Japanese Parser on the basis of the Lexical-Functional Grammar Formalism and its Evaluation

Hiroshi MASUICHI
Corporate Research Group, Fuji Xerox Co., Ltd.
430 Sakai, Nakai-machi, Ashigarakami-gun,
Kanagawa, Japan
hiroshi.masuichi@fujixerox.co.jp

Tomoko OHKUMA
Corporate Research Group, Fuji Xerox Co., Ltd.
430 Sakai, Nakai-machi, Ashigarakami-gun,
Kanagawa, Japan
ohkuma.tomoko@fujixerox.co.jp

Hiroki YOSHIMURA
Corporate Research Group, Fuji Xerox Co., Ltd.
430 Sakai, Nakai-machi, Ashigarakami-gun,
Kanagawa, Japan
hiroki.yoshimura@fujixerox.co.jp

Yasunari HARADA
School of Law, Waseda University
1-6-1 Nishi-waseda, Shinjuku,
Tokyo, Japan
harada@waseda.jp

Abstract

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

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, dialogue understanding, message extraction, etc. In this paper we report development and performance of a parser for the Japanese language based on the Lexical-Functional Grammar (LFG) formalism (Kaplan and Bresnan, 1982; Dalrymple, 2001).

The Japanese LFG grammar used for this parser is being developed in relation to the Parallel Grammar (ParGram) project (Butt et al., 1999, 2002). In this project, grammars for six languages, English, French, German, Japanese, Norwegian and Urdu[1], 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.[2]

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 show its coverage and accuracy. For the accuracy evaluation, we compared outputs of the Japanese LFG parsing system with outputs of standard *bunsetsu*[3] dependency parsers for Japanese.

[1] Korean has recently been added to this list. The Japanese LFG grammar is used as the basis for a grammar of Korean (Kim et al., 2003).

[2] 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 which took advantage of the f-structure parallelism.

[3] A bunsetsu is a widely accepted unit of syntactic and phonological phrase structures in Japanese. It consists of one content word plus optionally following function words.

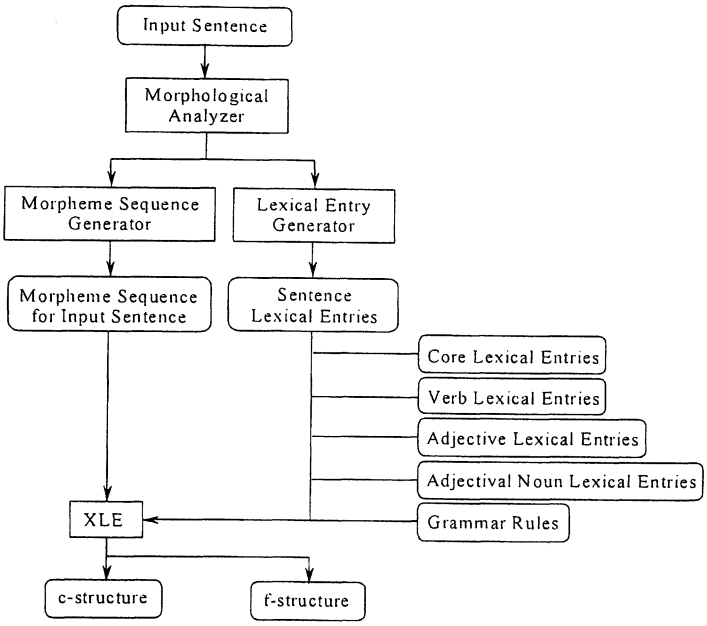

Figure 1: Diagram of the Japanese LFG system

This paper is organized as follows. Section 2 describes the architecture of the Japanese LFG system. In Section 3 we present the Japanese grammar written in the LFG formalism for the system described in Section 2. We show the coverage of the system in 4.1, and explain our approach to evaluating the accuracy of the system and report experimental results in 4.2.

2 Japanese LFG system

The ParGram project employs the XLE parser and grammar development platform (Maxwell and Kaplan, 1993). XLE produces packed representations, specifying all possible grammar analyses of the input. Japanese is the first Asian language that XLE was applied to (Masuichi and Ohkuma, 2003).

Figure 1 shows the diagram of the Japanese LFG system. An input Japanese sentence is segmented and tagged by the Chasen morphological analyzer (Matsumoto et al., 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, conjugation, etc. Unknown words are currently treated as nouns.

