論理型文法の枠組における言語分析支援環境

佐野 洋　田中 裕一　佐々木博司†　小野寺浩†　木下 聡‡

(財) 新世代コンピュータ技術開発機構
† 富士通エフ・アイ・ピー株式会社
‡(株) 東芝 総合研究所 情報システム研究所

概要

我々は、論理文法を基本とし単一化という枠組で文法を形式化することと、併せて、形式化の実証や文法の評価を行う統合支援環境の開発を進めている。後者で示すシステムは、記述の枠組についての評価や実証の手立てを構築することで日本語研究の成果を自然言語処理への応用へ反映させてゆくことを目的としている。単一化文法を基本とした句構造ベースによる文法の記述枠組と文法記述形式については改めて報告する予定である。

本稿は、文法の記述枠組や形式を支援し、言語知識についての実験評価を行う環境について報告する。この実験支援環境は、解析と生成の両面からさまざまな言語現象を調査し評価する能力を備えている。豊富な入出力デバイス、ウィンドウシステムやマウスを中心とするダイレクト操作を実現したこの統合環境はICOTで開発された逐次型推論マシンPSI-IIで動作する。

LINGUIST:
A Logic-Based Integrated Natural Language Processing System

SANO, Hirosi　TANAKA, Yuichi
INSTITUTE FOR NEW GENERATION COMPUTER TECHNOLOGY(ICOT)
4-28-1 MITA, MINATO-KU, TOKYO 108, JAPAN.
SASAKI, Hiroshi　ONODERA, Hiroshi
FUJITSU FACOM INFORMATION PROCESSING CORPORATION
DAISAN KYODO BLDG., 4-14, KAMIYAMA-CHO
SHIBUYA-KU, TOKYO 150, JAPAN
KINOSHITA, Satoshi
TOSHIBA Corporation
1, KOMUKAI-TOSHIBA-CHO, SAIWAI-KU,
KAWASAKI, KANAGAWA 210, JAPAN

Abstract

Logic-Based Integrated Natural Language Processing System (LINGUIST) is a system being developed in the Sixth Research Laboratory at ICOT. The system being developed in the framework of logic programming is an experimental natural language processing system. It will enable investigation and processing of various linguistic phenomena in the approach to parsing and synthesis of natural language by computer. The parser is based on a logic grammar formalism, Definite Clause Grammars(DCGs) (Pereira and Warren 1980), which allows logic grammars to be written in a 'phrase structure grammar' fashion.

The grammar uses a unification grammar which has a modular treatment of syntax and semantics. Parsing with the unification grammar assumes inversion of the analysis process. By its bidirectional nature, LINGUIST can parse and generate natural language using the same grammar.

1 Introduction

The research in the domain of natural language processing is an integral part of the area of artificial intelligence. A system has been developed whose main purpose is to cover linguistic phenomena in the approach of parsing and synthesis of natural language. LINGUIST (Logic-based Integrated Natural Language Processing System) is an experimental natural processing system with three purposes: (1) verifying the more detailed nature of the framework of natural language (Japanese) in a strict enough sense to take an objective view; (2) developing more useful grammars that can be used widely in the domains where the natural language interface to an information retrieval component is used as a device of intelligent system; and (3) having an tool for processing Japanese and trying thoroughly our ideas concentrated around grammars.

In point 1, although numerous linguists' ways of thinking about the Japanese language have been represented, there is no suitable, comprehensive framework for the computational linguistic treatment of Japanese. Very few serve a theoretical purpose. This has led to the development of LINGUIST, which is intended both for computational linguists and for those who take an interest in Japanese. In point 2, people who wish to pursue the research of intelligent system with a natural language interface has to develop its interface, before they concentrate on their own research interest. Using LINGUIST, they are not forced to write grammar rules and develop natural language interface any more. With regard to the last point, LINGUIST provides a useful and powerful tool for grammar writers who wish to have an interactive environment where they try out and verify their ideas about grammar rules for Japanese. Their advances in grammatical function and syntactical details can clarify a variety of linguistic phenomena.

In the next section, we discuss briefly the integrated environment. Section 3 describes software and hardware environment around PSI, on which the system LINGUIST implemented. In section 4, after overview of configuration of LINGUIST, summary and future directions conclude the paper.

