[*Machine Translation Summit*, September 17-19, 1987, Hakone Prince Hotel, Japan]

TAURAS: The Toshiba Machine Translation System

Shin-ya AMANO Hideki HIRAKAWA Yoshinao TSUTSUMI

Toshiba Corporation
R & D Center
1, Komukai Toshiba-cho,
Saiwai-ku, Kawasaki
210 JAPAN

Toshiba Corporation
Information & Communication
Systems Lab.
Suehiro-cho, Ome, Tokyo
198 JAPAN

1.INTRODUCTION

Design and philosophy of TAURAS(Toshiba Automatic Translation System Reinforced by Semantics), especially about its grammar system, are presented. A new grammar system developed for TAURAS enables the large-scale grammars be easily written for treating a large amount of documents which have various varieties of sentences.
The goal of this system is multilingual translation for science and technical documents.
The system is written in C for the sake of portability, efficiency, and readability, and implemented on Toshiba minicomputer AS3000 with UNIX*.

2. SYSTEM CONFIGURATION

The translation system has three main elements:
1) Translation unit
2) Bilingual editor
3) Software utilities, e.g. Japanese/English word-processors.

The total software configuration is shown in figure 2-1.
The bilingual editor is equipped with a man-machine-interface devices that makes possible the efficient processing of various problems related to translation, such as where and why errors occurred, for instance.

3. TRANSLATION METHOD

3.1 MORPHOLOGICAL ANALYSIS

The morphological analyzer divides a word into morphemes and constructs a word structure as shown in figure 3-1.

SW: source word (infinitive)
POS: category
NUM: number
GEN: gender
PSN: person
TW: target words (translations)
SM: semantic markers
OTHERS: tense, aspect, modality and so on

