An Extension of LangLAB for Japanese Morphological Analysis

Tomoyosi Akiba, Takenobu Tokunaga, Hozumi Tanaka
Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ôokayama Meguro Tokyo 152 Japan
{akiba,take,tanaka}@cs.titech.ac.jp

Abstract

LangLAB is a natural language analysing system which adopts a bottom-up, depth-first parsing strategy. LangLAB has useful features to analyse English text. This paper describes an extension of the LangLAB system in order to analyse Japanese. In particular, a morphological analysis program of Japanese is proposed. The representation form of grammar rules (DCG) and dictionary description (DRF) are also extended to handle the morphological information.

1 Introduction

LangLAB [5] provides the users a facility to build a natural language analysis system. The users only have to prepare grammar and a dictionary. Some of the significant features of LangLAB are listed below.

- Bottom-up depth-first parsing strategy is adopted. This allows left-recursive grammar rules.
- The grammar description form XGS provides a device to describe gapping phenomena which typically appear in relative clauses of English.
- A TRIE-structured dictionary is adopted to enable efficient and flexible reference to the dictionary. Easy way of handling idioms is also possible.
- Morphological analysis and syntactic analysis are performed at the same time. Semantic analysis can also be performed by using augmentation of DCG in which we can write any Prolog programs.
- LangLAB runs on any Prolog which has the same syntax as that of DEC-10 Prolog. The current version runs on C-Prolog, Quintus-Prolog and SICStus-Prolog.

This paper describes an extension of LangLAB system in order to analyse Japanese. In particular, a morphological analysis program of Japanese is proposed. Morphological analysis of Japanese is very different from that of English, because no spaces are placed between words. The analysis includes segmentation of words and handling inflections.

The overview of the extended system is shown in figure 1. LangLAB consists of two components: the BUP-XG system and the dictionary reference program. The BUP-XG system is a syntactic parser, which uses the Prolog program derived from the grammar rules. The BUP-XG

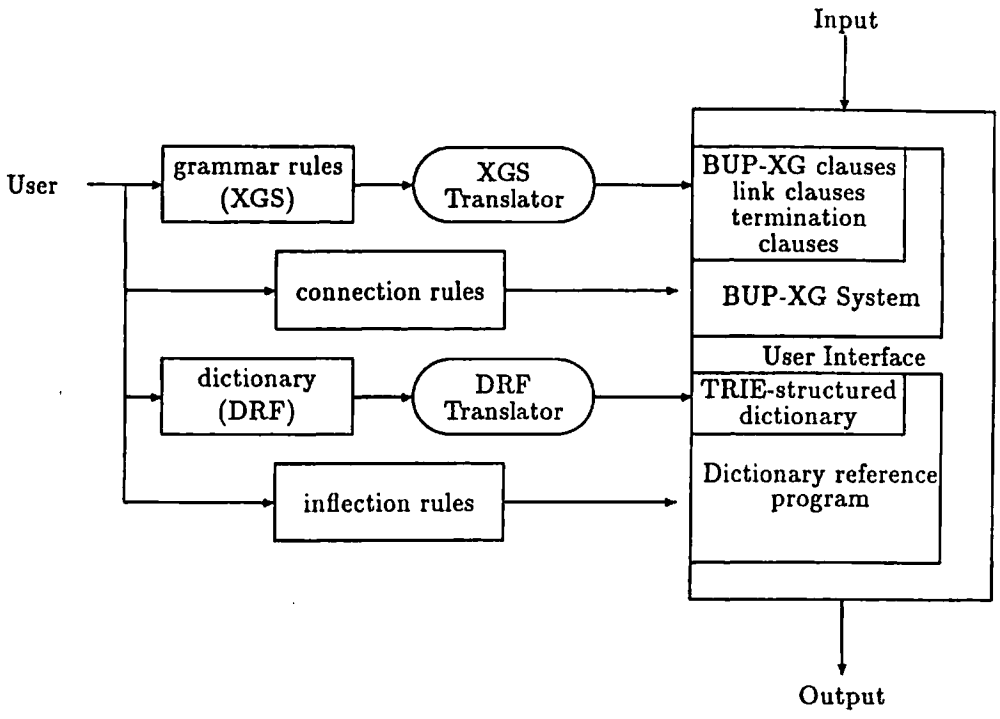

Figure 1: Overview of LangLAB (extended)

system also refer to the connection rules that check the connectability between adjacent words. The dictionary reference program consults the TRIE-structured dictionary which is translated from the dictionary the user prepares, and handles the inflection of morpheme using the inflection rules. Sample connection rules and inflection rules come with the distribution package.

2 Morphological analysis in LangLAB

Morphological analysis of Japanese is very different from that of English, because no spaces are placed between words. This is also the case in many Asian languages such as Korean, Chinese, Thai and so forth. Processing such languages requires the identification of the word boundaries in the first place. This process is often called *segmentation*. Segmentation is one of the most important tasks of morphological analysis for these languages, since the wrong segmentation causes fatal errors in the later stages such as syntactic, semantic and contextual analysis. However, correct segmentation is not always possible only with morphological information. Syntactic, semantic and contextual information may help resolve the ambiguities in segmentation. So, it is preferable to integrate the morphological and syntactic analysis.