Verb Lexical Entries (VLEs), Adjective Lexical Entries (ALEs) and Adjectival Noun Lexical Entries (ANLEs) were written based on 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 is intended to recover from erroneous analyses by Chasen. For instance, it is impossible to distinguish correctly the Japanese case marker (postpositional particle) *de* from the conjugated form *de*[4] of the auxiliary verb *da* at the level of morphological analysis.

[4] This form behaves like a sentential conjunction.

Therefore we ignore the SLE created for *de* regardless of the output of Chasen. Instead we wrote the CLEs for *de*, which represent the behaviors both for the case marker *de* and for the conjugated form *de* of the auxiliary verb *da*. XLE parses the input keeping both possibilities and selects the one that matches the syntactic structure of the whole input.

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 described in Butt et al. (1999). We explain GRs in detail in Section 3. 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 et al., 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 for the constituents whose processing has not been completed. This mechanism enables the parser to avoid time-outs and memory-shortage problems and can improve the robustness of the system.

3 Japanese LFG grammar

3.1 Basic Rules

Some of the most noticeable syntactic characteristics of Japanese sentences actually used are relatively free word order, rampant pro-drop and extensive use of complex predication. To account for the free word order, we adopted the following (1) as the most basic rule of our grammar.

```
(1) S →       PP*        V
          (↑ GF)=↓   ↑ = ↓
```

For every sentence in Japanese, native speaker judgements seem to converge on an 'optimal' word order (Shibatani, 1990). For instance, (2a) is an intuitively strange Japanese sentence, while (2b) is felt to be more natural. It is possible to write a grammar that does not allow for the sentences like (2a). Otani et al. (2000) proposed a Japanese Head-driven Phrase Structure Grammar that is very sensitive to the word order. We, however, adopt (1) to achieve broader coverage because sentences like (2a) frequently appear in real-world text.

```
(2a) Singaporu  e   Zyon  ga    Tokyo  kara  itta.
     Singapore  to  John  SUBJ  Tokyo  from  went.
     John went from Tokyo to Singapore.
```

(2b) Zyon ga Tokyo kara Singaporu e itta.

We use the following type of lexical entry to handle pro-drop. (3) is the (simplified) entry for the verb *yomu* (read).

```
(3) yomu V ( ↑ PRED)='yomu<( ↑ SUBJ)( ↑ OBJ)>'
           @(PD SUBJ)
           @(PD OBJ).
   PD(GF)=@(DEFAULT ( ↑ GF PRED) 'pro'
                    ( ↑ GF PRON-TYPE) null)
           ProDrop: OT.
   DEFAULT(ATT1 VAL1 ATT2 VAL2)=
           ATT1=VAL1
           ATT2=VAL2.
```

'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 like (3) are important for achieving broad coverage. (Frank et al., 2001)

We adopted relatively loosely constrained mono-clausal analyses for verbs to deal with various kinds of complex predications. However, we employed multi-clausal analyses when it is reasonable to consider that a verb includes multiple predicate-argument relations (PARs). This treatment is essential for such NLP applications as question answering, dialog understanding, machine translation, etc. We will describe the multi-clausal analyses in 3.2.

3.2 Predicate-Argument Relations and Cases

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

(4a) Zyon ga yonda hon
John SUBJ read book
(4b) Zyon no yonda hon
SUBJ
The book John read
(4c) Zyon no hon o yonda.
GEN OBJ
(Someone) read John's book.
(4d) Zyon ga hon o yonda.
(4e) Zyon wa hon o yonda.
TOPIC/SUBJ
John read the book.
(4f) Zyon wa yonda hon o nagesuteta.
TOPIC/SUBJ threw-away.
John threw away the book (someone) read.

For instance, *no* can be used as a SUBJ marker as in (4b) instead of *ga* as in (4a). However *no* in (4c) cannot be interpreted as a SUBJ marker. *Wa* in (4e) causes the omission of a SUBJ marker *ga* in (4d). On the other hand, *wa* in (4f) also causes the omission of a SUBJ marker but 'John' is the SUBJ of 'threw away' and the SUBJ of 'read' is dropped, that is, '*Zyon wa*' modifies *nagesuteta* rather than *yonda*. The generalized rules for these grammatical phenomena are: (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 widely been discussed, there is no Japanese syntactic parser that has formal rules to treat (I) and (II). We can write the following simple rule (5) based on the LFG formalism, which represents (I) and (II). 'PPsubj-no' in (5) represents a postpositional phrase with *no*, and 'PPsubj' represents a postpositional phrase with a SUBJ marker other than *no*.

