COMPUTER PROCESSING OF CHINESE & ORIENTAL LANGUAGES
Vol. 5, Nos. 3 & 4, November 1991

AN LFG-BASED MACHINE TRANSLATION SYSTEM

One-Soon Her, Dan Higinbotham and Joseph Pentheroudakis

ABSTRACT

The ECS machine translation system is based on the theory of LFG (Lexical Functional Grammar) and employs a transfer approach. LFG assigns two levels of linguistic information to a sentence: the constituent structure and the functional structure. The functional structure, a hierarchical attribute-value matrix representing the relatively language-independent underlying grammatical relations of a sentence, serves ideally as the basis for transfer, while the highly language-dependent constituent structure is discarded. This paper introduces the translation procedure, parsing and generation strategies, and linguistic techniques employed in the ECS system. Some rationale for the design of the system will be given. While rules and lexicons of seven language pairs have been under construction, the ECS system is designed as a universal tool for machine translation development.

Keywords: Machine translation, LFG, Lexical Functional Grammar, Parsing, Generation, Oriental Languages.

0. INTRODUCTION

This paper gives an overview of the ECS machine translation (MT) system, which is designed around the linguistic theory of LFG, Lexical Functional Grammar [1-3], as a universal tool for MT development. Four language pairs with specific direction of translation have been under development for the last five years and are currently reaching maturity: English-to-Chinese, English-to-Japanese, English-to-Korean, and Korean-to-English. All three Oriental languages can be processed in their traditional

---

Manuscript received on December 4, 1989; revised September 21, 1990. The authors are with ECS, Inc., 455 N. University Ave., Suite 202, Provo, Utah 84601.

writing systems. Few existing systems have multilingual capacity for oriental languages [4-8]. Prototypes of English-French, English-German and English-Spanish are also available. In addition, the MANTRA project at the University of Bergen employs the system for English and Norwegian translation.

1. BASIC APPROACHES TO MT AND THE ECS SYSTEM

The ECS system is an implementation of the indirect translation, transfer approach. Roughly three approaches to machine translation can be identified: direct translation, transfer, and interlingua [9]. The direct translation strategy, adopted by most first generation systems (e.g., SYSTRAN and Georgetown's GAT), manipulates the input sentence, usually in different stages, in ways motivated solely by the intended target sentence. Therefore, there is no independent linguistic analysis of the source or the target language. The indirect strategy, on the other hand, analyzes the source language sentence and synthesizes the target language sentence independently. The transfer approach and the interlingua approach are two alternative ways of linking the analysis and the synthesis. Figure 1 depicts the two options.

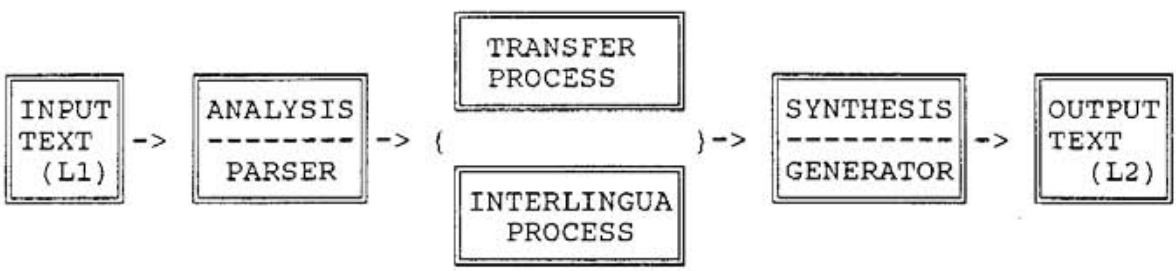


Figure 1. A model of the indirect translation

The interlingua process converts the analysis into a supposedly universal meaning/knowledge representation; this universal representation then serves as the basis for synthesis. In practice, the interligua is usually a conceptual dependency representation language extended with higher-level structures, e.g., Schank's scripts [10]. If a transfer strategy is implemented, then the analysis of the input sentence will be transformed into that of the desired target sentence. For example, if the result of the analysis is a tree, the transfer process will transform the tree into the shape that the tree of the intended target sentence would have. The ECS system produces both a tree representation for the constituent and precedence structure and a bracket dag (directed acyclic graph) representation for the functional structure of the input sentence, and only the functional structure enters the transfer process.

