COLING 82, J. Horecký (ed.)
North-Holland Publishing Company
© Academia. 1982

AN ENGLISH-JAPANESE MACHINE TRANSLATION SYSTEM BASED ON FORMAL SEMANTICS OF NATURAL LANGUAGE

Toyo-aki NISHIDA and Shuji DOSHITA

Department of Information Science
Faculty of Engineering, Kyoto University
Sakyo-ku, Kyoto, 606, JAPAN

This paper proposes a new model of machine translation. In this model, the lambda formula obtained from the syntactic and semantic analysis of a source language sentence is viewed as a target language generating function and the target language sentence is obtained as a result of evaluating the formula by functional application or λ-calculus. This model provides a systematic and powerful way of incorporating human knowledge on the languages. A prototype is constructed on the LISP system. The performance was tested for four sample texts taken from existing technical reports and computer manuals.

INTRODUCTION

This paper proposes a functional model of machine translation and describes its applcation to English-Japanese machine translation. In this model, we aimed to achieve:

- systematization of translation process,
- lexicon based autonomous framework, and
- a translation model based on semantic interpretation.

INTERMEDIATE REPRESENTATION

Intermediate representation of this model is EFR (English-oriented Formal Representation) and CPS (Conceptual Phrase Structure).

EFR is a logical language based on Cresswell's lambda categorial language (Cresswell (1973)), which can be considered to be a notationally simplified version of Montague Grammar (Montague (1974), Dowty (1981)). From an engineering point of view, EFR can be regarded as an artificial language in which each expression is unambiguous. So, there may be the cases in which more than one EFR expression can be associated with a given sentence. In such cases, ambiguities are resolved using inference, knowledge, or by human assistance.

CPS is an extended phrase structure in that (1) CPS is a more general element including syntactic knowledge on the concept, so (2) CPS is implemented as a frame and (3) CPS is not only a data structure which is an object under operation but also a function which can operate on other CPS's.

A CPS formula is a functional notation (lambda formula) of the operation sequence on CPS's. A CPS formula is evaluated to be a CPS or a functional value. The evaluation process is defined by a (pure) LISP like interpreter.

SOURCE LANGUAGE ANALYSIS

English sentence analysis is done using two layered rules, pattern directed augmented context free rules (AUGCF rules) and production type procedural rules. AUGCF rule is a descriptive rule. Context free rule is extended in several points, (1) attached function for checking syntactic details and semantic acceptability, (2) direct notation of gap in relative clauses or interrogative sentences. An AUGCF rule describes what EFR formula is associated with a given syntactic pattern and in what condition the pattern is acceptable. Some examples look like:

$$S \xrightarrow{\text{subjvp},+10,*\text{sem}_1(*\text{sem}_2)} NP.VP \quad \ldots \text{ R1}$$

$$NP \xrightarrow{\text{pm}_1,+0,\text{cons-np-rel}} NP.\text{"WHICH"}.(S\text{-}NP) \quad \ldots \text{ R2}$$

Although lots of syntactic phenomena can be easily formalized with AUGCF rules, the computer cannot efficiently analyze input sentences only with them. One reason is that the computer must examine which rules are applicable in a given situation and determine which one is plausible. Such processings make the computer very much slow and inefficient. Another reason is that some kind of heuristic knowledge, which is sometimes referred to as knowledge on control (Davis (1980)), cannot be effectively incorporated into the AUGCF rules. The knowledge on control provides heuristics on when and how to use each rule. Condition -> action formalism (production rule formalism) is considered to be suitable to write such level of knowledge.

Our second level rule is obtained by attaching control information to each AUGCF rule and transforming the rule format. The type of procedural rules are: E-rule, U-rule, B-rule, and L-rule.

- E-rule (expansion rule) is invoked when a goal is expected. E-rule specifies subgoal decomposition of the given goal.
- U-rule (up-ped rule) is invoked when a parse tree node is generated. This rule further specifies additional goals and if all of them succeed, a new node will be constructed. This rule is used mainly for left recursive type AUGCF rules.
- B-rule (Bottom-up rule) is referred to by a bottom-up parser incorporated in the rule interpreter.
- L-rule (Lexicon rule) is embedded in a dictionary and invoked when a key word is encountered in the given text.

