ATLAS II:

A Machine Translation System Using Conceptual Structure as an Interlingua

Hiroshi Uchida
Natural Language Processing Section
Software Laboratory, Fujitsu Laboratories, Ltd.
Kamikodanaka 1015, Nakahara-ku
Kawasaki-shi 211, Japan

ATLAS II is a semantic-based machine translation system which aims at high quality multilingual translation. In order to develop a system which deals with various languages with a high degree of precision, analysis and generation mechanisms must be independent of any language, and linguistic knowledge of one language must be independent of other languages. Therefore, we adopt the interlingua approach, which uses conceptual structure as an interlingua, and develop a language-independent processing method, with a language-independent dictionary structure. In this paper, we present the ATLAS II translation mechanism, emphasizing the processing method, and explain what kind of knowledge is used for translation.

1. Introduction

In 1984, Fujitsu marketed the machine translation systems ATLAS-I and ATLAS II. ATLAS-I was the world's first commercial English-Japanese translation system. Fujitsu is also conducting joint research and development of a Japanese-Korean machine translation system based on ATLAS-I's architecture, in cooperation with the Korean Advanced Institute of Science and Technology (KAIST).

ATLAS II aims at multilingual translation. At present, the commercial version of ATLAS II translates Japanese to English. However, some effort has been directed toward achieving multilingual translation. From 1983 to 1985, Fujitsu contributed technical support to the SEMSYN project, a Japanese-German translation system being developed at Stuttgart University, West Germany. At Tsukuba EXPO '85, we also conducted machine translation experiments, translating Japanese children's compositions into English, French and German, English news texts into Japanese, French and German, and bidirectionally translating simple sentences between Japanese, English, Swahili and Innuit (Eskimo).

2. ATLAS II System

ATLAS II aims to simulate human translation, understanding a sentence written in one language, then expressing it in another. Any language is based on the assumption that every person is able to understand a sentence from the meaning of the component words and context. Syntax rules are also based on this assumption. To be able to translate naturally, a computer should also be able to do this.

In order for humans and computers to understand text written in natural language, it is necessary to know the meaning of words and the meaning within the contexts they are used. An entry in the word dictionary of ATLAS II contains the concepts expressed by a word and grammatical characteristics of the word when it expresses a concept. In the world model of ATLAS II, the knowledge necessary for understanding the concept is written in a form understandable by the computer, called conceptual structure. The information necessary for understanding the use of words is provided in the form of grammar rules.

Fig. 1 shows the translation process of ATLAS II. Source language text is analyzed using the word dictionary, analysis rules and world model. The result is expressed as a conceptual structure, which is the interlingua of ATLAS II. From the conceptual structure, target language text is generated using the word dictionary, generation rules and language model. If necessary, the conceptual structure is converted to another conceptual structure to fit the target language speaker's way of thinking.

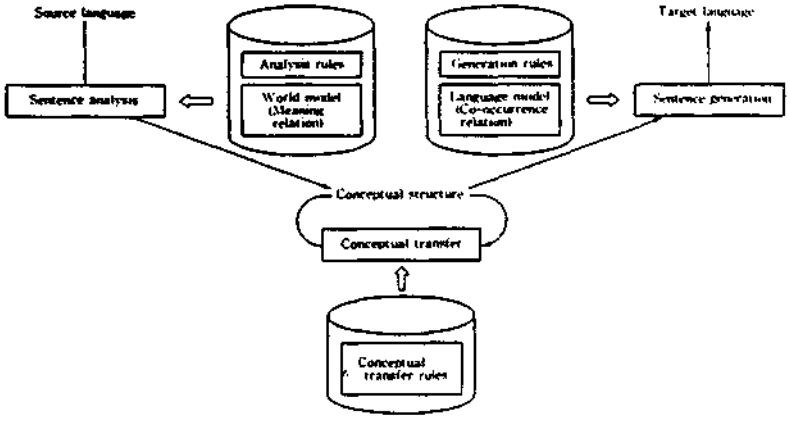


Fig. 1: Translation process of ATLAS II

3. Interlingua and the World Model

The conceptual structure, which is the interlingua of ATLAS II, is expressed by a set of binary relations between concepts and features attached to concepts. This is a semantic network representation of an input sentence. Fig. 2 shows the conceptual structure equivalent to "I bought a new car." The network consists of nodes and arcs. A node denotes a concept representing one of the meaning of the words "I", "buy", "car", "new". Arcs denote the deep case relations such as AGENT, OBJECT, and causal relations such as CAUSE. In addition to the above binary arcs, there are unary arcs which indicate a feature of a concept such as tense and style, etc. In Fig. 2, PAST indicates tense and ST indicates focus.