We will not go into the debate over transfer vs. interlingua. The transfer approach aims at a translation that is to a certain extent lexically, semantically, and syntactically invariant, while the interlingua approach strives only for semantic

invariance and thus is sometimes criticized as not to provide "translation" in a technical sense but rather merely "paraphrasing" in the target language. A system based on the interlingua approach, however, ideally can do multilingual translation since the interlingua representation is language-independent and can serve as the basis of synthesis for any target language, while the transfer process must have transfer rules for a specific language pair and a specific direction of translation. ECS prefers the transfer approach for we feel that given the current state of art of AI research an adequate, suitable interlingua is not yet possible and also that a totally language/culture-independent interlingua may not exist. Thus, in order to build a practical translation system, it is best to start off with a shallower analysis that may provide a relatively language-independent representation and deepen the analysis as experience and theory advance. The linguistic theory of Lexical Functional Grammar (LFG) and its formalism seem to be a perfect choice for this position.

2. WHY LFG?

LFG is generative in the original sense given by Chomsky yet non-transformational, expressive in formalism and constrained in theoretical constructs. The theory attempts to model the organization of human linguistic knowledge as well as the processing of linguistic information. this spirit is no doubt to a large extent due to the collaboration of the two main architects of LFG, Joan Bresnan, a generative linguist, and Ronald Kaplan, a psycholinguist and computer scientist.

The first reason for the choice of implementing a LFG-style grammar for linguistic analysis in the ECS MT system is therefore that LFG is a sound and computationally-oriented theory. Secondly, LFG and its formalism are well-known among computational linguists and are therefore much preferred to ad hoc system-specific local grammars such as the ones used at ALPNET, WEIDNER (WCC), and SYSTRAN. Various current MT research projects use LFG or LFG-like framework for linguistic analysis, such as the German-Japanese SEMSYN project at the University of Stuttgart, and the KBMT (knowledge-based machine translation) system developed at CMU (Carnegie-Mellon University) [11-12]. LFG was also employed in the previous English-Japanese project at UMIST (University of Manchester Institute of Science and Technology).

The greatest advantage that the LFG formalism offers a MT system is its division of constituent (c-) structure and functional (f-) structure. C-structure is mostly language-specific and the ordering of its elements is significant and not random, while f-structure is to a large extent language-independent and the ordering of the attribute-value pairs contained within it is entirely free and random. For example, given the following rather simplified lexical entries and rules in 1 and 2 (stated in our modified LFG notation), the sentence "Mary loves John" will be assigned the c- and f-structure illustrated in Figure 2.

1. a. Mary N,

$$\begin{bmatrix} \text{PRED} & \text{'MARY'} \\ \text{PERSON} & \text{3rd} \\ \text{NUMBER} & \text{SG} \end{bmatrix}$$

b. John N,

$$\begin{bmatrix} \text{PRED} & \text{'JOHN'} \\ \text{PERSON} & \text{3rd} \\ \text{NUMBER} & \text{SG} \end{bmatrix}$$

c. loves V,

$$\begin{bmatrix} \text{PRED} & \text{'LOVE <SUBJ OBJ>'} \\ \text{SUBJ} & \begin{bmatrix} \text{PERSON} & \text{3rd} \\ \text{NUMBER} & \text{SG} \end{bmatrix} \end{bmatrix}$$

2. a. NP: SUBJ
VP
-> (S)

b. V
NP: OBJ
-> (VP)

c. (DET)
ADJ* ϵ ADJUNCTS
N
-> (NP)