(5) Srel → {PPsubj | PPsubj-no} PP* V
(↑ SUBJ)= ↓ (↑ GF)= ↓ ↑ = ↓
(↓ TOPICALIZATION-FORM)≠'wa'

Another type of construction that relates to PAR capturing and case detection is a verb (a bunsetsu) which includes multiple PARs.

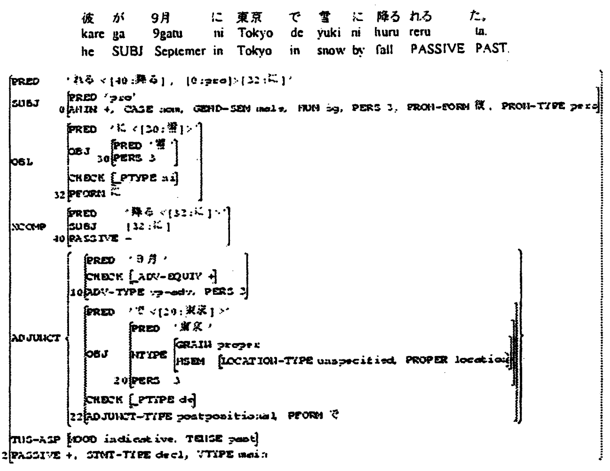
彼 が 9月 に 東京 で 雪 に 降る れる た。
kare ga 9gatu ni Tokyo de yuki ni huru reru ta.
he SUBJ Septemer in Tokyo in snow by fall PASSIVE PAST.
PRED
SUBJ
OBL
XCOMP
ADJUNCT
TNS-ASP
PASSIVE

Figure 2: F-structure for sentence (6)

(6) kare ga 9gatu ni Tokyo de yuki ni hurareta.
he SUBJ Septemer in Tokyo in snow by fell-PASSIVE.
*He was fallen by snow in Tokyo in September.

(7) zyoo ga Sirayukihime ni ringo o tabesaseta.
queen SUBJ Snow-White by apple OBJ ate-CAUSATIVE.
The queen made Snow White eat an apple. (The queen tempted Snow White to eat an apple.)

Sentence (6) is an example of the indirect passive (Masuoka et al., 1997). In this case, the intransitive verb *huru* (fall) is passivized. Sentence (7) is an example of a causative sentence. We adopted multi-clausal analyses for the indirect passives and the causative sentences, by regarding auxiliary verbs (fragments of bunsetsus) that cause passives (e.g. *reru* and *rareru*) and causatives (e.g. *seru*, *saseru*, *(te-)morau* and *(te-)itadaku*) as having PREDs.[5] These analyses enable us to capture the PARs '*huru-yuki*(SUBJ)' in (6) and '*taberu-Sirayukihime*(SUBJ)-*ringo*(OBJ)' in (7) as well as the main PARs '*reru-kare*(SUBJ)-*yuki*(OBL)-*huru*(XCOMP)' in (6) and '*saseru-zyoo*(SUBJ)-*Sirayukihime*(OBL)-*taberu*(XCOMP)' in (7). Figure 2 shows the f-structure generated by the LFG system for (6). The f-structure for (7) can be seen in Fig. 5. 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 the unit of analysis.

-A case marker shift caused by potential verbs
-A case marker shift caused by *te-aru* sentences
-Behavior of particular nouns that a verb in a relative clause cannot subcategorize for
-Case disambiguation in causative-and-passive sentences
-Case disambiguation in coordinated sentences with topic particles
-Case disambiguation in relative and embedded clauses with topic particles
-Case marker shifts caused by relative and embedded clauses

Figure 3: Examples of Japanese grammatical issues related to PAR and case detection

[5] 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).

私	は	彼	が	書く	た	本	を	ずっと	読む	でいる	ます	ちなみに。
watasi	wa	kare	ga	kaku	ta	hon	o	zutto	yomu	de-iru	masu	tinamini.
I	TOPIC/SUBJ	he	SUBJ	write	PAST	book	OBJ	always	read	PROG	POLITE	F.Y.I.

I am always reading the böok he wrote, F.Y.I.

