Parsing with look-ahead in real-time on-line translation system

Hiroyasu NOGAMI, Yumiko YOSHIMURA and Shin-ya AMANO

Research and Development Center
Toshiba Corporation
1, Komukai Toshihacho, Saiwai-ku,
Kawasaki-City, Kanagawa, 210 Japan

Abstract

In order to increase parsing efficiency in a real-time on-line translation system interfaced with a keyboard conversation program, we have developed a version of the ATN formalism with a look-ahead function. By permitting future input to be scanned, arcs can be reordered or suppressed. Various mechanisms utilizing this capability are presented.

1. Introduction

A real-time on-line communication system including automatic translation was realized by combining a keyboard conversation function with an English-Japanese bi-directional machine translation system implemented on a workstation [Amano 1986, 1987]. Using a satellite connection, bilingual conversations were held between members of this laboratory in Japan and visitors to the 5th World Telecommunications Exhibition *Telecom 87*, organized by the International Telecommunication Union, held in Geneva from 20th to 27th October in 1987 [Amano 1988a, 1988b] [Miike 1988] [Takeda 1988] [Asahioka 1988].

The general set-up, the screen display and the system configuration are illustrated in Appendix. The system operates as follows: an operator in Switzerland types his/her message in English which is displayed in the upper window and is transmitted via standard telecommunications devices to Japan, where it is immediately translated into Japanese. The operator in Japan receives the message in Japanese in the lower window and types his/her response in Japanese. This is translated and then transmitted to Switzerland. The system is like UNIX's* 'talk', except that contributions to the dialogue appear in the appropriate language.

An important feature of such a real-time translation system is that translation time must be reduced to an absolute minimum so that the conversation can proceed naturally. To reduce the parsing time, this system uses a version of the ATN formalism with a look-ahead function, based on the concept of a parsing method using global information. This parsing method proved to be very effective during this experiment, so that we could communicate with visitors very naturally irrespective of time taken for satellite communication and time required in typing his/her messages.

Focusing on the parsing of English, the paper discusses the concept of parsing using global information in Section 2, the realization of the parsing method as an ATN in Section 3 and the conclusion in Section 4.

2. The concept of parsing using global information

In parsing natural languages using large-size dictionaries and grammars, there are usually multiple categorial and syntactic possibilities for the current word, when using only the information associated with this word. The parsing methods in [Woods 1970][Pereira 1980] use only the information of the current word, so that these methods waste much time trying a lot of possibilities which eventually prove to be failure.

Such possibilities, however, can be suppressed or reordered even at the current word position by using global surface information from the input without really parsing it. This notion is especially useful, given the following features of English syntax:

i) constituents consist of at least one obligatory element, e.g. a sentence requires a verb, a noun phrase requires a noun, etc.

ii) many structures involve discontinuities, e.g. as --- as, not --- but, the more --- the more, both --- and, etc.

In order to suppress or reorder the possibilities, these features are used in real parsing as follows. These features suggest that each rule of a CF grammar requires at least one obligatory terminal element, as well as optional terminal and non-terminal elements. By looking for the obligatory elements in the global input as the first step in applying a given rule, the rule can be rejected or reordered dynamically at the current word position. This search is of great significance, especially if there are many intervening optional elements and/or these are themselves rather complex.

This function further has the additional advantage of provisionally partitioning the input into approximate constituents delineated by the obligatory elements which act as "stepping stones" through the input "stream". This

*UNIX is a Trademark of AT&T Bell Laboratories

might be useful for parsing with parallel processors, and may have implications for cognitive psychology applications of parsing. However, we have not pursued these particular aspects.

3. Realization of parsing using global information as an ATN

3.1. Look ahead mechanisms

We have realized the "Parsing using global information" method as an ATN, which uses the topdown depth-first search method, reinforced with a "look ahead" condition. This condition checks the specified obligatory elements which each syntactic or categorial possibilities (denoted as arcs of ATN) require in the future input, from the current word position to the number specified or to the end of sentence as default.

The ATN with "look ahead" conditions works very effectively for (1) rejection of syntactic possibilities, (2) reordering of them, (3) rejection of categorial possibilities, and (4) reordering of them. These are best shown with some examples of linguistic phenomena.

(1) rejection of syntactic possibilities

a) rejection of one syntactic possibility

The general description for rejection is shown in Figure 1 with an example at the right-hand side. The general network of Figure 1 has the following meaning:

- At arc 'A', if the obligatory condition 'c' which arc 'A' requires at arc 'D' is not satisfied in the future input, then arc 'A' is rejected without really parsing from arc 'A' to arc 'D'. (i.e. useless parsing from arc 'A' to arc 'D' can be avoided.)