The rules R1 and R2 are rewritten into procedural type rules as follows:

$$\text{goal=S} \Rightarrow \{T \rightarrow \text{expand}[(\text{NP VP});\text{subjvp};+10;*\text{sem}_1(*\text{sem}_2)]\} \quad \ldots \text{ R1}' \quad \text{(E-rule)}$$

$$\text{constructed=NP} \Rightarrow \left\{ ?\text{lex}[\text{"WHICH"}] \rightarrow \begin{array}{l} \textit{"set the next goal an S with} \\ \textit{exactly one NP deleted;} \\ \textit{if it succeeds, then apply R2."} \end{array} \right\} \ldots \text{ R2}' \quad \text{(U-rule)}$$

Where R1', for example, says that: given a goal S then expand it into subgoals NP and VP; if both of them succeed then reduce them into an S node; at that time, a function subjvp checks subject-verb agreement; +10 is the score for S; $*\text{sem}_1(*\text{sem}_2)$ is a pattern of the EFR expression for the S node, where $*\text{sem}_1$ denotes the EFR expression for its first son (NP), etc. If some anomaly is detected by those functional attachments, the application of the rule is rejected (functional augmentation of CF rule).

A notion of a frame is employed in order to implement feature semantics. A frame is an extended property list in which syntactic and semantic features are described. By passing and checking consistency among such features, (mainly semantic) constraints are implemented.

In practice, the knowledge incorporated in a system can never be total and complete, so human being should help computer analyze input sentences. The human halp is limited to resolving ambiguities. In order to make the human diagnosis efficient, some diagnostic facilities are implemented.

It is also important to construct and manage dictionaries. Dictionary manager is implemented to make human modification of dictionary flexible by use of pattern directed dictionary editing commands.

INTERPRETATION OF EFR AND TARGET LANGUAGE GENERATION

The interpretation of an EFR expression can be defined in the conceptual level. For example, given an EFR expression:

a(λy[a*(communication))(λx[(((*ap(for)(x))(facility))(y)])]),

which corresponds to a noun phrase "a facility for communication". A detailed description of the conceptual interpretation in our conceptual model (Nishida (1980)) is given below.

(1) conceptual interpretation of a(λy[...]) associates a conceptual element "something" (individual concept) with the variable y.
(2) conceptual interpretation of a*(communication)(λx[...]) associates a conceptual element "(a) communication" with the variable x.
(3) (*ap(for))(x) is interpreted as an adjective concept "for the sake of x", which becomes "for the sake of (a) communication" from (2).
(4) the adjective concept obtained in (3) is applied as a function to the interpretation of "facility" (i.e., a noun concept "facility"). Thus we obtain a complex noun concept "system for the sake of (a) facility" for ((*ap(for))(x))(facility).
(5) the application of a noun concept p to an individual concept q yields a sentence concept: "q is a p." This interpretation rule is used for the fragment: (((*ap(for))(x))(facility))(y). The result is a sentence concept: "something (y) is a facility for the sake of (a) communication."
(6) Finally the interpretation of a given EFR expression results in a noun phrase concept: "something y: such that y is a facility for the sake of (a) communication." This noun phrase concept is a higher order concept which gives a name to an individual: "a facility for the sake of (a) communication." This higher order concept will be reduced if it is applied to a one place predicate (roughly speaking, a property like "being constructed", "being an x such that the paper is concerned with x", etc.).

The above process of interpretation is stepwise and includes no "gap" nor "skip". Such property is crucially important in constructing large and complex systems including machine translation systems. This process can be simulated in the "linguistic" domain; our idea of target language generation is this:

- each conceptual element is accompanied with a target language phrase structure which gives the name of the concept.
- each semantic interpretation of a complex structure is accompanied with a syntactic operation of creating new phrase structure from those for function part and argument part conceptual elements.

Two types of Japanese phrase structure manipulating rule can be associated with functional application:

- embedding one phrase into another phrase as a modification part (generate KAKARI-UKE relation)
- transforming one phrase by use of the information from another phrase.