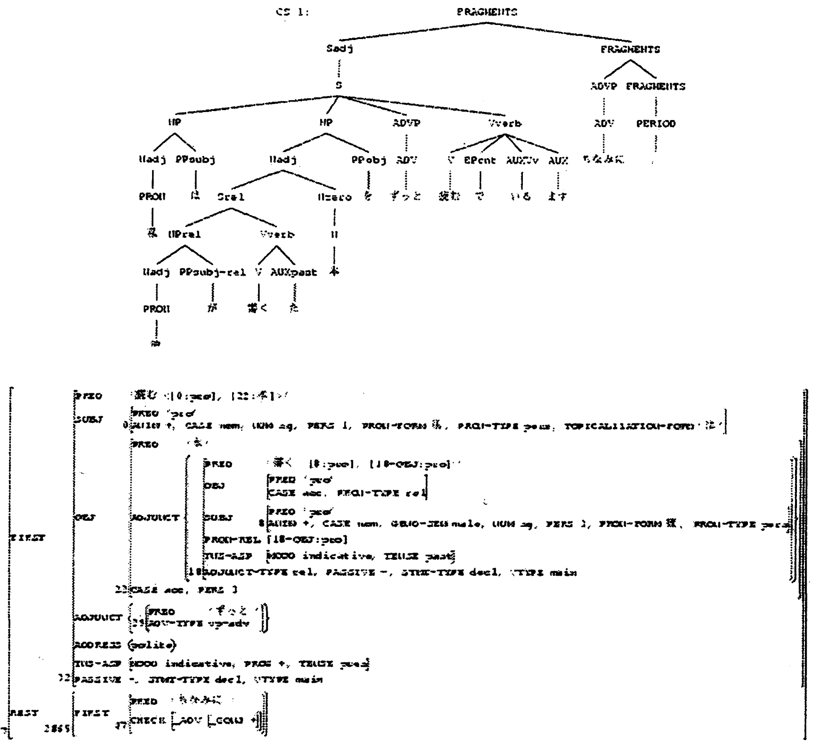

Figure 4: Examples of Fragment c-/f-structures

There are several grammatical issues concerning PAR capturing and case detection. Figure 3 shows the list of examples of the grammatical issues for which we have already written LFG rules.

3.3 Robustness Techniques

As mentioned in 3.1, we use Optimality Theory (OT) marks to delete dispreferred parses. We also use OT marks for efficiency. When OT marks are divided into groups that are ranked according to their preference, XLE processes the input in multiple passes. The core grammar that consists of the rules with OT marks in the highest preference group is used for the first pass. If a valid parse is found, then XLE will stop. Otherwise, XLE will process the input again with the core grammar plus the rules with OT marks in the second highest preference group for the second pass. This multiple-pass-parsing mechanism is useful for writing rules for rare grammatical phenomena without increasing unintended ambiguity.

(8a) aruku koto ga itibanda.
walk NOMINALIZER SUBJ is-best.
To walk is best.
(8b) *aruku ga itibanda.
walk SUBJ is-best.

(8c) makeru ga katida.
lose SUBJ is-victory.
To lose is a victory.

For instance, the SUBJ marker *ga* ordinarily does not follow the canonical form of verbs as shown in (8a) and (8b). However, it exceptionally occurs in some idiomatic expressions as in (8c). We divided the 42 OT marks in our grammar into four groups, and we put the OT mark for the rule for the sentences like (8c) in the lowest preference group. The OT marks in the lowest preference group are added to the rules for rare idiomatic expressions or 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) 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 Fig. 4. In this case, the adverb *tinamini* (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 the left of it.[6] Figure 4 shows that the whole sentence except *tinamini* is correctly analyzed.

4 Experimental Evaluation

4.1 Coverage

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

(A) 10,000 sentences in the Japanese EDR corpus (EDR, 1996), mainly composed of newspaper text,
(B) 460 sentences in 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 of (A)-(C) consist of randomly selected unseen sentences. Most sentences in (A) and (B) are grammatical. On the other hand, (C) includes a lot of ungrammatical and colloquial sentences, because the sentences in (C) are the transcriptions of telephone calls from customers to a customer service center.