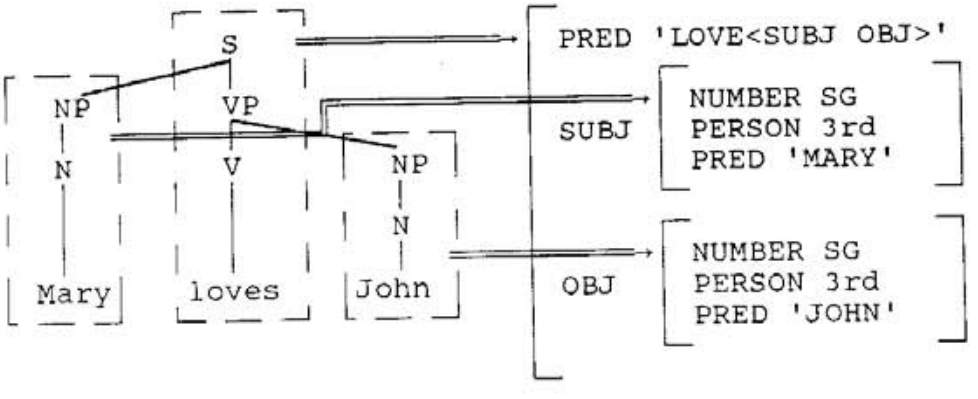


Figure 2. Co-description of c- and f-structure

Without such a division, a transfer system has to manipulate the constituent (tree) structure of the source sentence in order to generate a target sentence, such is the case in WEIDNER (WCC) system, the Japanese Government Project for Machine Translation, and the METAL system developed at the LRC of UT Austin. In the ECS system, the c-structure of the source sentence is discarded after parsing. Only the f-structure of the source sentence is used as the basis for generating the target sentence; therefore, only minimal amount of manipulation needs to be performed during the transfer stage. In addition, a knowledge-based approach should be easily implementable within the LFG formalism. Its formalism can therefore be implemented with either a transfer approach such as the ECS system or a interlingua approach such as CMU's KBMT system.

3. PARSING STRATEGIES AND THE ECS PARSER

The parser is certainly the most important part of the entire translation software. The ECS system uses an active chart parser that is responsible for both parsing and generation [13]. The parser proceeds bottom-up first, and then top-down, either left-to-right or right-to-left, and in a breadth-first manner. If a lexical entry found in the sentence to be parsed is the central element of a rule, then that rule will be executed first. Edges are arcs that span stretches between words in a sentence; a complete edge is a constituent with all its components already found; it therefore remains inactive. After all relevant bottom-up rules (up-rules) are tried, perhaps with some complete edges created, the parser then proceeds in a top-down manner, with some of the edges created by the up-rules remaining incomplete (thus still active). In fact, the very characteristic of a chart parser is that every partial match is kept around in case it can be completed later.

There being much debate over the top-down versus bottom-up strategy of parsing, the ECS approach capitalizes the advantages of both. Up-rules are either fired by a lexical item or by another up-rule. Thus, up-rules that are irrelevant to the words in the sentence will never be tried. However, bottom-up parsers are local and tend to lack foresight in determining when enough optional elements are found. Therefore, by parsing top-down unless otherwise specified, the ECS parser allows the linguist to construct rules in ways most efficient for parsing. Whether the parser works left-to-right or right-to-left can be specified before translation according to the language that is to be parsed. Since increasingly more memory is possible and speed is a major consideration in the evaluation of MT systems, a breadth-first strategy is preferred to a depth-first strategy.

Every lexical item is associated with a functional structure (represented in a bracket dag format), as seen in examples of 1. When words are found and combined into phrases by the parser, their functional structures are unified destructively by the parser to form a new functional structure associated with the phrase. Therefore, a higher category will not be built even if all the elements are found but unification has failed. Whenever a higher category is built there is always a functional structure

associated with it. It is possible to have more than one final parse tree, but only the ones associated with well-formed f-structures will be selected. Again, only the final f-structure enters the transfer module to produce a functional structure known as the "transfer dag" as the basis for generation.