PLR: pointer to the lexical rules

Figure 3-1 WORD STRUCTURE

* UNIX is a Trademark of Bell Laboratories

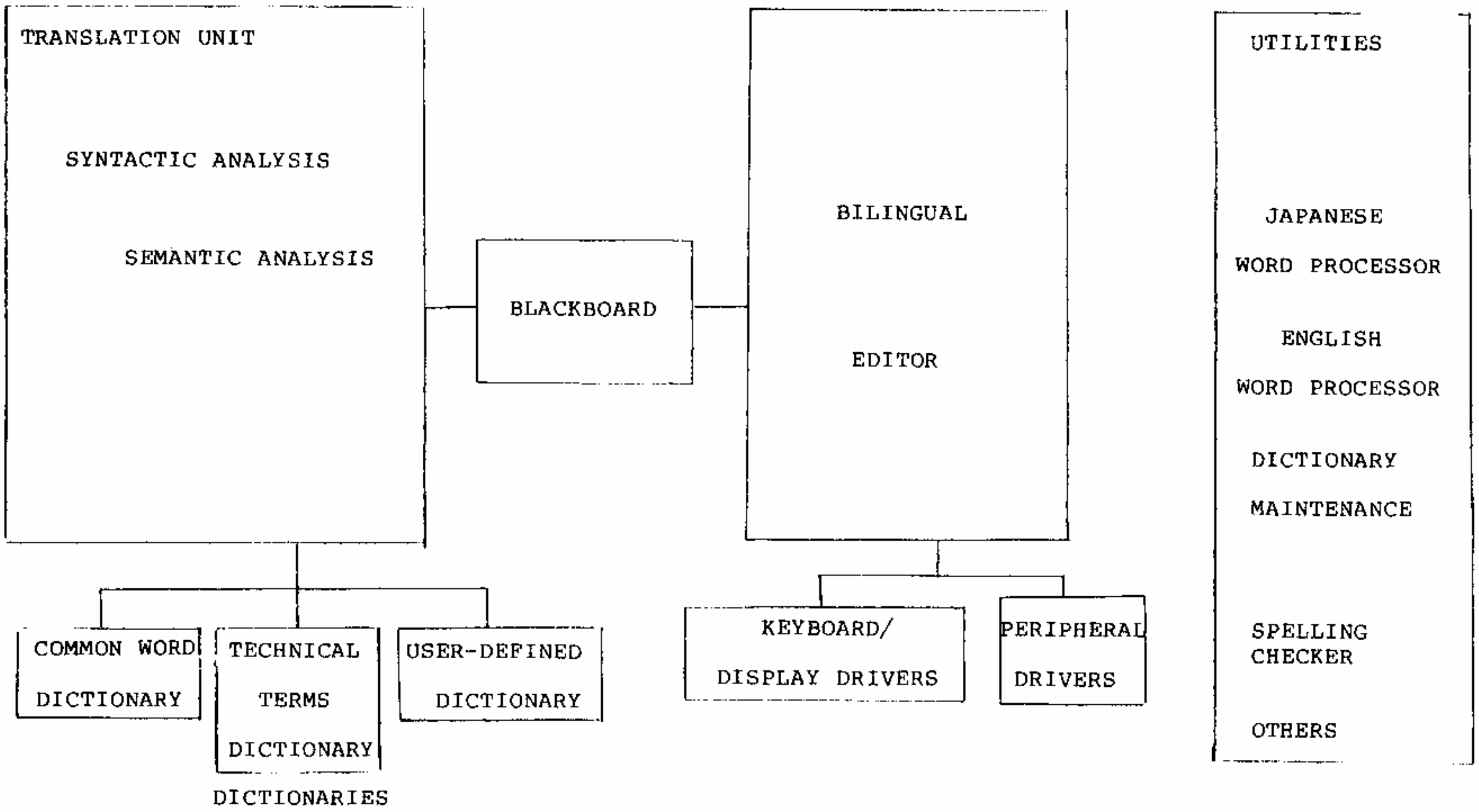

Figure 2-1 SOFTWARE CONFIGURATION

SW, POS, TW, SM, and PLR are necessarily provided by the dictionaries. NUM is also provided by the dictionaries only if the word has an irregular form such as "feet." The input sentence (1) is transformed into a string of word structures (2), for instance .

(1) He is a singer.

(2)

SW	he	be	a	singer	.
POS	pronoun	vbe	det	noun	punc
NUM	singular	singular	singular	singular	-
GEN	male	-	-	-	-
PSN	3rd	3rd	-	-	-
TW	KARE	*	*	KASHU	-
SM	human	-	-	human	-
OTHERS		(TENSE: present)			
PLR	-	*	*	-	_

Here, * means that translations are decided by lexical rules.

Figure 3-2 SENTENCE STRUCTURE

3.2 SYNTACTIC ANALYSIS

Though the syntactic and semantic analyzer are separate in our system, they are not completely independent: rather than working sequentially, they proceed in an interactive manner. Figure 3-3 shows the flow of the syntactic and the semantic processing and their relation.
The features of the syntactic analysis of the system are as follows:

1) The syntactic analyzer always derives only one syntactic structure for a string of categories of a sentence. Structural ambiguities are implicitly represented in the syntactic structure. Semantic analyzer will construct a plausible conceptual structure, resolving such implicit ambiguities.

2) The syntactic analyzer is purely syntactic, and syntactic rules have no semantic conditions.
A well-known example is the following:

1) He promised her to go.
2) He persuaded her to go.

These two sentences have the same surface syntactic structure, but they have different conceptual structures, corresponding to the different interpretations of the deep subject of "to go."