Table 1 lists the coverage results. The Total coverage for (C) is almost the same as that of (A) and (B). On the other hand, the STANDARD coverage for (C) is lower than (A) and (B). This is because (C) includes a lot of ungrammatical sentences, and the ratio of the sentences that could be covered by the STANDARD grammar was low; the fragment grammar covers the ungrammatical sentences. The main

Table 1: Results of the coverage tests

	Num. of sentences	Ave. num. of words in a sentence	Coverage (%)					Ave. time. to analyze a sentence without Skimming* (sec)	Ave. num. of STANDARD results
			Total	STANDARD	FRAGMENT	Skimmed STANDARD	Skimmed FRAGMENT		
(A)	10,000	22.6	97.3	91.6	5.7	4.7	1.2	1.9	14.6
(B)	460	21.3	98.7	94.4	4.3	2.0	0.7	1.1	10.1
(C)	9,637	16.3	97.9	87.0	10.9	1.9	0.9	4.8	46.2

*2.8GHz CPU
2GB memory

reason for failure of analysis for (A)-(C) was time-outs caused by the sentences that even the skimming mode could not cover. The table shows that the system has over 97% coverage in total and 87% coverage with the STANDARD grammar even for (C).

4.2 Accuracy

4.2.1 Approach

Outputs of standard Japanese parsers such as KNP (Kurohashi and Nagao, 1994) and Cabocha (Kudo and Matsumoto, 2002) are trees that express bunsetsu dependencies (Bunsetsu Tree). 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 change 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, (9a) is one bunsetsu, but the f-structure for (9a) includes two PREDs as described in 3.2. (9b) is 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 (9c), but its f-structure can include multiple PREDs corresponding to the nouns that compose the compound noun.

(9a) kakaseru (one bunsetsu)
kaku (write) seru (make) (two morphemes)
(9b) kakanai (one bunsetsu)
kaku (write) nai (not) (two morphemes)
(9c) sekisyokuzetuenwaiya (one bunsetsu)
sekisyoku zetuen waiya (three nouns)
(red-color) (insulation) (wire)

Another problem is that a bunsetsu is a language-dependent and phrase structural constituent. As described in Section 1, 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 the information concerning bunsetsus should appear in a c-structure and not in an f-structure. 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, and these correspond to bunsetsus. We created a Bunsetsu Tree from a c-structure and an f-structure as follows:

1. List the grammatical categories that correspond to bunsetsus.
2. Specify the nodes in a c-structure, which 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 regarding '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 5 shows the c-/f-structures and the Pred/Bunsetsu Trees for sentence (7). 'NP', 'NPobl' and 'Vverb' in the c-structure are the grammatical categories that correspond to bunsetsus. The numbers in Fig. 5 indicate the correspondence of the grammatical categories in the c-structure to the AVMs in the f-structure. We can specify the Bunsetus AVM by referring to the correspondence and create the Bunsetsu Tree.[7] The Bunsetsu Trees are comparable to the trees generated by the standard bunsetsu dependency parsers for Japanese.

[6] It is one of the principles of Japanese syntax that a bunsetsu should modify another bunsetsu to the right of it.

[7] When we created the Pred Trees and the Bunsetsu Trees for our tests, we did not make a node that corresponds to a PRED of an OBL as in Fig. 5. Instead we included the PRED information in the link labels.