2 The Integrated Natural Language Processing System

This section outlines the integrated natural language processing system LINGUIST implemented on the Prolog-based environment of the PSI machine. Generally, tools for natural language processing and the systems built around these tools have an aspect as worldly-wise workbenches, built on artificial intelligence workstations. Using the system, linguists attempt to form ideas about a grammar and can inspect a number of the research interests on computational linguistics. They can verify these ideas about the grammar and can proceed with certainty as to what lies ahead for more important and significant linguistic phenomena.

After the overview of the implementation strategy of the system, the current state of LINGUIST will be described.

2.1 The system LINGUIST

The system LINGUIST has been developed in the sixth research laboratory at ICOT within the research activity "Acquiring sound facilities for representation of Japanese grammar in current Japanese and for the development of Japanese grammar". This research aims at investigating characteristics of current Japanese, exploiting a formal representation of these features, and developing grammar rules proper to be used in computers. This grammar is to cover a wide band of linguistic phenomena. Moreover, the coverage of the grammar dose not set a limit on written language. It will remain close to the spoken language and will have the power to deal with sentences that are made up of incomplete structures.

The general attitude of this research is to operate in an integrated environment where different sources of linguistic knowledge are used and tools such as parser, generator and system manager are adjusted so that the efficiency for developing grammar rules is increased. Hence, LINGUIST has been designed to make it possible to experiment with the tools interactively. The system's architecture contains translator, debugger, pooling component, database manager and system manager. The system's first step is to establish a state for use of parsers where the grammar written in DCG style is translated into two parsers. One, for analysis, is a parser that works in bottom-up fashion and another, for generation, is a parser that handles input data in top-down fashion. In addition to the translation, the style of the grammar form is checked at this stage. Here, the parsers are ready for use.

Next, the syntactic analysis and text generation are performed. The role of the polling component is to maintain the data flow from input devices to parsers and vice versa. This also supports the communication between two kinds of parser underlying LINGUIST. A tool that shows constituent structure would be convenient for viewing the output of the parsers. An inspector attached to the pooling component performs this function.

If parsing has failed, the user can investigate error parts of grammar rules. In general, one problem with recovering faults is that it takes painstaking effort to discover the cause of errors and it takes a long time to correct them. However, instead of revising the source grammar rules and re-translating them into two kinds of parser, LINGUIST serves as a debugger with facility in performing quick recovery from the failure. The debugger consists of a visual tracer that is able to keep parsing process in view and an editing tool dealing with error parts of grammar rules. The grammar rules which produce the error are modified and re-translated partially. Using the debugger, the error part is focused on and the user can handle only some not all of the grammar rules.

The database manager deals with the process of entering data and providing editing the lexicon. Of course, the word data is reconstructed partially when required. To use useful and powerful tools effectively mentioned above, LINGUIST is kept under control of system manager. This reports system status, guides the user and shows the way of what is right and just. The system manager also deals with recovering errors that are due to the carelessness of the user.

3 The State of LINGUIST

The challenge of natural language processing by computer is providing an increasing number of interesting ideas for computational linguistics. These ideas have also contributed to the progress of Artificial Intelligence. LINGUIST has been doing well as a development and verification tool in the research of Japanese grammar with respect to computational linguistics. The grammatical formalism which is employed and has been implemented in LINGUIST is Localized Unification Grammar (LUG) and the representation of LUG is developed by SANO,H and FUKUMOTO,F. An introduction to LUG is in preparation. This section is devoted to a description of the configuration of LINGUIST and presents some concepts of interface management.

3.1 Overview of the system

LINGUIST is implemented on a PSI-II machine. Figure 1 shows a simplified configuration of the system. These basic units may be found in any Natural Language Processing system. Some type of input and output devices must be used to provide a means of communication between LINGUIST and the outside world. LINGUIST has several types of input and output devices, and has a data stream mechanism. The input and output streams that store data as a memory does enable wide use of devices such as file system, window system, keyboard equipment and so on. The input feeds data, a token stream for analyzing and constituent structure for generation, into each stream. The output collects the results from the streams after they are parsed.