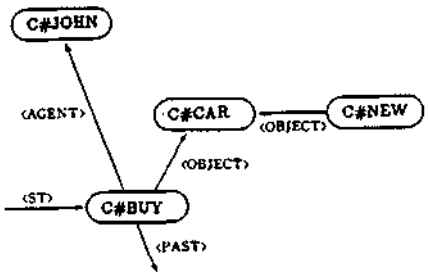


Figure 2: Conceptual structure for "John bought a new car"

In the same way as humans use their knowledge when understanding a sentence, ATLAS II refers to its world model when translating a sentence into the interlingua. The world model defines every probable relation between concepts. In other words, the world model contains every conceptual structure for every meaningful sentence. For example, the concept "birds fly" is expressed by the binary relation (C#BIRD, C#FLY, C#AGENT). The concept "birds fly with wings" is expressed by the two binary relations (C#BIRD, C#FLY, C#AGENT) and (C#WING, C#FLY, C#INSTRUMENT). If the conceptual structure of the input sentence is included in the world model, the system accepts it; if it is not, the system rejects it and asks for another sentence analysis.

The vocabulary of the interlingua consists of concepts and relations. Relations between concepts should be as universal as possible. But this universality does not apply to all concepts, because each language has a number of unique concepts. These unique concepts are included as interlingua vocabularies. Some of these unique concepts can be expressed by other concepts in a conceptual structure. If the result of analysis contains such a concept, conceptual transfer is performed by using the correspondence between concepts and conceptual structures in the generation process.

There are two reasons why we adopt the interlingua approach. First, the interlingua interface completely separates analysis and generation, enabling the development of analysis and generation systems for one language to proceed independently from those of other languages. Developers of these systems need only know the interlingua and the language being analyzed or generated. Second, the interlingua allows the common use of knowledge. World knowledge is needed in semantic analysis, which is essential for high quality machine translation. Knowledge described in interlingua may be used by the analysis systems for each language.