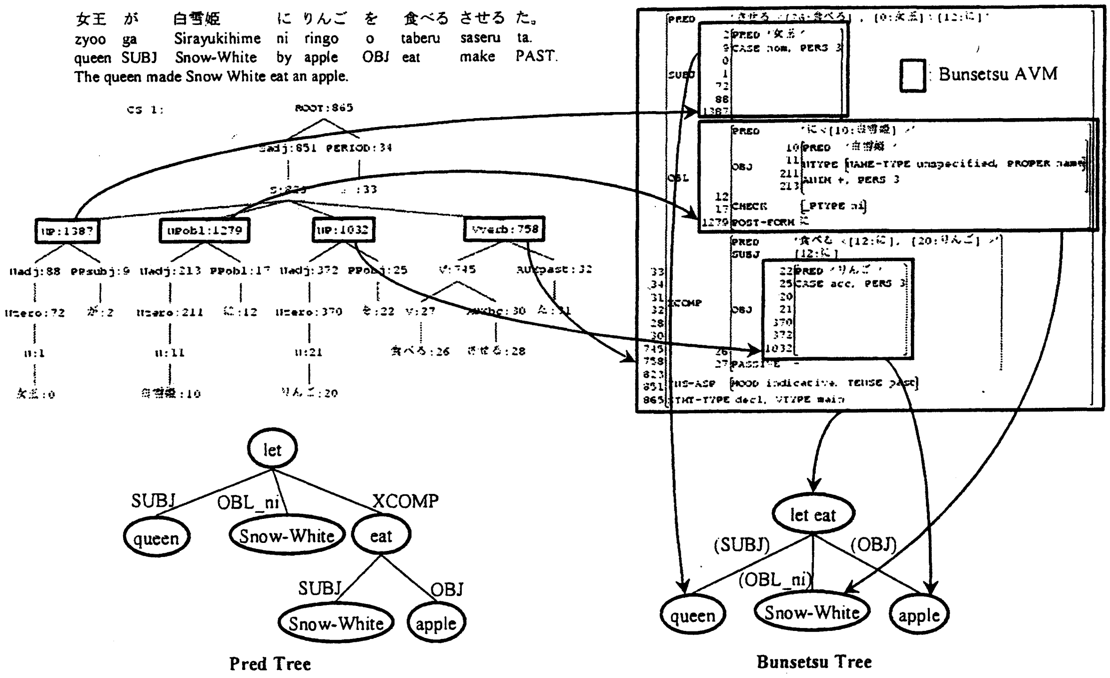

Figure 5: C-/f-structure and Pred/Bunsetsu Trees for sentence (7)

4.2.2 Experiments and Results

We randomly selected 200 unseen sentences from the EDR corpus for which the Japanese LFG system returns 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 Bunsetus Trees were created from the outputs of the LFG system. In this experiment, the link labels in the trees are grammatical functions as in Fig. 5, and we omitted other attributes and values in f-structures such as TENSE and MOOD.

A Japanese linguist and a Japanese native speaker created gold standard Pred Trees and Bunsetsu Trees for the 200 sentences consulting the outputs of 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 2 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 Section 2). The Upper Bound is the average for 200 parses each of which is the best parse for a sentence. The Upper Bound is the ideal value we can obtain if the system could always select the best parse for each sentence. The Average 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.

[8] We did not refer to the syntactic annotations the EDR corpus has. Comparing our gold standard with these annotations is future work.

Table 2: Results of the Bunsetsu Comparison

		Prec. (%)	Rec. (%)
KNP (B-B)		77.2	78.9
Cabocha (B-B)		82.6	82.0
LFG (B-B)	Upper Bound	89.1	87.7
	Average	81.3	79.5
LFG (B-G-B)	Upper Bound	86.9	85.4
	Average	77.9	76.2
BASELINE (B-B)		58.6	56.0

Table 3: Result of the Pred Comparison

		Prec. (%)	Rec. (%)
LFG (P-G-P)	Upper Bound	87.5	87.8
	Average	79.7	80.4
LFG (P-P)	Upper Bound	90.0	90.3
	Average	83.3	84.0

Table 3 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.

4.2.3 Discussion

Table 2 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 check the difference among the three systems, and found the LFG system based on the linguistically fine-grained grammar advantageous in some points. For example, KNP and Cabocha output the parses in which a noun phrase which ends with *no* modifies a verb that is not in a relative clause. This never happens with the LFG system as described in 3.2. As another example, the dependencies of noun phrases topicalized by *wa* were more correctly analyzed by the LFG system than the other two because of the precise grammar rules for the case detection described in 3.2. On the other hand, rules for coordination in our grammar are not sophisticated enough, so the analyses for the sentences including coordination structures on the basis of the non-linguistic methods of KNP and Cabocha were better.

The results in Table 3 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 bunsetsu-internal analyses by the LFG system is reliable. Examples of the bunsetsu-internal analyses are the analyses for (9a)-(9c). For instance, the LFG system is required to determine the dependencies of the nouns in the compound noun (9c), that is, both *sekisyoku* (red-color) and *zetuen* (insulation) modify *waiya* (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 grammar development for the grammatical issues that have not yet been considered. In addition, by using a statistical method for disambiguation (e.g. 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.

5 Conclusion

We have reported 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 compared with the standard Japanese dependency parsers. The LFG parser shows roughly equivalent performance on the bunsetsu dependency accuracy to the standard Japanese parsers. It also provides reasonably accurate results for case detection and bunsetsu-internal analyses.

We are incorporating the LFG system in some off-line NLP applications. However the time performance of the system is not enough for real-time applications, so we will customize XLE for the Japanese grammar to achieve better performance.