Other special features of the ECS parser include: a weighing scheme for parse selection, successive levels of rules, and error recovery.

Implementation of the weighing scheme is for the parser to determine the optimal parse when more than one parse is reached. (The parser produces all possible parses.) For example, the sentence "John moved in Mary's house" has two parses where the PP "in Mary's house" may or may not be subcategorized by the verb and thus creates two different interpretations of the sentence. However, the preferred reading is likely to be the one with the PP subcategorized by the verb. If so, then what the linguist can do is to put more "weight", by assigning a higher numerical value, on the PP when it is interpreted as the subcategorized oblique locational phrase.

That rules can be applied in successive levels also has important implications. It has been observed that ungrammatical sentences are often understood and therefore parsable for humans but most parsers do not reflect that characteristic of human parsing. If these parsers are to reflect human linguistic behavior, they would predict that people do not comprehend ungrammatical sentences. While certain ungrammatical sentences are indeed unintelligible, others are without question easily understandable. Charniak's parser "Paragram", where a numerical rating scheme is employed, is primarily motivated to deal with this criticism [14]. Furthermore, many types of linguistic structures are in the "gray zone" of native judgement of grammaticality, for instance the numerous ?-preceded sentence examples in linguistic articles. Dialectal variations are also to be considered. The ECS system allows rules to be applied at successive levels; the higher the level, the less confining its rules, and the parse is therefore less likely to fail. It thus provides the linguist with a device to account for the grammatical variance within a language. To illustrate, subject-verb agreement can be strictly constrained by rules of level one and ignored by rules at a higher level. An otherwise good sentence with violation of subject-verb agreement will fail parsing at level one and proceed to higher levels and succeed. The parser will stop when a successful parse is reached and will not go on to the next higher level. Another possible and very useful application of successive levels of rules is to account for the frequency of use of syntactic structures. This enhances the efficiency of parsing. Rules accounting for the less frequently used structures can be placed at a higher level so that a sentence with structures of high frequency can be parsed earlier, and thus more efficiently since less rules are tried.

If all rules have been tried and no parse spans the whole sentence, the parser then creates a best guess by pasting together the longest constituent phrases in the order they occur. the transfer process then still proceeds to build a transfer dag. A recovered translation is clearly marked for the convenience of post-editing.

Generation, as depicted in Figure 3 below, proceeds in three stages: first, target words in the transfer dag are looked up in the target lexicon; second, necessary inflection of morphological elements and target function words, if any, are added; and finally, the linearization rules map lexical items and grammatical functions in the target f-structure to the target sentence.

Incidentally, while it is irrelevant to the translation process per se, we note that the development environment of the ECS system includes a well-developed multi-level debugging capability, a flexible rule editing component, and a suite of satellite utilities allowing the linguist to inspect and maintain the lexicon and other aspects of the linguistic database.

4. TRANSLATION PROCEDURE

The following figure depicts the flow of operations during the execution of the translation engine.