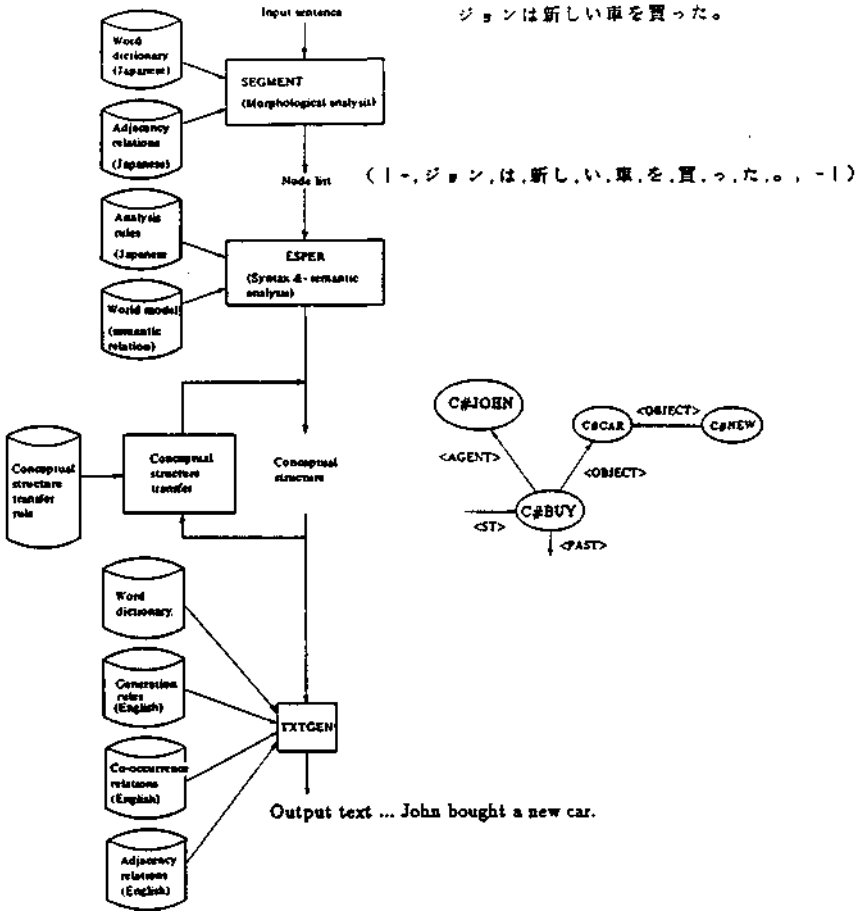


Fig. 3: Translation Flow of ATLAS II

4. Sentence Analysis

The sentence analysis phase analyzes an input sentence and produces a representation of its meaning in interlingua. This phase consists of two modules: SEGMENT for morphological analysis; ESPER for syntactic and semantic analysis. This phase uses the word dictionary, word adjacency relations, source language analysis rules and a world model which defines probable semantic relations between concepts. Fig. 3 show how each module uses the dictionaries and rules, and the output format.

SEGMENT extracts words (morphemes) from the input sentence and produces a node list for analysis. ESPER receives the node list and performs syntactic and semantic analysis. The result is expressed as conceptual structure. This result is checked for inclusion in the world model. If not included, ESPER selects another alternative and generates another result.

4.1 Morphological Analysis

An input sentence is first divided into morphemes. English sentences have spaces between words; in Japanese there is no clear boundary. SEGMENT performs a morphological analysis using the word dictionary and adjacency relations.

Morphological analysis is often thought to be highly language-dependent. This system, however, adopts a language-independent method for multilingual translation.

Starting at the left of the input string, every corresponding morpheme is taken from the word dictionary, and is checked whether it can be adjacent to the leftmost morpheme by referring to the adjacency relations. If it can be, the selected morpheme is removed from the input string and the next matching is performed until no further morphemes are found. Matching is based on the length of the morpheme and the frequency of its appearance. The longest, most frequently appearing morpheme is chosen first. If some strings remain unmatched, the system backtracks to construct an acceptable morpheme list.

Morphemes extracted from the input string are output in an analysis node list. ESPER receives this node list and each morpheme is treated as a terminal node. The sequence of nodes is the same as that of the input morphemes. Each node has been assigned grammatical and semantic information from the word dictionary. Grammatical information is a set of grammatical attributes. Each terminal node contains the most probable word of several candidates.

4.2 Syntactic and Semantic Analysis

Syntactic structure must be analyzed to understand an input sentence. Syntactic analysis requires determining the connection between elements of the sentence and the role of each element.

ESPER receives a node list from SEGMENT and performs simultaneous syntactic and semantic analyses using analysis rules based mainly on context-free grammar. ESPER consists of a status stack, analysis window, and control section. The status stack monitors the status during analysis; the analysis window views two adjacent nodes.

The general format of an analysis rule is:

<CONDITION> <GRAM1> + <GRAM2> =
<GRAM3> <TYPE> <RELATION> <ACTION> <PRIORITY>

CONDITION indicates the conditions under which this rule is applied. **GRAM1**, **GRAM2**, **GRAM3** are sets of grammatical attributes. **TYPE** is one of twelve rules. RELATION is a modifying relation between the two nodes. **ACTION** indicates the status after this rule is applied. **PRIORITY** determines which rule will be applied first when more than one rule can be applied.

At first, the analysis window is set on the first and second node, with the status stack empty, as shown in Fig. 4. ESPER finds an appropriate rule by referring to the two nodes and the status stack. ESPER checks if all of the symbols in the condition field of the rule are present on the status stack. If they are, ESPER checks if all of the grammatical attributes in the **GRAM1** and **GRAM2** fields of the rule are present in the analysis windows of the first and second node, respectively. The rule is selected if the condition is satisfied and grammatical attributes are present. When more than one applicable rule is selected, the rule with the highest priority is applied. There are twelve types of analysis rules, as shown in Fig. *5*.

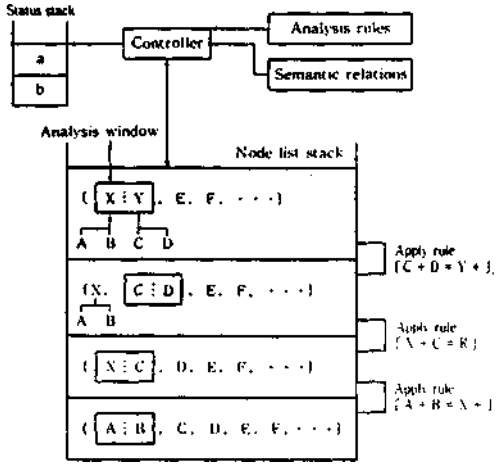
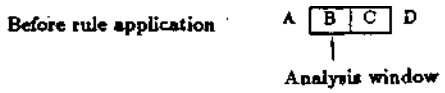


