Development System for Feature Structure-Based Unification Grammar

Takashi Miyata and Yuji Matsumoto
Graduate School of Information Science,
Nara Institute of Science and Technology
Email: {takashi,matsu}@is.aist-nara.ac.jp
Tel: +81–743–72–5245, Fax: +81–743–72–5249

1 System Features

Our grammar development system has following features:

- Graphical user interface (GUI) for browsing parsing ambiguities, parse tree structures, selective application of schema and principles, and manipulation on feature structures
- Annotation-aware parsing
- Modular structure, which enhances the system's portability

The first feature almost totally hides the parser implementation. Although the current version does not support editing the grammar through GUI, it allows users to browse parses of a given sentence on the level of linguistic abstraction. This feature also enables our system for beginners to learn about feature structure-based unification grammar formalisms.

The second feature means that our syntactic parser accepts bracketed and/or tagged sentences as input. These annotations reduce ambiguities of a long sentence without modifying the currently developed grammar. This feature can be regarded as an interface with preprocessor such as chunking and dependency analysis, but there is more essential merit to annotate input sentences. The reason why such feature is required is explained in detail in the next section.

The last feature concerns implementation and design of our system. Our system consists of four modules: morphological analysis system ChaSen (Matsumoto et al., 1999), syntactic parser SAX (Matsumoto et al., 1993), unification engine ProFIT (Erbach, 1995), and GUI module. The GUI module is subdivided into three parts. The modules are mutually independent and can be replaced with another implementation as far as the interface protocol and the format of the grammar are preserved. Especially the feature structure editor can be used as a stand alone system. Our system is organized as illustrated in Figure 1.

2 Background

We have studied and developed HPSG (Pollard and Sag, 1994) like grammar. Our research goal is to unify semantics in such feature structure unification-based grammar with lexical conceptual structure (LCS) (Jackendoff, 1990) and generative lexicon (GL) (Pustejovsky, 1996). Since LCS and GL require quite flexible symbolic computation, manipulation of feature structures in these framework should be supported by sophisticated user interface.

Another goal is to develop a framework for integration of various linguistic knowledge, especially statistic one. Statistical approaches have been recently applied in research on various linguistic phenomena, such as dependency analysis among phrases and anaphora resolution. These statistical approaches have achieved to obtain large scale heuristics on particular linguistic phenomena, but few researches have investigated interaction of these heuristics across linguistic phenomena. One of the difficulty of such investigation is that there is no framework in which the heuristics obtained from each research can be uniformly described. We try to develop such framework based on feature structure-based unification grammar like HPSG. From the viewpoint of feature structure-based unification grammar, each particular heuristics can be seen as constraints among the specific set of features. These sets of features must have non empty intersection, especially syntactic features. We want to find out minimum set of features that commonly used in various heuristics and study how particular linguistic phenomena can be explained by extending such minimum feature set.

Our approach is a bottom up one like those sorts of previous researches, but we aim to construct a grammar which has wide coverage. Generally speaking, grammar development requires enormous efforts, especially in early stage. In such situation, tackling problems one by one makes the grammar vague and extension of the grammar apts to become difficult because of ad hoc solutions. So we have decided that when a given sentence cannot be analyzed by the current grammar, we actively annotate the sen-

