to the nodes specified in 2 (Bunsetsu AVMs).

4. Create a Bunsetsu Tree by treating “a set of PREDs” in a Bunsetsu AVM as a node in the Bunsetsu Tree, and the outside-inside relation of the nested Bunsetsu AVMs as the parent-child relation between the nodes.

Figure 8 shows the c-/f-structures and the Pred/Bunsetsu Trees for Sentence (8). “NP”, “NPobl” and “Vverb” in the c-structure are the grammatical categories corresponding to Bunsetsu. The numbers in Figure 8 indicate the correspondence of the grammatical categories in the c-structure to the AVMs in the f-structure. We can specify the Bunsetsu AVM by referring to the correspondence, and thus create the Bunsetsu Tree. The Bunsetsu Trees are comparable to the trees generated by the standard Bunsetsu dependency parsers for Japanese.

5.2.2 Experiments and results

We randomly selected 200 unseen sentences from the EDR corpus for which the Japanese LFG system returned at least one f-structure. The 200 sentences were analyzed with KNP, CaboCha, and the Japanese LFG system. The outputs of KNP and CaboCha are Bunsetsu Trees, and Pred Trees and Bunsetsu Trees were created from the outputs of the LFG system. In this experiment, the link labels in the trees are grammatical functions as in Figure 8, and we omitted other

7 When we created the Pred Trees and the Bunsetsu Trees for our tests, we did not make a node correspondences to a PRED of an OBL as in Figure 8. Instead we included the PRED information in the link labels.
attributes and values in f-structures, such as TENSE and MOOD.

A Japanese linguist and Japanese native speaker created gold standard Pred Trees and Bunsetsu Trees for the 200 sentences consulting the outputs of the KNP, CaboCha and the LFG system.\(^8\) We compared the gold standard Bunsetsu Trees with the Bunsetsu Trees generated by KNP, CaboCha and the LFG system (Bunsetsu Comparison). We also compared the gold standard Pred Trees with the Pred Trees generated by the LFG system (Pred Comparison).

Table 3 shows the results of the Bunsetsu Comparison. The precision and recall are based on the number of correct Bunsetsu-Bunsetsu (B-B) dependencies. The precision and recall of “B-G-B” in the table are based on the number of the correct triplets of a bunsetsu, its parent Bunsetsu and a link label (a grammatical function) between them. The LFG system outputs all possible grammar analyses (as in 3). The Upper Bound is the average for 200 parses each of which is the best parse for a sentence. The Average is the macro average (that is, the average of the 200 averages for all parses for each sentence). The BASELINE is for the Bunsetsu Trees obtained assuming all Bunsetsus modify their right-hand neighbors.

Table 4 shows results of the Pred Comparison. The ‘P-G-P’ is based on the number of correct triplets of a PRED, its parent PRED and a link label between them. The ‘P-P’ is based on the number of the correct parent-child PRED pairs.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Results of the Bunsetsu comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prec. (%)</td>
<td>Rec. (%)</td>
</tr>
<tr>
<td>KNP (B-B)</td>
<td></td>
</tr>
<tr>
<td>77.2</td>
<td>78.9</td>
</tr>
<tr>
<td>CaboCha (B-B)</td>
<td></td>
</tr>
<tr>
<td>82.6</td>
<td>82.0</td>
</tr>
<tr>
<td>LFG (B-B) Upperbound</td>
<td></td>
</tr>
<tr>
<td>89.1</td>
<td>87.7</td>
</tr>
<tr>
<td>Average</td>
<td></td>
</tr>
<tr>
<td>81.3</td>
<td>79.5</td>
</tr>
<tr>
<td>LFG (B-G-B) Upperbound</td>
<td></td>
</tr>
<tr>
<td>86.9</td>
<td>85.4</td>
</tr>
<tr>
<td>Average</td>
<td></td>
</tr>
<tr>
<td>77.9</td>
<td>76.2</td>
</tr>
<tr>
<td>BASELINE (B-B)</td>
<td></td>
</tr>
<tr>
<td>58.6</td>
<td>56.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Results of the Pred comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prec. (%)</td>
<td>Rec. (%)</td>
</tr>
<tr>
<td>LFG (P-G-P) Upperbound</td>
<td></td>
</tr>
<tr>
<td>87.5</td>
<td>87.8</td>
</tr>
<tr>
<td>Average</td>
<td></td>
</tr>
<tr>
<td>79.7</td>
<td>80.4</td>
</tr>
<tr>
<td>LFG (P-P) Upperbound</td>
<td></td>
</tr>
<tr>
<td>90.0</td>
<td>90.3</td>
</tr>
<tr>
<td>Average</td>
<td></td>
</tr>
<tr>
<td>83.3</td>
<td>84.0</td>
</tr>
</tbody>
</table>