Fig. 4: Configuration of ESPER



Types of rules	Sub-tree generated	Semantics of nodes generated	Sub-structure generated	Window position after rule application
+Composition (+)	E; B C; b ⟨c⟩	b , ⟨ c ⟩	b ⟨r⟩	(A E , D); B C
-Composition (−)	E; B C; ⟨b⟩ c	c , ⟨ b ⟩	⟨r⟩ c	(A E , D); B C
Right modification (⟶)	E; B C; b,⟨r⟩ c	c	b ⟨r⟩ c	(A E , D); B C
Left modification (⟵)	E; B C; b⟨r⟩, c	b	c ⟨r⟩ b	(A E , D); B C
Forward read (F)	E; B; b	b	—	(A E , C, D); B
Backward read (B)	E; C; c	c	—	(A, B E , D); C
Right shift (R)	—	—	—	(A, B, C D)
Left shift (L)	—	—	—	(A B , C, D)
Exchange (X)	C B	—	—	(A C , B, D)
Copy (C)	B B C	—	—	(A B , B, C)
Back track (?)	—	—	—	—
Word change (!)	B C	—	—	(A, B C , D)

Fig. 5: Types of analysis rules

When an analysis rule is applied, a node created by combining the first and second nodes becomes the root node of a syntactic sub-tree. GRAM3 indicates grammatical attributes for the node, where attributes of the previous (i.e. first and second) nodes may be inherited and new attributes may be added. The analysis window moves down the node list to apply rules until the analysis tree is completed. If no applicable rule is found, ESPER backtracks and returns to the most recently applied rule to find an alternative.

ESPER performs syntactic and semantic processing simultaneously. A conceptual sub-structure corresponding to the syntactic sub-tree produced by a rule is generated when the rule is applied. The semantic correctness of syntactic processing is verified by checking whether the conceptual sub-structure is included in the world model or not. When the analysis tree is completed, the entire conceptual structure is again checked against the world model. ESPER backtracks if it is incorrect.

5. Sentence Generation

The target text is generated from the conceptual structure. The two-dimensional network is converted to a one-dimensional character string. The generation system traverses the network and outputs morphemes in the order it visits each node of the network. The order of traversal is specified by the generation rules, and morphemes are selected by referring to adjacency relations and co-occurrence relations. This mechanism can deal with both syntactic structuring and morphological synthesis at the same time, and is language-independent. Sentence generation is divided into two phases: transfer and generation.

5.1 Transfer Phase

The transfer phase fills the gap between interlingua and the target language. Differences in languages stem from the cultural background of the people speaking these languages. Superficially, they appear as a difference in words and grammar; internally, they appear as a difference in concepts and in the speaker's way of thinking. If concepts in conceptual structure are not of the target language or the same meaning is expressed by other concepts, the conceptual structure is transferred.

We illustrate some cases which require such a transfer. For example, the Japanese sentence *"Heya niwa mado ga futatsu aru"* would be literally translated into English as "There are two windows in this room." But the natural translation is "This room has two windows." In Japanese, the concept "exist" is used, but in English, the concept "possess" is used.

The general format of a transfer rule is:

(PartialNetl, PartialNet2, Relation, Condition)

This rule replaces **PartialNetl** by **PartialNet2** if both **Relation** and **Condition** are satisfied.

5.2 Generation Phase

The generation system consists of a generation window, output list and a rule interpreter. The rule interpreter traverses each node of the conceptual structure by moving the generation window and returns the output list of the translation results. Fig. 6 shows the generation mechanism, which consists of generation rules, word dictionary, co-occurrence relations and adjacency relations.