In parsing the example (a) without looking-ahead, "the lecture" can be analyzed as the subject of "that-clause". Eventually, this interpretation is rejected since there is no verb after "lecture" in (a). In this case, control returns to the previous backtrack point in the sub-network for "noun phrase", and then wastes still more time trying all the remaining possibilities which will eventually lead to failure. On the other hand, by looking-ahead for a "verb" in the future input, parsing "the lecture" as the subject of "that-clause" can be avoided.

In realistic, large-size ATN, states have a lot of arcs. For example, the first state of a sub-network for a noun phrase has a lot of arcs for determiner, pronoun, noun, adjective, present-participle, past-participle, adverb, possessive, prefix, negative, numeral, quantifier, intensive, interrogative, ---, and also for certain particular words such as "that", "so", "both", "such", "from" (ex. from 2 to 5 inch), etc. The number of arcs from this state alone is generally greater than the number of words in the average input sentence. And what's the next state also has lots of arcs. Therefore, even in parsing a noun phrase we must check an enormous number of arcs compared with the look-ahead checks (see 3.2).

The look-ahead function works more effectively as the length and the complexity of input sentences increase. For example, the look-ahead is very useful if "lecture" has many coordinands and is modified by many optional elements such as a prepositional phrase, a participle phrase, etc. It is also useful for long-distance discontinuities, e.g. identification of phrasal verbs, or constructions such as "so adj. --- that", "it[formal subj] --- that[logical subj]", "not --- but", "the more --- the more", "either --- or", etc.

Compare the sentence "I heard the lecture and then had lunch." to the sentences of Figure 1. Looking-ahead for "verb" beyond "and" in the future input is of no effect since "had" isn't an element within "that-clause". Therefore, a boundary condition which can terminate the look-ahead search of "verb" at "and" can be considered.

However, it is very difficult to give such conditions without real parsing, and such conditions might be rather complex and time-consuming. For this reason, we use only simple look-ahead conditions without considering search boundary. Therefore, these conditions generally check categories or words of the future input to the end of the sentence (cf. (2)(4)). Note that such a simple check of a word in the future input takes about the same time as the traversal of one arc (see 3.2), and only conditions which never cause any mistakes beyond the search boundary must be written. Fortunately it is not difficult, because almost of all look-aheads work as an existential check of word or word sequence, which assures fail safe property. For example, the look-ahead condition shown in Figure 1 never causes any mistakes.

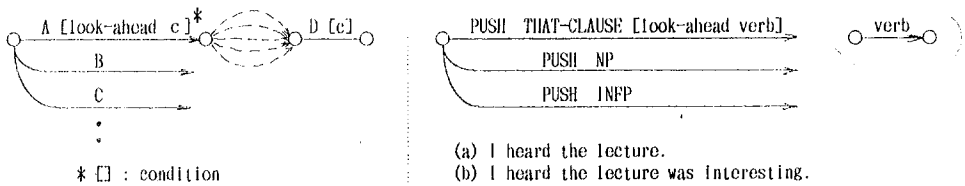


Figure 1 Rejection of one syntactic possibility

b) rejection of all possibilities which require the same obligatory element

In this case, a "fail arc" is very useful, as shown in Figure 2. The general network of Figure 2 has the following meaning:

- At fail arc 'D', if the condition 'c' is satisfied in the future input, then sister arcs 'E', 'F', --- are pruned (i.e. the current backtrack point is released), control returns to the previous backtrack point and arc 'B' is tried. In this way, useless parsing of sister arcs can be avoided.

In practice, fail arcs often have negation of conditions which apply to sister arcs. In the example (a), parsing "repairs" as the subject of a coordinated S can be avoided. This alternative is useful, especially when there are many arcs at a state: rather than check repeatedly the condition on each, a fail arc with the negation of the condition leads us back to the previous state.

c) rejection of all but one possibility

In this case, an "anchor arc" is very useful, as shown in Figure 3. The general network of Figure 3 has the following meaning:

- At anchor arc 'C', if the condition 'c' is satisfied in the future input, arc 'F' is tried and sister arcs 'D', 'E', --- are pruned, irrespective of the ultimate success or failure of parsing after arc 'F'. Otherwise, the next arc 'D' is tried.

Anchor arcs, like fail arcs, often have negative conditions. Consider the text segment in the example (a): if there is not another candidate imperative verb in the input, analysis of "turn on the power" as a coordinand of "start the machine" can be rejected (cf. example (b)).

Also consider parsing "if NP1 vt NP2, NP3 vt NP4." By looking-ahead, parsing NP3 as a coordinated NP of NP2, and parsing "NP3 vt NP4" as a coordinated S of "NP1 vt NP2" can be avoided. Note that all possibilities of a noun phrase must be checked in both cases before the success.

d) rejection of the input sentence

In this case, a "stop arc" is used, as shown in Figure 4, which has the following meaning:

- At stop arc 'A', if the condition 'c' is satisfied in the future input, the parsing of the whole sentence is stopped irrespective of the midst, because this sentence has no possibilities of success.