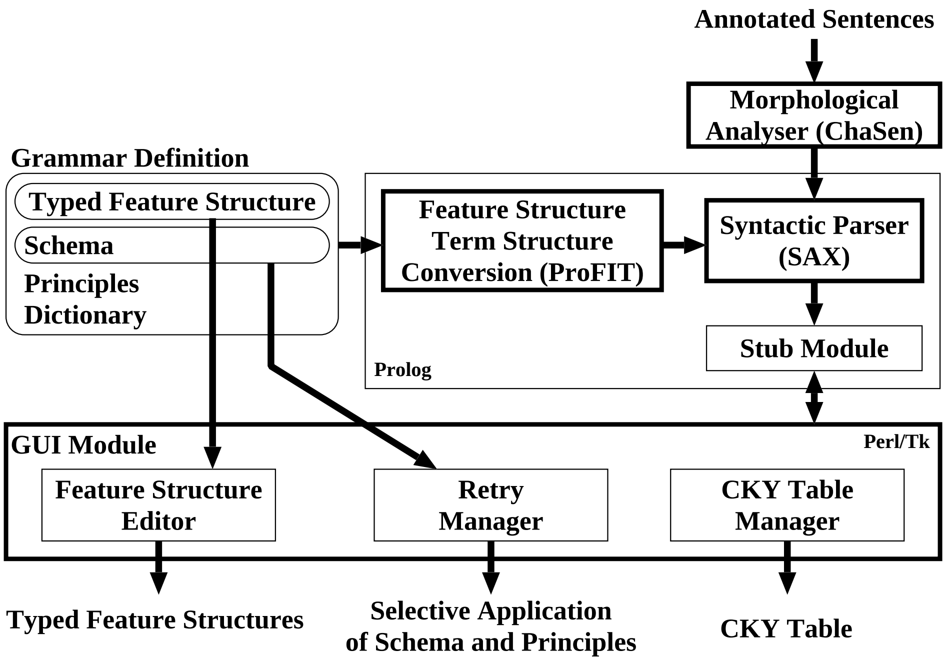

Figure 1: Organization of Modules

tence in order for the grammar to handle it. We defer extending the grammar itself until we have enough annotated sentences to view the problems as a whole. These annotated sentences are also useful to get information about the constructed grammar; they describe a kind of specification of the grammar. When we modify the constructed grammar in order to achieve more efficiency in parsing or adapt to specific domain, we can consult annotated sentences instead of large and complex grammar. Moreover, these annotation can be seen as preprocessing; this is not an empty assumption since recent researches on statistical sentence segmentation and phrasal dependency analysis become more and more practical.[1]

In order to view the problems as a whole, it is also important to analyze a number of sentences shared by many people. We are currently preparing GUI-based, easy-to-use tools for such collaboration.

3 Acknowledgement

We are in debt to Dr. Graham Wilcock for advice of use ProFIT and combine it to SAX. The previous VisIPS system was developed by Mr. Takuya Nakayama (Nakayama and Matsumoto, 1996), and he gave us various support for extending the system. Mr. Kazuma Takaoka built early prototype system and collaborates with us to develop the grammar. We thank Dr. Kiyota Hasimoto, Dr. Koiti Hasida, and Dr. Akira Otani for a number of advice on the Japanese grammar.

[1] Global Document Annotation (GDA) project (Hasida, 1997) offers another viewpoint to annotation. This project proposes standard tag sets and aims to promote development and spread of useful applications which exploit those tags.

References

Gregor Erbach, 1995. *ProFIT 1.54 User's Guide.* Universität des Saarlandes, December. `http://www.coli.uni-sb.de/~erbach/formal/profit/profit.html`.

Koiti Hasida. 1997. Multilingual summarization as an application of global document annotation. In *NLPRS'97 Satellite Workshop on Multilingual Information Retrieval and Summarization*, December. (`http://www.etl.go.jp/etl/nl/GDA/`).

Ray Jackendoff. 1990. *Semantic Structures.* MIT Press.

Yuji Matsumoto, Yasuharu Den, and Takehito Utsuro, 1993. *Natural Language Parsing System SAX Manual.* NAIST, Takayama 8916–5, Ikoma, Nara, JAPAN, March. version 2.0 (in Japanese).

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, and Yoshitaka Hirano. 1999. Japanese morphological analysis system ChaSen version 2.0 manual. IS Technical Report NAIST–IS–TR99009, Graduate School of Information Science, NAIST, April. (`http://isw3.aist-nara.ac.jp/IS/TechReport2/report/99009.ps`).

Takuya Nakayama and Yuji Matsumoto, 1996. *Visual Interface for Parsing Systems VisIPS (Tcl/Tk version) Manual.* NAIST, Takayama 8916–5, Ikoma, Nara, JAPAN, May. version 1.0 (in Japanese).

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

James Pustejovsky. 1996. *The Generative Lexicon.* MIT Press.