In the LangLAB system, the morphological analysis and the syntactic analysis are performed at the same time. The semantic analysis can also be performed by using augmentation of DCG in which we can write any Prolog programs.

It is known that constraints on connectability between adjacent words can be represented

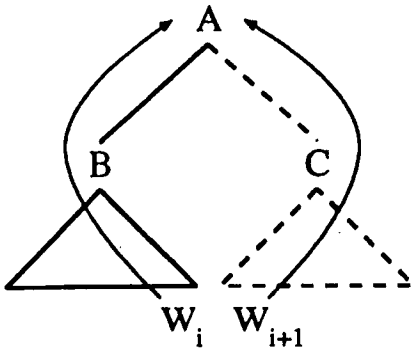

Figure 2: Connectability check by CFG

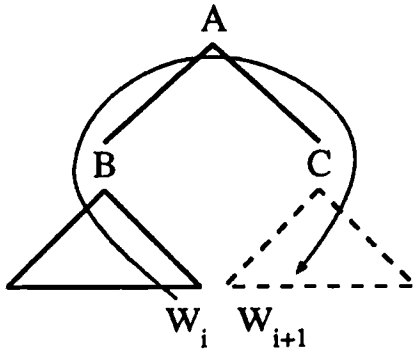

Figure 3: Connectability check by LangLAB (Top-down prediction)

by regular grammars. On the other hand, the syntactic constraints are often represented by context free grammars (CFGs). These constraints are usually kept separately in order to make maintenance and extension of the constraints easier. From a viewpoint of processing, however, it is preferable to integrate the morphological and syntactic analysis into a single framework.

One possible approach is to integrate the two constraints into a single constraint representation such as CFG. One of the drawbacks of this approach is the delay of checking the connectability between adjacent words in segmentation [4]. For example, in figure 2, in order to check the connectability between adjacent words, w_i and w_{i+1}, the morphological attributes of each word should be propagated up to their mother nodes B and C, and the check is delayed until the application of the rule A $\rightarrow$ B C. In the LangLAB system, this connectability check is done through top-down category prediction of the left-corner parsing algorithm. Thus we can avoid the delay of checking connectabilities. In figure 3, the morphological attribute of word w_i propagates top-down from nodes C and the connectability can be checked as soon as the word w_{i+1} is consulted in the dictionary.

3 What the users need for analysis

In order to use LangLAB, the users have to prepare four resources:

- grammar rules
- a dictionary
- inflection rules
- connection rules between adjacent words

The inflection rules and the connection rules are already prepared in the distribution package. They are extracted from JUMAN[2], the Japanese morphological analyser developed at Kyoto university and Nara Advanced Institute of Science and Technology[1]. They can be replaced by the users if necessary.

[1]This morphological information originally was developed at ICOT

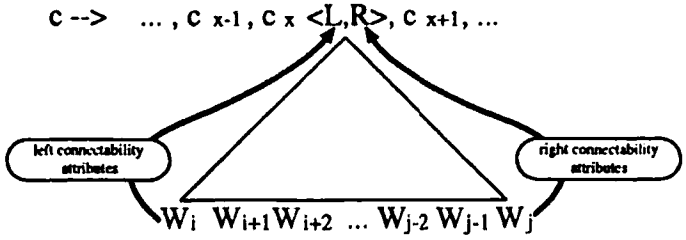

Figure 4: Connectability attributes used by a grammar rule

3.1 Grammar Rules

In LangLAB, a dictionary and grammar rules (of the XGS format) are prepared in separate files. This makes maintenance of the dictionary and the grammar rules easier.

XGS has been developed by extending DCG so that slash (/) category can be handled, and basically, XGS maintains upper-compatibility with DCG. However, it seems that slash category is not necessary for Japanese analysis. The users can develop the grammar in the DCG format.

The users can write the grammar in the DCG format as follows:

```
np(NP) --> vp(VP),np(NPO),{ program(VP,NPO,NP) }.
```

where NP,VP,NPO are arguments (logical variables) that carries the information of each category, and the enclosed by the braces is the augmentation in which we can write any Prolog program that is executed when this rule is used.

In section 2, we have explained that the morphological attributes are passed through the grammar rules and used in the top-down prediction (see figure 3). The morphological attributes are usually invisible to the users. But, sometimes the users might want to manipulate these attributes explicitly in the grammar description. The morphological attributes can be manipulated through the extended notation. For example, we might want to constrain the verb that modifies a noun must have the noun modification form of morpheme. This constraint can be written as follows:

```
np(NP) --> vp(VP)<M>,{ noun_modification_form(M) },
           np(NPO),{ program(VP,NPO,NP) }.
```

where the enclosed by the angle brackets is the extra argument which denote the morphological attribute. In this example, 'noun_modification_form/1' checks if vp has noun modification form.