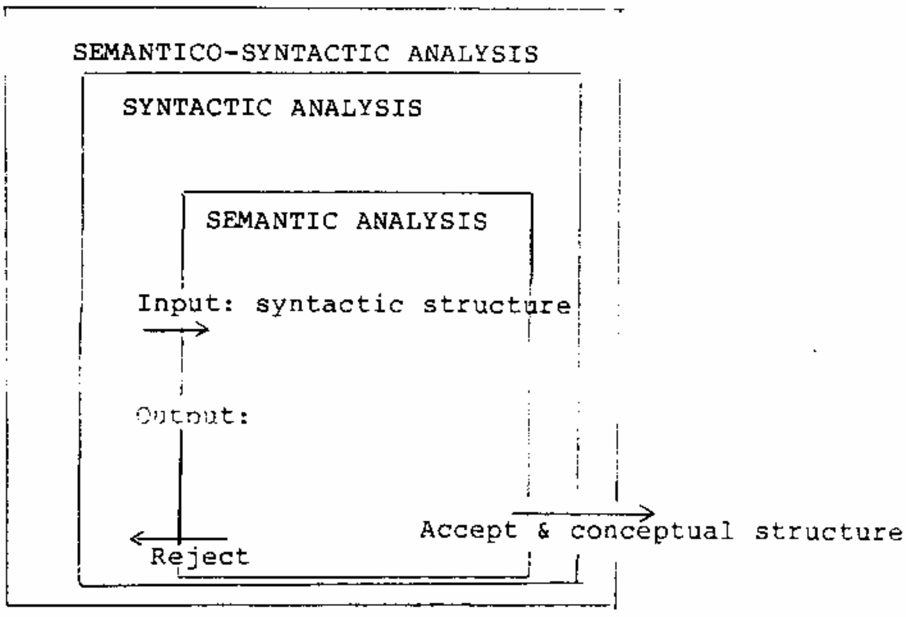
SEMANTICO-SYNTACTIC ANALYSIS
SYNTACTIC ANALYSIS
SEMANTIC ANALYSIS
Input: syntactic structure
Output:
Accept & conceptual structure
Reject

Figure 3-3 SEMANTICO-SYNTACTIC ANALYZER

In our method, the syntactic analyzer makes a unique syntactic structure for the same sequence of categories with different conceptual interpretations. Thus, syntactic rules do not need to have semantic conditions.
The syntactic rules are described by an Augmented Transition Network Grammar(ATNG) formalism.

3.3 SEMANTIC ANALYSIS

3.3.1 METHOD OF SEMANTIC ANALYSIS

Our method of semantic analysis is based on the following hypothesis.

HYPl: Meaning is lexical
The typical example is shown in 3.2. That is,

1) He promised her to go.
2) He persuaded her to go.

Though both sentences have the same sequence of categories, they have different deep subjects for the infinitive "to go." This means these two sentences have different conceptual structures and this difference results from the difference of the meanings of "promise" and "persuade". HYPl implies more than this fact. From the computational linguistic point of view, it insists that semantic rules should not be mixed with syntactic rules. According to this schema, semantic rules are attached to words in the dictionary as lexical rules. If semantic rules are incorporated into syntactic rules in the form of conditions or the like, syntactic rules will be overly intricate. Moreover, syntactic rules, which originally only define the order of words in a sentence, must be written considering the meaning of a resultant sentence. For these reasons, we adopt lexical rules for semantic processing. The conceptual diagram of our semantic processing system is provided in figure 3-4.

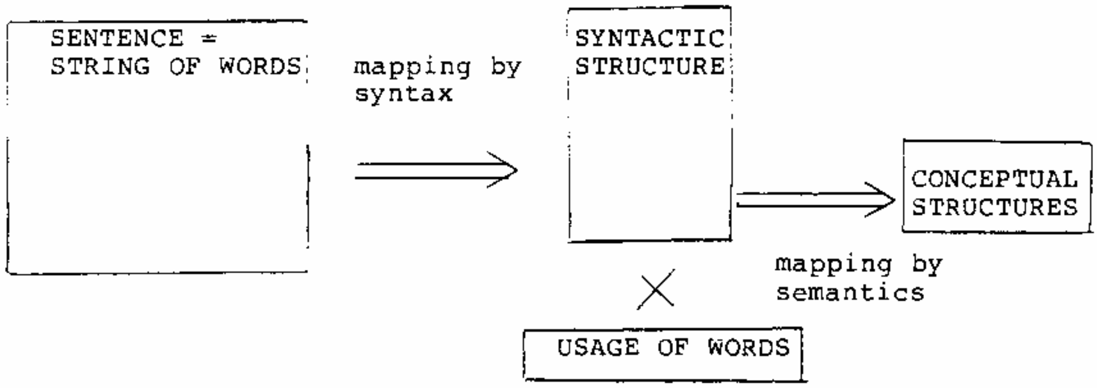

Figure 3-4 ANALYSIS OVERVIEW

3.3.2 NOTATION OF SEMANTICS RULES

Semantic rules consist of tree-to-tree conversion together with conditions and actions. The form of a semantic rule is as follows:

MP = TP; COND; ACT; CTRL

Here, MP is a matching pattern which must match against with a subtree of syntactic structure. If conditions, which are represented by COND, also hold true, then the subtree is converted into the target pattern represented by TP, and actions represented by ACT are executed. CTRL is the control of flow of lexical rules attached to a word. More detailed explanations are given below, using a simple example.