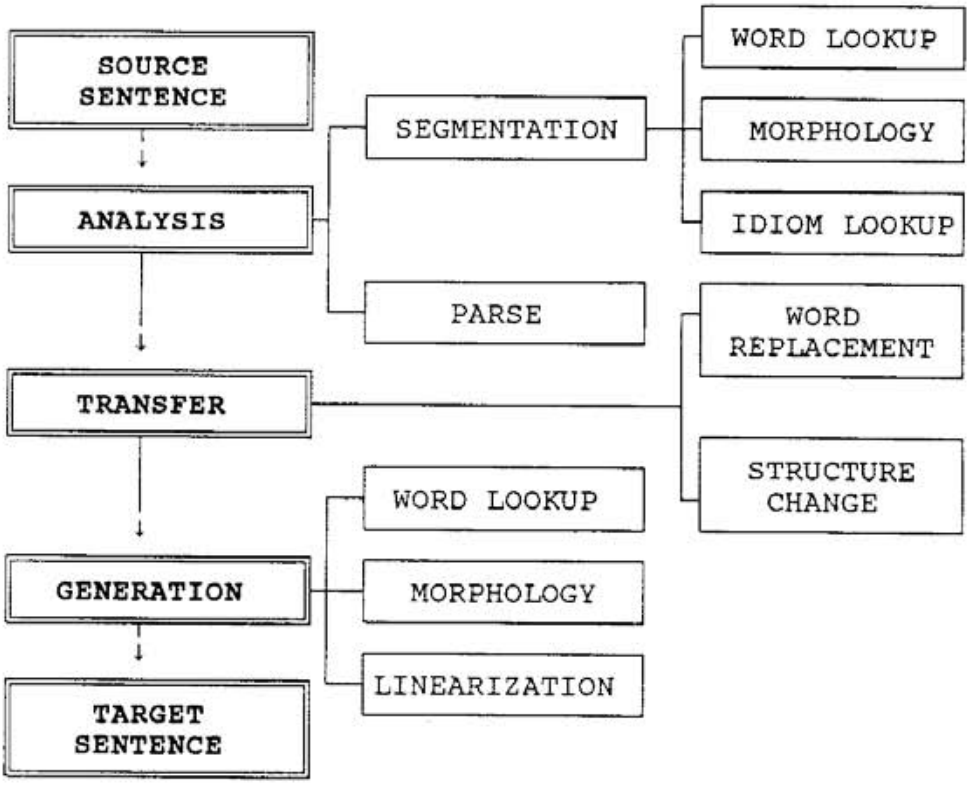


Figure 3. The flow of translation operation in the ECS system

As discussed before, the system employs a transfer approach as illustrated in Figure 1. The figure above is a more detailed account of the translation process. Note,

however, the modules in Figure 3 may be conceptual and do not necessarily correspond exactly to the modules that constitute the translation software.

5. LINGUISTIC TECHNIQUES AND THE ORGANIZATION OF LINGUISTIC ANALYSIS

Although eventually, after compilation, all linguistic information is stored in the dictionary as data base which the parser accesses during translation, conceptually linguistic analysis in the system can be organized as shown in Figure 4. Linguistic analysis before compilation is organized into text files accordingly by the linguists.