Acknowledgements

We acknowledge our indebtedness to all the members of the ParGram project, especially Ronald Kaplan, Tracy Holloway King and John Maxwell at Palo Alto Research Center, and Mary Dalrymple at King's College London for their comments and discussions on early versions of the Japanese LFG grammar and system. We would also like to thank Tracy Holloway King at PARC, along with Martin van den Berg and Lorenzo Thione at FX Palo Alto Laboratory for their helpful comments on an earlier draft of this paper.

References

Joan Bresnan. 2000. Optimal Syntax. In *Optimality Theory: Phonology, Syntax and Acquisition*, (Joost Dekkers, Frank van der Leeuw, and Jeroen van de Weijer, eds.) pages 334-385. Oxford University Press, Oxford.

Miriam Butt, Helge Dyvik, Tracy H. King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02) Workshop on Grammar Engineering and Evaluation*, pages 1-7.

Miriam Butt, Tracy H. King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook.* Number 95 in CSLI Lecture Notes. CSLI Publications, Stanford, California.

Mary Dalrymple. 2001. *Lexical-Functional Grammar.* Academic Press, New York. Syntax and Semantics, volume 34.

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

Anette Frank, Tracy H. King, Jonas Kuhn, and John T. Maxwell III. 2001. Optimality Theory Style Constraint Ranking in Large-scale LFG grammars. pages 367-397. In *Formal and Empirical Issues in Optimality Theoretic Syntax,* Peter Sells, ed. CSLI Publication, Stanford, California.

Anette Frank. 1999. From Parallel Grammar Development towards Machine Translation. In *Proceedings of MT Summit VII. "MT in the Great Translation Era"*, Singapore, pages 134-142.

Fuji Xerox Co., Ltd. 2000. *Document Gate operation manual first edition*, DE-1006. (in Japanese)

IPA (Information-technology Promotion Agency). 1987. *IPA Lexicon of the Japanese Language for Computers IPAL.* (in Japanese)

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations* (Joan Bresnan, ed.), pages 173-281. The MIT Press, Cambridge, Massachusetts.

Roger Kim, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, Hiroshi Masuich, and Tomoko Ohkuma. 2003. Multilingual Grammar Development via Grammar Porting, In *Proceedings of the 2003 European Summer*

School in Logic, Language, and Information (ESSLLI2003) workshop "Ideas and Strategies for Multilingual Grammar Development". To appear.

Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *Proceedings of Sixth Conference on Natural Language Learning (CoNLL '02)*, Taipei, Taiwan, pages 63-69.

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.

Hiroshi Masuichi and Tomoko Ohkuma. 2003. Constructing a practical Japanese parser based on Lexical-Functioncal Grammar. *Journal of Natural Language Processing*, 10(2):79-109. (in Japanese)

Takashi Masuoka, Yoshio Nitta, Takao Gunji, and Satoshi Kinsui. 1997. *Grammar*. Iwanami Press, Tokyo, Japan. (in Japanese)

Yo Matsumoto. 1996. *Complex Predicates in Japanese: A Syntactic and Semantic Study of the Notion 'Word'*. CSLI Publications, Stanford, California and Kurosio Publishers, Tokyo. Studies in Japanese Linguistics volume 7.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. 1999. *Japanese Morphological Analysis System ChaSen version 2.0 Manual 2nd edition*. Technical report, Nara Institute of Science and Technology, Japan. (in Japanese)

John T. Maxwell III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571-589.

Akira Ohtani, Takashi Miyata, and Yuji Matsumoto. 2000. On HPSG-Based Japanese Grammar -Refinement and Extension for Implementation-. *Journal of Natural Language Processing*, 7(5):19-49. (in Japanese)

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA, Pages 271-278.

Peter Sells. 1985. Lectures on Contemporary Syntactic Theories: *An Introduction to Government-Binding Theory, Generalized-Phrase Structure Grammar, and Lexical-Functional Grammar*. CSLI Publications, Stanford, California.

Masayoshi Shibatani. 1990. *The languages of Japan*. Cambridge University Press, Cambridge, United Kingdom.

Kenji Yokota. 2001. Complex-predicate Formation and Some Consequences in Japanese. In *On-line Proceedings of the LFG01 Conference* (Miriam Butt and Tracy H. King eds.). URL: http://csli-publications.stanford.edu/.