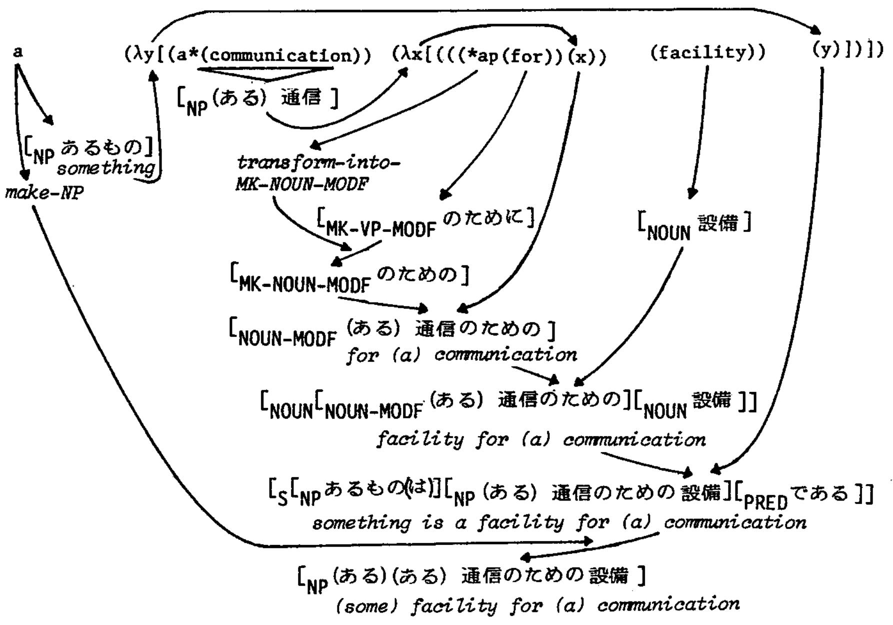
a
(λy[(a*(communication)) (λx[(((*ap(for))(x)) (facility)) (y)])])
[NP (ある) 通信]
[NP あるもの] something
make-NP
transform-into-MK-NOUN-MODF
[MK-VP-MODF のために]
[MK-NOUN-MODF のための]
[NOUN 設備]
[NOUN-MODF (ある) 通信のための] for (a) communication
[NOUN[NOUN-MODF (ある) 通信のための][NOUN 設備]] facility for (a) communication
[S[NP あるもの(は)][NP (ある) 通信のための 設備][PRED である]] something is a facility for (a) communication
[NP (ある) (ある) 通信のための 設備] (some) facility for (a) communication

Fig.1. Outline of a sample generation from an EFR expression.

Thus, a functional application corresponds to a primitive syntactic operation of Japanese language.

CPS is defined to be a structure which conveys not only conceptual information on a concept but also syntactic information about the concept. All those information is structured as a frame. The descendant slot of a CPS is either a terminal value (lexicon frame) or a list of CPS's. Thus CPS can be linked as a tree structure. A CPS corresponding to a noun phrase: "the typewriter" looks like:

$[_{NP}\ [_{DET}$ 'the' with Q=DEFINITE]
$[_{NOUN}$ 'typewriter' with CLASS=PHYSOBJ ...] with NBR=SGL ...].

A CPS works both as a data and as a function; it is sometimes applied to other CPS's to result in another CPS or functional value, or it sometimes is a data structure under some operation. Thus CPS is a higher order object. The semantics can be modeled in the notion of a categorial grammar. A CPS of an adjective concept, for example, maps a CPS of a noun concept into another (compound) CPS of a modified noun. This principle can be written as: ADJ=NOUN/NOUN. On the other hand, the adjective CPS can be modified by an adverbial CPS. Thus ADV=ADJ/ADJ.

A CPS formula specifies a sequence of operations on given CPS's. A CPS formula involves CPS as a data. Other elements of CPS formula are: variable (with coersion specification), lambda expression, functional application formula, transformational rules, conditional expression, and composition function. The evaluation process of a CPS formula is defined as a function like LISP interpreter.

Fig.1 illustrates an outline of target language generation process for a phrase "a facility for communication". (CPS formula is ommited there.)

In practice, our system involves one step called the REFORM step after the CPS evaluation process. This step is needed mainly because, (1) some direct output is not readable; the content can be understood without ambiguity, but it is much redundant or not commonly used, or much more worse (2) the output is semantically wrong. Such cases arises where the EFR expression extracted from the source language is not well defined to the language expression in question. This case occurs when the system designer commits misconception or fails to correctly capture the phenomenon. In principle, the second case is obviously bad but no theory has ever succeeded in modelling all phenomena in natural language. So in practice, the second case is unavoidable.

The REFORM process uses heuristic rules to 'reform' those CPS structure into reasonable one. Pattern directed transformation rules are used. Those rules are applied until no rule is applicable to the given CPS structure.

EXPERIMENTS

A prototype of the system has been constructed on a personal LISP system (Doshita (1978)), which is developed on a minicomputer with LISP-oriented storage subsystem. As to the analysis module, sixth version is in use; as to the generation module, first version is in use. About two years since the last COLING conference at Tokyo were mainly devoted to the development.

At the first stage of experiment, sample sentences were tested for several sentence patterns. At the second stage, our purpose was to extend the system for practical test; to translate existing texts even if introducing human assists to some (reasonable) extent. Four sample texts (totally 40 sentences) selected from existing technical reports and computer manuals. Each of the s ple texts orresponds to one section or a short chapter in the material. All s ences of each sample texts have been successfully translated into Japanese. No pre-editing is done except for three minor modifications to the original text (e.g., "16- or 32- bit" => "16-bit or 32-bit"). Human assist is limited to resolving ambiguities in the analysis phase. One example is shown in Fig.2.