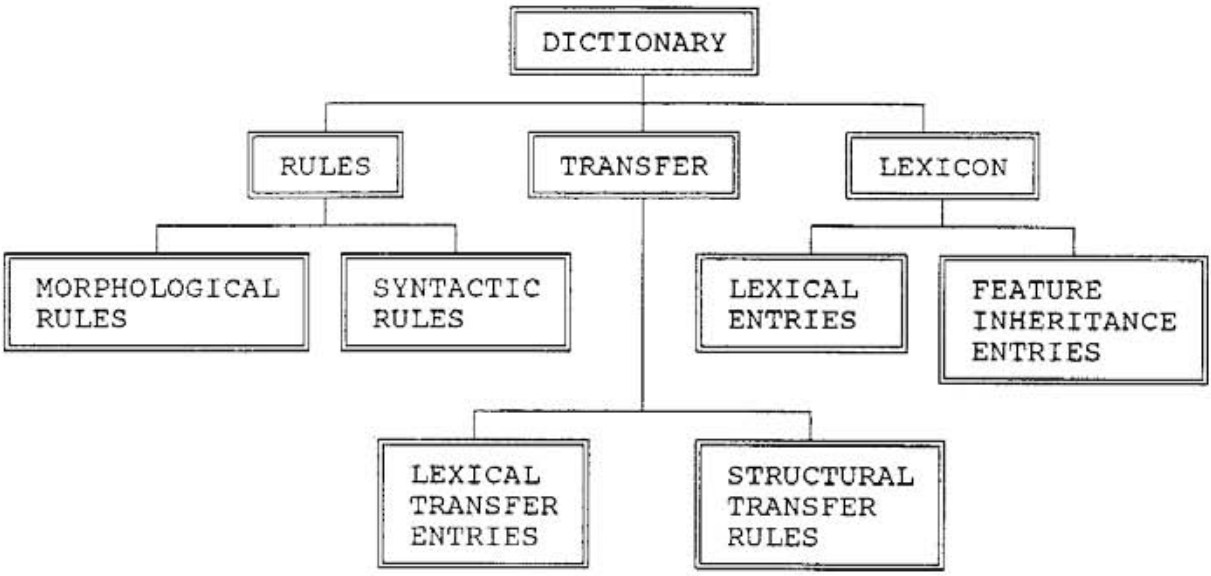


Figure 4. Conceptual organization of linguistic information

The source language and the target language each has a monolingual lexicon. A lexical entry is a dag, an attribute-value pair property list. A certain class of lexical items, e.g., countable nouns or ditransitive verbs, may share certain features. In other words, certain features are totally predictable through the presence of other features. To capture linguistic generalizations as well as to save space and increase speed, a lexical entry contains only features that are idiosyncratic, or unique, to that lexical item. The predictable features can be inherited from the specified "feature inheritance entries". It is important to note that the unification between a lexical dag and a feature inheritance dag is by way of default, or extension as it is sometimes called. That is to say if there is a conflict in terms of the value of certain feature, unification will not fail; rather, the value of the lexical dag is preserved and the conflicting value in the inheritance entry will be ignored. By allowing unification by extension, idiosyncratic behaviors of a lexical item can be fully accounted for and the generalizations can be stated most generally. For example, while most English animate nouns are also concrete and countable, some like "mankind" are not countable.

3. a. FI-N-ANIM:
```
[ ANIMATE   + ]
[ CONCRETE  + ]
[ COUNTABLE + ]
```

b. cat:
```
N,
[ PRED 'CAT' ]
\FI-N-ANIM
```

c. dog:
```
N,
[ PRED 'DOG' ]
\FI-N-ANIM
```

d. mankind:
```
N,
[ PRED 'MANKIND' ]
[ COUNTABLE -    ]
\FI-N-ANIM
```

A lexical entry may inherit information from more than one inheritance entry, which may in turn inherit data from other inheritance entries. Incidentally, this feature inheritance operation of ECS' linguistic formalism, or as a theoretical construct, is not part of the LFG theory and formalism and is not implemented in the LFG Workbench, a grammar writing and testing tool, developed at CSLI (Center for the Study of Language and Information), Stanford University [3]. The implementation of inheritance structure allows under-specified lexical entries and therefore a lexicon of reduced size and increased modularity.

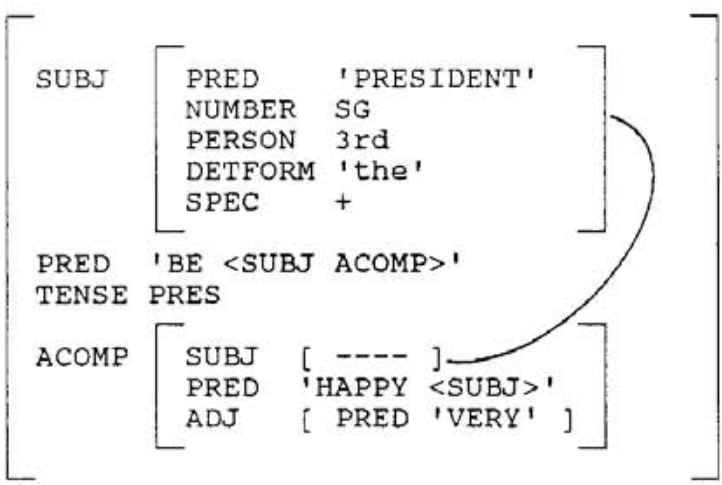
Two types of rules are recognized in each language. Morphological rules account for the analyzable inflectional structure of words and are called by the parser for base form reduction. LFG syntactic rules are context-free phrase structure rules augmented by functional expressions to build the c-structures and the f-structures by pattern matching as well as by unifying their associated dag structure.