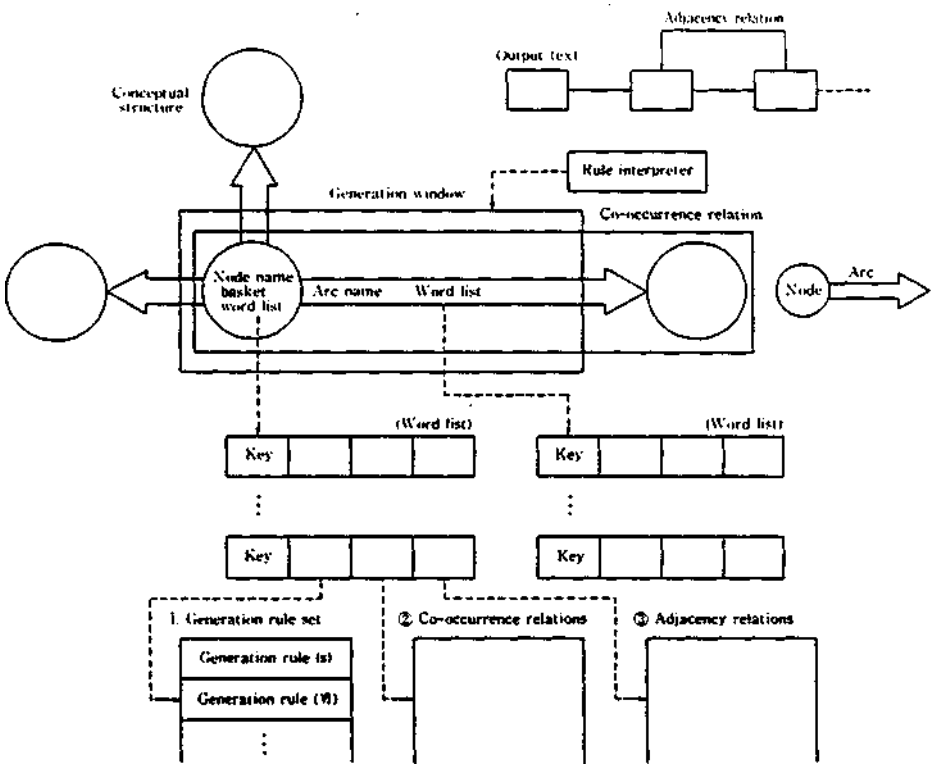


Fig. 6: Generation mechanism

The generation window is set at a node of the conceptual structure to see the node and arcs. The output list stores each word in order of generation.

A node of the conceptual structure consists of a node name, a basket and a word list. The node name indicates a semantic symbol. The basket stores messages sent from the node itself or from other nodes. The word list is a list of words which express the concept of the node.

An arc of the conceptual structure consists of an arc name and word list. The arc name indicates a relation between nodes. The word list is a list of words which represents the relation between nodes. Both the node and arc name are keys to retrieve words from the word dictionary. Word dictionary entries contain generation symbols which serve as keys to access a generation rule set.

The rule interpreter interprets each generation rule, traverses each node by moving the generation window, and selects words from nodes and arcs by checking the co-occurrence and adjacency relations. Each word selected is added to the output list.

Co-occurrence relations between two words define the boolean value of whether the two words can co-occur in the same sentence with a specified relation. In general, a concept may be expressed as several different words. Co-occurrence relations are used to select the most appropriate word.

Adjacency relations are used to select appropriate morphemes on the basis of whether two morphemes can be adjacent to each other.

A generation rule set is an ordered set of at least two generation rules. The order specifies the sequence of application, thus determining the word order of the output sentence.

The general format of a generation rule is as follows:

<CONDITION> <ARCNAME> <ACTION> <MESSAGE>

CONDITION indicates the conditions under which this rule is applied. **CONDITION** is checked against the messages in the BASKET. If they match, this rule is applied; if they do not, the next rule is tried. **ARCNAME** indicates an arc name to apply the rule. **ACTION** specifies the type of processing. The primary types of rules are as follows:

(1) Node generation rule for generating a word corresponding to the node.

(2) Out-arc generation rule for generating a phrase from a subnetwork starting at the specified out-arc.

(3) In-arc generation rule for generating a sentence from a subnetwork starting at the specified in-arc.

(4) Word generation rule for directly generating a word.

MESSAGE indicates message to be sent to the BASKET of the node itself or to nodes connected to the node with arcs.

The generation system receives a conceptual structure in which each node and arc has a corresponding word list. Sentence generation starts at a node with an in-arc <ST>.

6. Conclusion

We have analyzed and generated text in Japanese, English, French, German, Chinese, Swahili, and Innuit (Eskimo) using ATLAS II, with no software modifications. Therefore, we believe that the language-independent mechanism of ATLAS II is suited to multilingual translation.

ATLAS II translates sentence by sentence at present, but has a means of sending messages to the next sentence analysis. We plan to introduce context analysis and generation via this mechanism.

Translation quality presents the biggest problem to all machine translation systems. Unfortunately, current technology cannot produce perfect results, so post-editing is required. However, post-editing ATLAS II translations takes 30-50% less time than full manual translation. Thus, ATLAS II is time and cost-effective, even at the current level of technology.