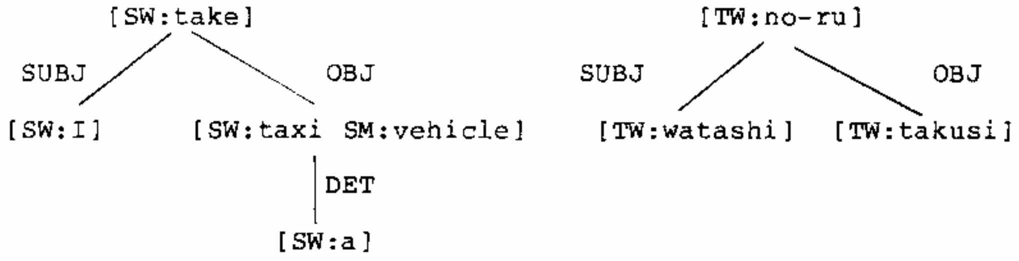

Figure 3-5 SIMPLE EXAMPLE OF ENGLISH/JAPANESE INTERNAL NOTATION

Here, the notation using square brackets is an abbreviation of the word structure and the relevant features. [SW:I] means that the element under question is only SW and other features are irrelevant.

```
MP:  nodel [SW:take]      TP:    [TW:no-ru]
             |
             |  OBJ
             |
      node2 [SM:vehicle]
```

COND: While MP represents conditions about syntactic relations in a syntactic structure, COND represents conditions about features in a word structure.
E.g.

COND: semantic marker of node no.2 = "vehicle"

This condition denotes condition that the translation of "take" into Japanese depends on the nature of the object.

ACT: ACT treats features in a word, adding and deleting some features, and so on.
E.g.

ACT: set-feature(nodel;TW;no-ru)

CTRL: This decides the type of rules: accept-type or reject-type

3.3.3 ROLES OF SEMANTIC RULES

1) Selection of translation
In the above example, the lexical rules for "take" can select a proper translation by referring to the SW slot or SM slot.

2) Processing of idioms
For our purpose, an "idiom" is any sequence of more than one word which must be treated as a single unit for translation purposes. Idioms can be non-contiguous or have variables, such as "put on", "abandon oneself to", and so on. Such idioms can not be registered in the dictionary as a single word, because "put" and "on" may sometimes be separated by intervening words, and "abandon" and "oneself" may both undergo grammatical variability, e.g. inflection, person concord. Syntactic analysis does not treat these as idioms, but simply as individual words and makes a syntactic structure in the usual way. Semantic analysis interprets these words in the syntactic structure as an idiom using lexical rules which are attached to the head word of the idiom. Hence, the idioms are represented as subtrees in the rules, not as a string of words.

3) Lexically structural transfer
Lexical rules for idioms are an example of lexical rules with structural transfer. More generally lexical rules other than idioms have structural transfer. For example, English has a negative determiner "no" while Japanese does not have such a determiner and such negation must be expressed by negation of the predicate. Hence, if this negative determiner appears in a sentence, a lexical rule which transfers the noun negation to the predicate is needed. One of lexical rules of "no" is as follows;

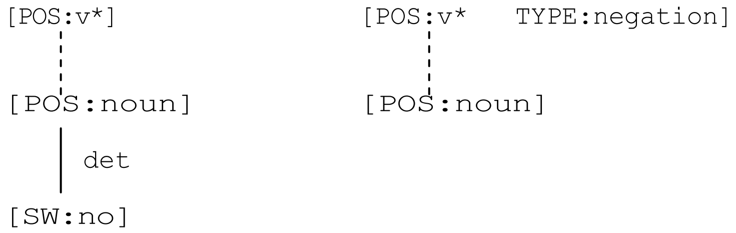

Figure 3-6 TYPICAL TRANSFER PROBLEM CAUSED BY LANGUAGE PAIR DIFFERENCE

Here, "*" in "v*" is a wild-card character and "v*" means verb group. The arc depicted by a dotted line means an ancestor-descendent relation while a rigid line indicates a parent-child relation.