Recall that the transfer component only operates on the f-structure of the parse. The lexical transfer function operates on the source word and replaces it with the target word specified in the transfer entry; therefore, this part of the operation is sometimes referred to as the bilingual dictionary. Every lexical item in the source lexicon has a corresponding lexical transfer entry in the transfer component. This design, which totally separates the source lexicon, the transfer lexicon, and the target lexicon, allows more flexibility than others that integrate source and transfer lexicons [15]. A word may have several translations in the target language, depending on the domain of the input text (e.g., in English-Chinese translation, "operation" should translate as 演習 ,

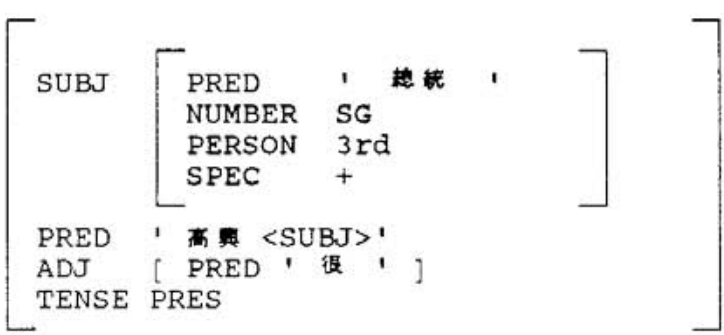
not 手術 , when domain-specification is military), the syntactic context of the word (e.g., "Tell" is translated as 說 when it takes an NP object such as in "tell a lie", but when it takes a sentential complement, it translates as 告訴), or the semantic context (e.g., "raise" translates as 提出 when it takes an idea-type object like "question"). The dag structure of lexical entries and functional structures allows the translation selection to have access to information in all three areas, used in evaluating selection conditions associated with each lexical matching.

Necessary structural changes on the f-structure are performed by transfer rules which are evoked by lexical transfer entries, and the final output of the transfer operations is the "transfer dag" as the basis for generation. For instance, English adjectives, when used predicatively, call upon a transfer rule to absorb the content of ACOMP (adjectival complement) into the higher level and thus deletes the "be" verb, as in "The President is very happy".

F-structure of parse:



Transfer dag:



7. DICTIONARY MAINTENANCE UTILITY

The translation system is also equipped with a most useful and sophisticated tool for the end users to interact with the data base, the dictionary, in order to improve the quality of translation: DMU (Dictionary Maintenance Utility). Through DMU, the end

user can access the dictionary to add or delete an entire lexical entry or change information, e.g., the translation or semantic features, in a lexical entry. However, certain entries in the dictionary can be marked as "reserved" and thus can be looked at but cannot be changed or deleted by the end user. Yet, rules and feature inheritance entries are inaccessible to end users.

8. HARDWARE AND EVALUATION

Currently the ECS system runs on Intel 386 or 486-based hardware under UNIX System V, v.3 or higher, SCO XENIX, v.2.2 or higher, and 6MB of memory is recommended. In order to handle the oriental languages in their traditional writing systems, the hardware must support a Chinese, Japanese, or Korean character set.

As for the speed of translation, no doubt the type of hardware, the size of the lexicons, and the complexity of the linguistic rules are all variables, while the completeness of lexicons and rules determines the quality of translation. Thus, a different language pair yields different results in terms of speed and quality. To ensure translation quality, ECS has a corpus of some 2,00 test sentences which cover various syntactic constructions extensively. On a 386 machine the English-Chinese system with a lexicon of 40K English words and fairly mature English and Chinese rules, for instance, currently translates 95% of the test sentences appropriately at the average speed of 3,300 words per hour. We will give ten sample sentences in the appendix. Note that the output sentences have not been edited.