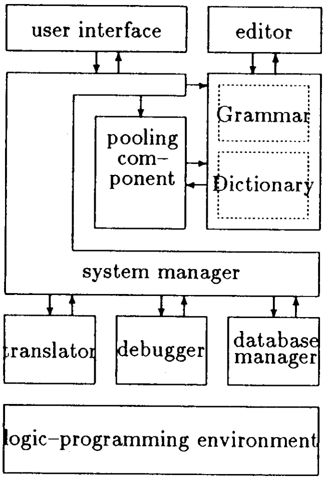


Figure 1. Control Structure

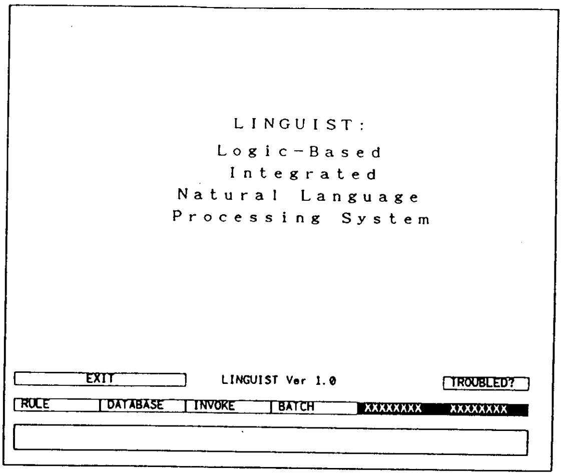
The data pool, as a memory, holds input data and makes it available for parsing when as required. After parsing, the results are stored in the data pool. The inspector accessory enables of closer inspection of the results. Data and results stored in the data pool can be removed easily.

LINGUIST is made up of a series of tools. As each tool is based on windows, the grammar writer works directly with visual representation of the grammar-related objects. The visualization offered by a direct manipulation style tool helps us understand problems related to grammars. Using a mouse, a user such as a grammar writer selects a desired behavior from menus attached to the windows.

3.2 Tools

Agent

Figure 2 shows the main window of LINGUIST, in which the major mode of processing things is selected using menus attached to this window.

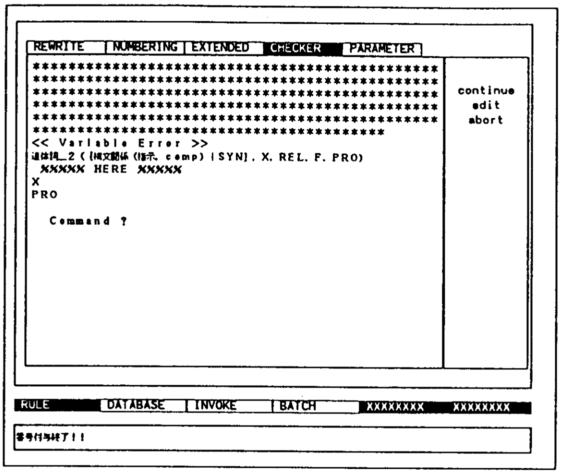


LINGUIST:
Logic-Based
Integrated
Natural Language
Processing System

Figure 2. The Main LINGUIST Window.

Rule manager

Rule manager has the window given in figure 3 where one can view, examine a grammar to find out whether it is accurate, and if so modify it with a text editor attached to this window. One can also specify the file in which the grammar is written.



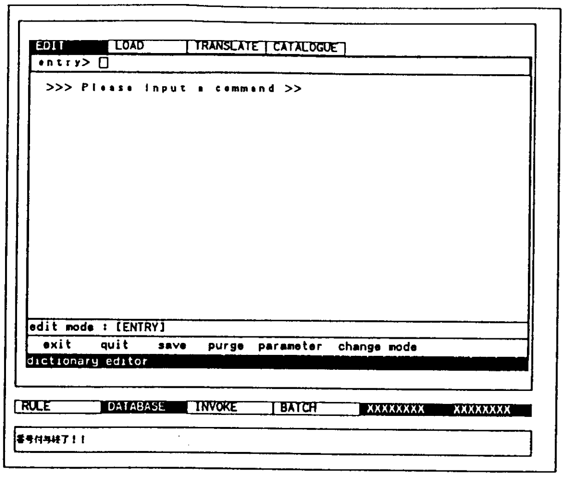
```
****************************************************
****************************************************
****************************************************
****************************************************
****************************************************
*****************************************
<< Variable Error >>
連体詞_2 ( [構文関係 (指示, comp) | SYN] . X. REL. F. PRO)
 XXXXX HERE XXXXX
X
PRO

   Command ?
```

番号付与終了!!