CONCLUSION

This paper proposes a new approach to machine translation based on a functional semantics of natural langauge. The effectiveness of this approach is tested by experiments for short chapters and an abstract taken from existing technical reports and computer manuals.

ACKNOWLEDGEMENT

This research was partially supported by Grant-in-Aid for Scientific Research. The authors want to thank Mr. Kiyoshi Agusa and Mr. Shigeo Sugimoto for providing conveniences of editing and printing this material.

REFERENCES

[1] Cresswell, M.J., Logics and Languages, (Methuen, 1973).
[2] Davis, R., Meta-rules: reasoning about control, AI 15 (1980), 179-222.
[3] Doshita, S., Hiramatsu, K., and Kakui, K., Implementation of LISP system

using direct accessible bulk memory, in: Trans. of IECE of Japan, vol. J61-D, no. 5, (1978), (in Japanese).
[4] Dowty, D., Wall, R., and Peters, JR., Introduction to Montague Semantics, (Reidel, 1981).
[5] Montague, R., Proper treatment of quantification in ordinary English, in: Thomason (ed.), Formal Philosophy, (Yale University, 1974).
[6] Nishida, T. and Doshita, S., Hierarchical meaning representation and analysis of natural language documents, in: Proc. COLING 80, (1980), 85-92.

```
    ETHERNET IS A BRANCHING BROADCAST COMMUNICATION SYSTEM FOR CARRYING
(1) DIGITAL DATA PACKETS AMONG LOCALLY DISTRIBUTED COMPUTING STATIONS /
     ETHERNETはデジタルデータパケットを局所的に分散される計算ステーションの間
     に伝搬するための分岐状の放送通信システムである。
                    ——— ——— ——— ——— ———

(2)  THE PACKET TRANSPORT MECHANISM PROVIDED BY ETHERNET HAS BEEN USED TO
    BUILD SYSTEMS WHICH CAN BE VIEWED AS EITHER LOCAL COMPUTER NETWORKS OR
    LOOSELY COUPLED MULTIPROCESSORS /
     ETHERNETによって提供されるパケット輸送機構は局所的な計算機ネットワークま
     たは疎に結合されるマルチプロセッサとみなされることができるシステムを構築するため
     に用いられてしまう。
                    ————————————————————

(3)  AN ETHERNET'S SHARED COMMUNICATION FACILITY * ITS ETHER * IS A PASSIVE
    BROADCAST MEDIUM WITH NO CENTRAL CONTROL /
     ETHERNETの共有された通信設備（そのETHER）は中央の制御を持たない受動
     性の放送媒体である。
                    ——— ——— ——— ——— ———

(4)  COORDINATION OF ACCESS TO THE ETHER FOR PACKET BROADCASTS IS
    DISTRIBUTED AMONG THE CONTENDING TRANSMITTING STATIONS USING CONTROLLED
    STATISTICAL ARBITRATION /
     パケット放送のためのそのETHERへのアクセスの調整は制御される統計的調停を用い
     てその競合している送信側のステーションに分散される。
                    ——— ——— ——— ——— ———

(5)  SWITCHING OF PACKETS TO THEIR DESTINATIONS ON THE ETHER IS DISTRIBUTED
    AMONG THE RECEIVING STATIONS USING PACKET ADDRESS RECOGNITION /
     そのETHERの上の目的地へのパケットの交換はパケットアドレス認識を用いてその受
     信側のステーションの間に分散される。
                    ————————————————————

(6)  DESIGN PRINCIPLES AND IMPLEMENTATION ARE DESCRIBED BASED ON EXPERIENCE
    WITH AN OPERATING ETHERNET OF 100 NODES ALONG A KILOMETER OF COAXIAL
    CABLE /
     設計原則と実働化は1Kmのコアキシャルケーブルに沿った100のノードの稼動中のE
     THERNETについての経験に基づいて述べられる。
                    ——— ——— ——— ——— ———

(7)  A MODEL FOR ESTIMATING PERFORMANCE UNDER HEAVY LOADS AND A PACKET
    PROTOCOL FOR ERROR-CONTROLLED COMMUNICATIONS ARE INCLUDED FOR
    COMPLETENESS /
     重い負荷の下での性能を評価するためのモデルと誤り制御された通信のためのパケットプ
     ロトコルは完全のために含まれる。
```

Fig.2. Translation of a sample text: Metcalfe, R.M. and Boggs, D.R., Ethernet: distributed packet switching for local computer networks, CSL-75-7, Xerox Palo Alto Res. Centr., (1980), (ABSTRACT).
Online print out of the system is shown. --- separates sentences and —— separates paragraphs. Underlined are bad (-) or wrong (=) results.