In general, the users would be able to write one or two argument within the angle brackets. A single argument in the angle brackets, as the above example, denotes the right connectability attribute of the right-most morpheme of the category. Two arguments in the angle brackets (e.g. "vp(VP)<L,R>"), denote the left connectability attribute of the left-most morpheme (L) and the right connectability attribute of the right-most morpheme (R) of the category (see figure 4).

3.2 Dictionary

The users describe a dictionary with DRF (Dictionary Representation Form), an extended version of the DCG format. The dictionary written in DRF is translated into a TRIE-structured dictionary by the DRF translator.

For the morphological analysis, the users have to describe the morphological information at the dictionary, in addition to the syntactic information that is written in original DCG form. The morphological information is described with the extended form of DCG. For example, the users can write a dictionary entry as follows:

```
verb(iku)<[iku,ka5,_]> --> "行".
```

where the enclosed by the angle brackets is the morphological information. We refer to the arguments within the angle brackets as morphological arguments hereafter.

The morphological information is used by both the inflection processing and the connectability check between adjacent words. For the inflection processing the information about the inflection pattern of the word is necessary, while for the connectability check the information about connectability attributes is necessary. The inflection processing program, which will be described in the next subsection, determines how to use the morphological information.

The connectability attributes written in the morphological arguments are passed to the grammar rules and used as the top-down prediction of connectability. On the other hand the information written in the original argument of DCG are passed to the grammar rules in the bottom-up manner.

3.3 Inflection rules

A TRIE-structured dictionary is consulted by the dictionary reference program. In addition to dictionary reference, this program calls the program that analyses the inflection of Japanese morpheme. The users only have to write the root form of lexical items. Morphological inflection is automatically handled by the program.

The users need to define the inflection processing program, which is the predicate `morph/10`. By defining this predicate, the users can control how to use the morphological information. Some users might want to prevent the system from the morphological processing, and some users might want only the connectability check, while other users might want both the connectability check and inflection processing.

For example, the predicate can be defined as follows:

```
morph(_,A,[M],X,Y,A,_,M,M,Z) :-
        inflection_rule(M,Y,Z).

inflection_rule([_,ka5,mizen],[か|Z],Z).
inflection_rule([_,ka5,renyou],[き|Z],Z).
...
inflection_rule([_,sa5,mizen],[さ|Z],Z).
...
```

In this example, the variable `M` is the morphological information described as the morphological argument in the dictionary. The differential list `X-Y` is the string of the root form of a lexical item. This program gives the inflected form of the lexical item that is represented by the differential list `X-Z`. The predicate '`inflection_rule/3`' is used for seeking the inflected postfix substring. In addition, the variable `A` is the original argument of the category; the users could also manipulate the argument in this program.

3.4 Connection rules

Connection rules are constraints on connectability between adjacent words. Each word has a connectability attribute and the rules define what two attributes can be connected.

For the connection check between adjacent words, the system calls the predicate `connect/2`. The two arguments of the predicate are the pair of the right connectability attribute of the left word and the left connectability attribute of the right word that are connectable. For example, the connectable pairs of attributes are enumerated as follows:

```
connect([_,ka5,mizen],[nai,_,_]).
connect([_,ka5,mizen],[seru,_,_]).
...
```

The users who do not need the connectability check would define the predicate as follows.

```
connect(_,_).
```

4 Summary

An extension of the LangLAB system for analysing Japanese has been described. The system performs the Japanese morphological analysis including the segmentation, the inflection processing and the connectability check between adjacent words. The connectability check is done using the top-down prediction mechanism of left-corner parsing which prevent us from the delay of checking. We also extended the notation of grammar and a dictionary for handling the morphological information.

Appendix: Information of the resources

Name of the resource: LangLAB

Brief summary of resource: LangLAB is Natural language analysis system with extensions for Japanese.

Availability of the documents/manuals: The manuals for original system are available both in English and in Japanese. The manual for Japanese extension is available only in Japanese.

Price: free

Limitation: no limitation

FTP site: `nlp-server.cs.titech.ac.jp:/pub/LangLAB.tar.gz`

Media: FTP

Format: Unix tar and gzip compress file

Style of demonstration: on-line

Contact person: email:`nlp@cs.titech.ac.jp`

Implementation language: yacc (for translator of grammar and dictionary), Prolog (for runtime system)

Size: 2MB (compressed)

References

[1] Matumoto, Y., et al. BUP: A Bottom-Up Parser Embedded in Prolog, New Generation Computing, Vol.1, No.2, pp.145-158, 1983.

[2] Matsumoto, Y. and others, JUMAN Users Manual, Kyoto University and Nara Institute of Science and Technology, 1993

[3] Pereira, F.C.N. and Waren, D.H.D. Definite Clause Grammars for Language Analysis: A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artificial Intelligence, Vol.13, pp.231-278, 1980.

[4] Tanaka Hozumi, Tokunaga Takenobu and Aizawa Michio. Integration of Morphological and Syntactic Analysis Based on LR Parsing Algorithm. In proceeding of Third International workshop on Parsing Technologies, 1993.

[5] Tokunaga, T. and Iwayama, M. and Kamiwaki, T. and Tanaka, H. LangLAB: A Natural Language Analysis System, In proceedings of COLING, 1988.