Figure 3. The Rule Manager Window.

Database manager

Entries of words, especially lexicons, are entered and modified with Database manager with the window shown in figure 4. In the course of entering and modifying a word, the database manager provides a set of subwindows for easy operation.



```
entry> □
 >>> Please input a command >>
edit mode : [ENTRY]
```

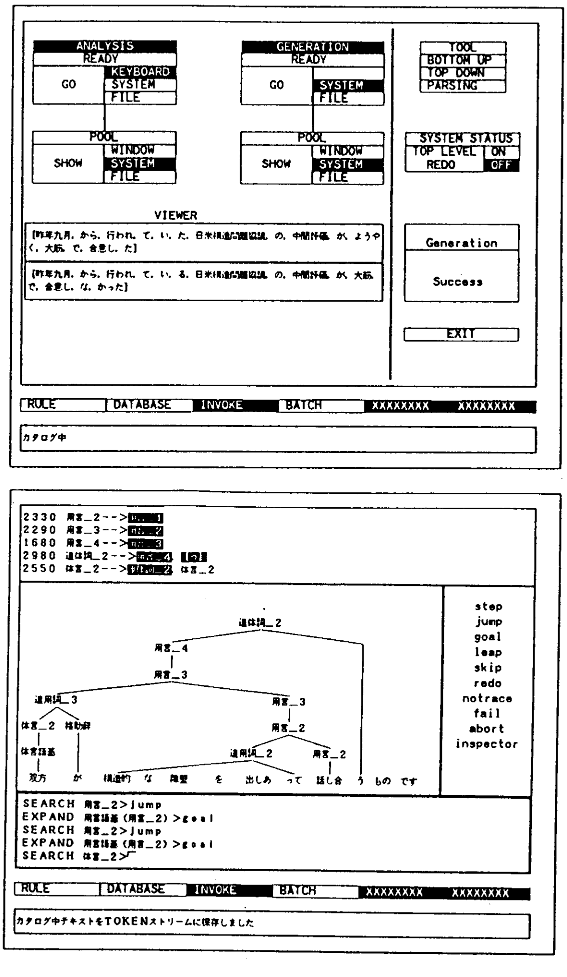
番号付与終了!!

Figure 4. The Database Manager Window.

Viewer

The viewer is a window where one can view, analyze and generate a sentence. The viewer also deals with the complex processes in natural language processing. The process being analyzed is traced graphically with a visual debugger, as is generation process. The debugger serves as a subwindow of the viewer and includes a set of tracing functions: step, jump, leap, fail, abort, and so on. The data structure at any step of parsing can be inspected.

The tool for translating a grammar into ESP codes is integrated in the viewer. This tool is also used for revising and maintaining the grammar with a text editor specialized in the manner of editing the grammar.



[昨年九月. から. 行われ. て. い. た. 日米構造問題協議. の. 中間評価. が. ようやく. 大阪. で. 合意し. た]

[昨年九月. から. 行われ. て. い. る. 日米構造問題協議. の. 中間評価. が. 大阪. で. 合意し. な. かった]

カタログ中

```
2330 用言_2-->[illegible]
2290 用言_3-->[illegible]
1680 用言_4-->[illegible]
2980 連体詞_2-->[illegible] [illegible]
2550 体言_2-->[illegible] 体言_2
```

```
SEARCH 用言_2>jump
EXPAND 用言語基 (用言_2) >goal
SEARCH 用言_2>jump
EXPAND 用言語基 (用言_2) >goal
SEARCH 体言_2>
```

カタログ中テキストをTOKENストリームに保存しました

Figure 5. The Viewer.

4 Summary and Future Directions

This paper has provided an overview of LINGUIST that is being developed in the framework of logic programming. This system is an experimental natural language processing system. Parsing with unification grammar formalism is a characteristic that assumes inversion of analysis process. The usefulness of a generation based on the same grammar and dictionary data is a means by which to try the quality of representation of grammar formalism we have developed on the system.