In the example, useless parsing of ungrammatical sentence which has no verbs after relative pronoun by typing errors can be avoided. This is used for ungrammatical sentences including undefined words, typing errors, etc. It is particular useful when false paths would otherwise be a costly detour.

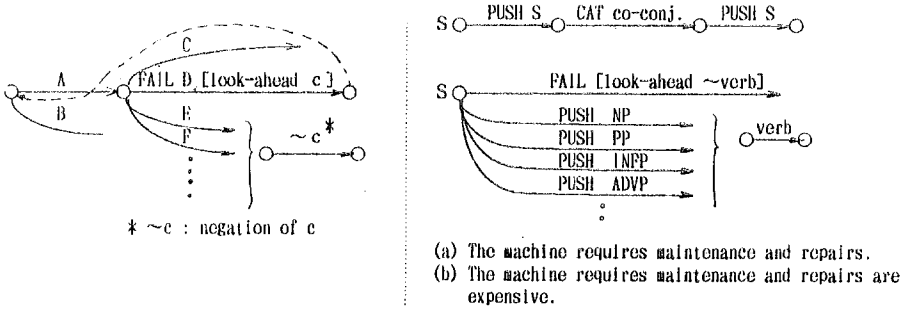


Figure 2 Rejection of all possibilities of one state

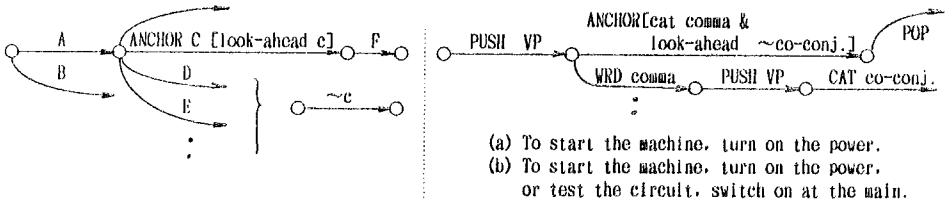


Figure 3 Rejection of all but one possibility

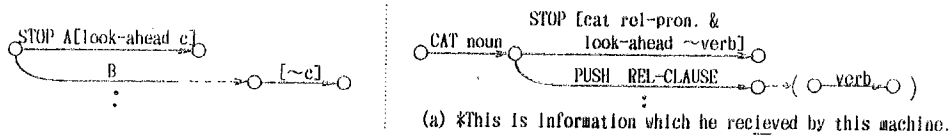


Figure 4 Rejection of the ungrammatical input sentence

(2) reordering of syntactic possibilities

In the traditional parsing methods, arcs with higher-possibility at the state are written at upper side of arcs with lower-possibility. Therefore, the application order of arcs can be dynamically reordered only by information of the current word, not by the future input. But by looking ahead the future input, arcs can be dynamically reordered to select the arc with highest possibility. This function reduces the time for finding a successful parse.

A "jump arc" used for change of application order is shown in Figure 5. The general network of Figure 5 has the following meaning:

- If the condition 'c' is satisfied in the future input, arc 'A' and 'B' are tried in this order; otherwise, arc 'B' and 'A' are tried in this order. In either case, both interpretations are possible at the current word position: only the ordering is affected.

In the example (a), parsing "restaurants and cafeterias" as the subject is preferred, and in the example (b), parsing as the object of "in" is preferred.

Also, we can specify in conditions the number of words to be searched. This condition works as a heuristic for reordering arcs in order to reduce time required for success (see (4)). It is difficult to use this number in order to reject arcs, because there may be intervening optional elements and so this number may cause mistakes (see (1)).

(3) rejection of categorial possibilities

There are many words which have categorial ambiguities in English. In parsing English, categorial ambiguity is a big problem, especially in the case of very commonly used words. Examples are "as" (preposition, adverb, conjunction, relative pronoun), "after" (preposition, subordinate conjunction, adverb), and "that" (demonstrative pronoun, relative pronoun, conjunction). How the look-ahead function works effectively for categorial disambiguation is as follows, and reordering of categorial possibilities is presented in (4).

The use of arcs is the same as (1). Here the example shown in Figure 6 is the rejection of "as" as an adverb, if a second "as" is not found, and the rejection of "as" as a conjunction, if a verb is not found. In the example (c), parsing "as" as adverb and conjunction can be avoided by linear search.

Such categorial ambiguity is also common in Japanese, for example, "no", "de", etc. These categorial ambiguities in Japanese can be also avoided by look-ahead.