3.4 SYNTACTIC GENERATION

So far we have computed a conceptual structure for the target language. The next stage is the generation of the target sentence.
The roles of syntactic generation are as follows:

1) Determine word order in a conceptual tree
2) Attach postpositions (Joshi in Japanese)

Word order is represented by an augmented context free grammar. Postpositions which represent cases are usually given as literals in rules, since there are typical postpositions representing each specific case. For example, "ga" , "wo", and "e" represent subject, object, and goal respectively. But some verbals do not take these typical postpositions; for example, "suki-da" ("like") takes "ga" as an object case marker instead of "wo". For such special cases, the variable postposition mechanism is used. In the case of "ga" for object, a postposition is not given by a rule but from the case slot in the word structure of "like."

3.5 MORPHOLOGICAL GENERATION

Most information for morphological generation is included in the dictionaries. For example, one of the translations for "write" is "yo-mu". This is an infinitive form and the following information is needed to get the conjugated form:

1) Stem: "yo"
2) Conjugation type: "5-dan"
3) Kind of conjugational part: "ma"
4) Other information (such as an irregular form in special use): "onbin-kei"

Information which is got during syntactic analysis, such as tense, aspect, modality and so on, is attached to the head verb as morphological information. Figure 3-7 gives *a* rough sketch of the above process.

```
INPUT:               I could not go.


Morphological      [SW:I] [SW:can TENSE:past] [SW:not] [SW:be]
Analysis          [SW:go  TENSE:present]


Syntactic          [SW:I] [SW:go  MODALITY:(can (TENSE:past))
Analysis                    TYPE: negation]]
```

Figure 3-7 GENERATION OF MORPHOLOGICAL INFORMATION

4. CONCLUSIONS

Brief explanation of TAURAS is presented. The points of the system features are as follows:

1) Semantics with low computation cost and high performance is introduced, which is a new semantico-syntactic approach.
2) TAURAS has been developed as a total translation system on a engineering workstation.

REFERENCES

Alain Colmerauer(1970).'LES SYSTEMES-Q OU UN FORMALISME POUR ANALYSER ET SYNTHETISER DES PHRASES SUR ORDINATEUR', TAUM, Universite de Montreal

W.A.Woods(1970).'Transition Network Grammar for Natural Language Analysis',Communication of the ACM Vol.13, No.10

J.Chauche(1972).'ARBORESCENCES ET TRANSFORMATIONS', GETA, Universite Scientifique et Medicale de Grenoble

Gilles Stewart(1975).'Le language de programmation REZO', Department d'Informatique Faculte des Arts et des Sciences, Universite de Montreal

John Chandioux(1976).'METEO: un systeme operationnel pour la traduction automatisee des bulletins meteorologiques', META, Vol.21, No.2 pp.127-133

Bernard Vauquois(1977).'L'EVOLUTION DES LOGICIELS ET MODELES LINGUISTIQUES POUR LA TRADUCTION AUTOMATISEE', GETA, Universite Scientifique et Medicale de Grenoble

John Lehrberger(1978).'Automatic Translation and the Concept of Sublanguage', TAUM, Universite de Montreal

Isabelle,P., Bourbeau,L., Chevalier,M., Lepage,S.(1978). 'Description d'un systeme de traduction automatisee des manuels d'entretien en aeronautique, TAUM, Universite de Montreal

William Woods(1980).'CASCADED ATN GRAMMARS', American Journal of Computational Linguistics, Vol.6,No.1

Jonathan Slocum (1981).'THE METAL PARSING SYSTEM',Linguistic Research Center, The University of Texas

Jun-ichi Tsujii, Jun-ichi Nakamura, Makato Nagao(1984).'Analysis Grammar of Japanese in the Mu-Project - A Procedural Approach to Analysis Grammar',Proceedings of COLING84

Makoto Nagao, Toyoaki Nishida, Jun-ichi Tsujii(1984).Dealing with Incompleteness of Linguistic Knowledge on Language Translation - Transfer and Generation Stage of a Mu Machine Translation Project',Proceedings of COLING84