The interactive aspect of LINGUIST involves two important roles for developing grammar rules. One is efficiency, the other is related to the factorability of natural language processing in computational linguistics. These take into account an integrated environment so that the computer can be viewed as a co-operator in doing work around the research area of computational linguistics.

As experience with LINGUIST for developing grammars and ideas concentrated around grammars increases, more will be understood about the requirements for LINGUIST. Some desired characteristics for a natural language processing system such LINGUIST include the following.

- Usability.

  Interactive system for Natural Language Processing is a complex software system. It is clear that usability is a sufficient quantity for efficiency and productivity of the user.

- Guidance.

  In operating the system, wrong operations and errors throw the user into confusion. With guidance and help from the system, the user instantly recovers from the confused state and saves time.

- Extensibility.

  The system, as an experimental environment, should be extensible. It is clear that research on natural language processing is unlimited. So the tools integrated in the system can be easily modified.

- Database.

  Outputs of all tools within LINGUIST, including revised grammar, its program code, documentation and even system current status, should be stored and brought back whenever necessary. LINGUIST used and now uses file systems provided by SIMPOS. This is convenient but lacks flexibility. Hence, an advanced database management system is required.

Acknowledgement

The authors are grateful to their colleagues who work in cooperation with them in JBA (Japan Business Automation Corp.). Special thanks are due to KUNIEDA, Yumiko and NAGASAWA, Yoko for their constructive proposal, as well as for their work in the development of the dictionary embedded in LINGUIST.

References

[1] Fernando C.N. Pereira, David H.D. Warren. (1980), *Definite Clause Grammars for Language Analysis—A Survey of Formalism and a Comparison with Augmented Transition Networks* Artificial Intelligence 13 (1980), 231–278.

[2] H. REX HARTSON, DEBORAH HIX. (1989), *Human-Computer Interface Development: Concepts and Systems for Its Management* ACM Computing Surveys, Vol.21, No.1, March 1989, p32–p44

[3] Jerry R. Hobbs, Mark Stickel, Paul Martin, Douglas Edwards. (1989), *Interpretation as abduction* 26th Annual Meeting of the ACL Proceedings, 1989.

[4] Kevin Knight. (1989), *Unification: A Multidisciplinary Survey* ACM Computing Surveys, Vol.21, No.1, March 1989.

[5] Martha E. Pollack, Fernando C.N. Pereira. (1988), *An Integrated Framework for Semantic and Pragmatic* 26th Annual Meeting of the ACL Proceedings, 1988.

[5] Martha E. Pollack, Fernando C.N. Pereira. (1988), *An Integrated Framework for Semantic and Pragmatic* 26th Annual Meeting of the ACL Proceedings, 1988.

[6] Marc Moens, Jo Calder, Ewan Klein, Mike Reape, Henk Zeevat. (1989), *Expressing generalizations in unification-based grammar formalisms* Fourth Conference of the European Chapter of the ACL Proceedings, 1989.

[7] Mats Wiren. (1989), *Interactive Incremental Chart Parsing* Fourth Conference of the European Chapter of the ACL Proceedings, 1989.

[8] Yuji Matsumoto, Masaki Kiyono, Hozumi Tanaka. (1983), *BUP toransreta* Densou-ken-i-hou, Vol.47, No.8 (1983).

[9] Michael C. McCord. (1989), *DESIGN OF LMT: A PROLOG-BASED MACHINE TRANSLATION SYSTEM* Conputational Linguistics, Volume 15, Number 1, March 1989.

[10] Oliviero Stock. (1989), *PARSING WITH FLEXIBILITY, DYNAMIC STRATEGIES, AND IDIOMS IN MIND* Conputational Linguistics, Volume 15, Number 1, March 1989

[11] R.Johnson, M.Rosner. (1989), *A Rich environment for experimentation with unification grammar* Fourth Conference of the European Chapter of the ACL Proceedings, 1989.

[12] Robert C. Moore. (1989), *Unification-Based Semantic Interpretation* 27th Annual Meeting of the ACL Proceedings, 1989.

[13] Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, Fernando C.N. Pereira. (1989), *A Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms* 26th Annual Meeting of the ACL Proceedings, 1989.