REFERENCES

1. J. Bresnan, (Ed.), *The Mental Representation of Grammatical Relations,* MIT Press, Cambridge, Mass., 1982.

2. R. Kaplan and J. Bresnan. "Lexical-Functional Grammar: A Formal System for Grammatical Representation," in J. Bresnan (Ed.), 173-281, 1982.

3. C. Kiparsky. LFG Manual, Manuscript, Xerox Palo Alto Research Center, Palo Alto, California, 1985.

4. M. Li and G. Erickson. "Special Parsing Techniques for the Chinese Language," Proceedingof the 1986 International Conference on Chinese and Oriental Languages Computing, 41-45, 1986

5. K. Su., et al. "A Powerful Language Processing System for English-Chinese Machine Translation," Proceedings of the 1987 International Conference on Chinese and Oriental Languages Computing, 260-4, 1987.

6. H. Pan. "An English-Chinese Machine Translation System for Scientific Paper Titles," Proceedings of the 1986 International Conference on Chinese Computing, 311-8, 1986.

7. Y. Liu. "Some New Advances in Computers and Natural Language Processing," Proceedings of the 1986 International Conference on Chinese Computing, 8-14, 1986.

8. M. Nagao, J. Tsujii, and J. Nakamura. "The Japanese Government Project for Machine Translation," *Computational Linguistics* 11.2-3: 91-100, 1985.

9. J. Slocum. "A Survey of Machine Translation: Its History, Current Status, and Future Prospects," *Computational Linguistics* 11.1: 1-17, 1985.

10. A. Tucker. "Current Strategies in Machine Translation Research and Development," in r. Nirenburg, (Ed.), *Machine Translation: Theoretical and Methodological Issues,* Cambridge University Press, Cambridge, 22-41, 1987.

11. J. Carbonell and M. Tomita. "Knowledge-based machine Translation, the CMU Approach," in Nirenburg, R. (Ed.), *Machine Translation: Theoretical and Methodological Issues,* Cambridge University Press, Cambridge, 68-89, 1987.

12. S. Nirenburg. "Knowledge-Based Machine Translation," *Machine Translation* 4.1: 5-24, 1989.

13. T. Winograd. *Language as a Cognitive Process (Volume 1: Syntax),* Addison-Wesley Publishing Company, Reading, Mass., 1983.

14. E. Charniak. "A Parser with Something for Everyone," in M. King, (Ed.). *Parsing Natural Language,* Academic Press, London, 117-150, 1983.

15. M. Li and G Erickson. "Modular Dictionary Design for Chinese-to-English Machine Translation," Proceedings of the 1987 International Conference on Chinese and Oriental Languages Computing, 249-52, 1987.

APPENDIX

1. The investors' buying stocks in Japan is a problem.
   投资者购买在日本的股票是一个难题。

2. All investors who are hoping to buy stocks are friends of the President.
   全部希望购买股票的投资者是总统的朋友。

3. The newest takeover bid is like throwing a match into kerosene.
   最新的出价盘进是像丢进煤油的一根火柴。

4. The investors saw and bought the company that the manager was trying to sell.
   投资者看见了又购买了经理尝试了卖的公司。

5. Economists are more optimistic that the economy will improve than the president.
   经济家比总统更乐观经济会改善。

6. There are no good managers working in this stock company.
   没有任何好经理在这家股票公司里工作。

7. There are three million workers who will lose their jobs if the rumor is true.
   假如谣言真实，有三百万个工人会失去他们的工作。

8. This new stock company will hire one thousand one hundred fifty-two employees.
   这家新股票公司会雇用一千一百五十二个员工。

9. Nobody is buying stocks and the investors do not trust any big companies.
   没有人购买股票，而且投资者不信任任何大的公司。

10. The books that the president 's wife donated to the museum are very valuable.
    总统的妻子捐赠了给博物馆的书很值钱。