\(^8\) We did not refer to the syntactic annotations of the EDR corpus. Comparing our gold standard with these annotations is a future work.
5.2.3 Discussion

Table 3 shows that the Average “B-B” of the LFG system is roughly equivalent to that of KNP and CaboCha. The results of “B-G-B” are encouraging, because the “B-G-B” results are not substantially worse than the “B-B” results of the LFG system. This means the case detection by the LFG system works well.

We examined the Bunsetsu Trees to investigate differences among the three systems, and found the LFG system based on the linguistically fine-grained grammar advantageous in some regards. For example, KNP and CaboCha output parses in which a noun phrase ending with な modifies a verb that is not in a relative clause. This never happens with the LFG system as described in 4.2. As another example, the dependencies of noun phrases topicalized by は were more correctly analyzed by the LFG system than by the other systems, because of the precise grammar rules for case detection described in 4.2. On the other hand, rules for coordination in our grammar are not adequately sophisticated; therefore, the analyses for the sentences including coordination structures using the non-linguistic methods of KNP and CaboCha were better.

The results in Table 4 show that both precision and recall of the LFG system in the Pred comparison are higher than those in the Bunsetsu Comparison. This means the accuracy of the intra-Bunsetsu analyses by the LFG system is reliable. Examples of the intra-Bunsetsu analyses are shown in (10a–10c). For instance, the LFG system is required to determine the dependencies of the nouns in the compound noun (10c), that is, both せきしょく (red-color) and せつえん (insulation) modify わいや (wire).

The LFG system is based on a hand-coded grammar. We think we will be able to increase the accuracy of our system by continuing the development to address grammatical issues that have not yet been considered. In addition, by using a statistical method for disambiguation (such as the method proposed in Riezler et al. (2002)), it will be possible to select better parses from among the parses XLE generates for a sentence than random selection.

6 Conclusion

We have described a Japanese parsing system based on the LFG formalism. The system is the first Japanese LFG parser with over 97% coverage (91% on average without FRAGMENT analyses) for real-world text. We evaluated the accuracy of the system by comparing it with standard Japanese dependency parsers. The LFG parser shows a roughly equivalent performance on the Bunsetsu dependency accuracy to the standard Japanese parsers. It also provides reasonably accurate results for case detection and intra-Bunsetsu analyses.
We are incorporating the LFG system into off-line NLP applications. However, the processing speed of the system is not yet sufficient for real-time applications; therefore we will customize XLE for the Japanese grammar to achieve better performance.

Acknowledgement

We acknowledge our indebtedness to all the members of the ParGram project, especially Ronald Kaplan, Tracy Holloway King, John Maxwell, Mary Dalrymple and Yasunari Harada for their comments and discussions on early versions of the Japanese LFG grammar and system.

Reference


Hiroshi Masuichi: He is an executive research principal at Research & Technology Group of Fuji Xerox Co., Ltd. He received his Bachelor’s, Master’s degrees in engineering from Kyoto University and doctorate degree in engineering from Tokyo University of Agriculture and Technology in 1989, 1991,
and 2004, respectively. He is a member of the Japanese Society for Artificial Intelligence, the Information Processing Society of Japan, and the Association for Natural Language Processing, and an executive board member of the Japanese Society for Artificial Intelligence. His research interests include data mining, knowledge processing, and natural language processing.

**Tomoko Ohkuma**: She is a research principal at Research & Technology Group of Fuji Xerox Co., Ltd. Her main research interest is Information Extraction Technology based on Natural Language Processing (NLP). She is a member of organizers in NTCIR-11 MedNLP-2. She was a part-time lecturer of Tokyo Woman’s Christian University (in 4 years). She earned her Ph.D and M.A. from Keio University, where she majored NLP. She also earned B.A. in linguistics from Tokyo Woman’s Christian University.

(Received August 22, 2002)
(Revised December 2, 2002)
(Accepted January 10, 2003)