(4) reordering of categorial possibilities

How look-ahead works effectively for reordering of categorial possibilities is presented.

The use of arcs is the same as (2). An example is

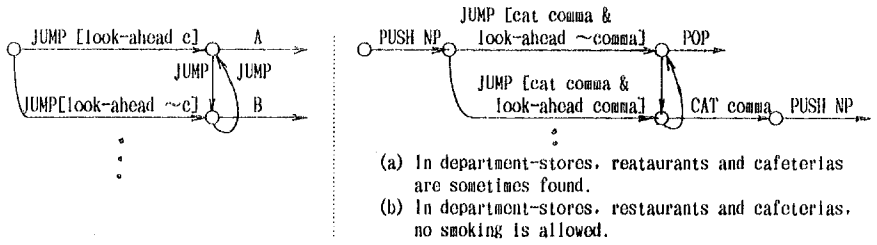


Figure 5 Reordering of syntactic possibilities

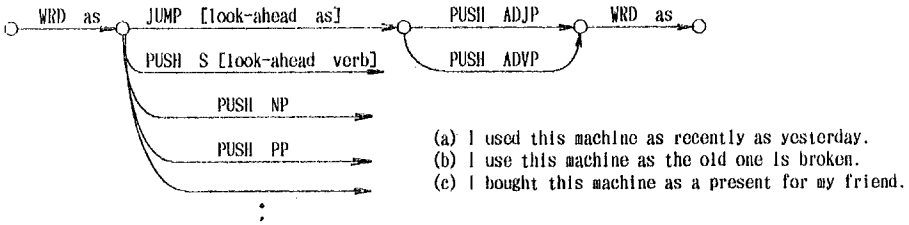


Figure 6 Rejection of categorial possibilities

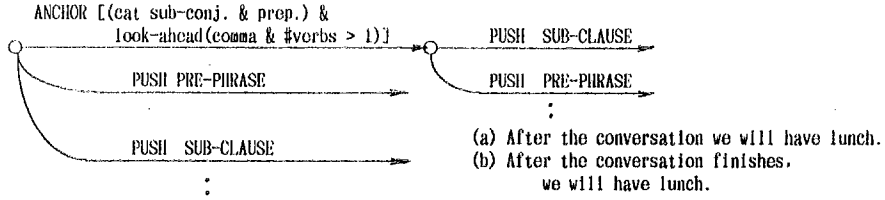


Figure 7 Reordering of categorial possibilities

shown in Figure 7. This network applies for a word like "after" which has multiple category ambiguity. At the beginning of a sentence, if there is comma and more than one verb in the future input, the subordinate conjunction interpretation is preferred (i.e. tried first), otherwise, the preposition interpretation is preferred. In the example (a), parsing "after" as preposition is preferred, and in the example (b), parsing "after" as conjunction is preferred.

3.2. Parsing with vs without look ahead

Despite of its advantages, mentioned earlier, a question may arise in using look-ahead conditions:

Is look-ahead rather time-consuming?

The top-down depth-first search like ATN takes k_1C^n time[Aho 1972], where C is a constant, n is the number of input words, and k_1 is a coefficient which is determined by time taken for the traversal of one arc. The look-ahead search presented here however, takes k_2n time, where k_2 is a coefficient which is determined by time taken for one word check of a look-ahead condition.

If a look-ahead condition is satisfied, k_1C^n can be reduced to k_2n. If not, time taken for look-ahead is wasted. Therefore, the effect of look-ahead is determined by these trade off. This increases as n and C increase (i.e. the length of sentences and the size of the grammar increase) and as k_2 decreases.

In our ATN, k_1 is nearly equal to k_2, since one check of look-ahead conditions which we use take about the same time as the traversal of one arc, as discussed in (1). Therefore, the effect of look-ahead is generally determined by the difference of the number of checks between arc traverses and looking-ahead. Since the number of arcs from one state often is greater than the number of the input words, as discussed in (1), and the number of checks for looking ahead is less than the number of input words, extra time needed in look-ahead is not too significant. Of course, look-ahead conditions are attached to arcs only when it is effective.

For the above reasons, the number of arcs with effective look-ahead increases in realistic, large-size grammars such as our ATN.

4. Conclusion

We have shown how a parsing method using global information as well as information about the current word, is very effective for increasing the efficiency in the face of natural language phenomena such as categorial and syntactic alternatives.

This is especially significant in a real-time translation system using large-size dictionaries and realistic grammars for natural language. In such a system, enormous numbers of rule applications, which are caused by categorial and syntactic alternatives and which eventually lead to failure, can be rejected or put off by the linear search of an input sentence.

In this paper, we have focussed on the parsing of English, but this method is also very effective for Japanese, inasmuch as it has similar features to English. For example, a Japanese sentence also requires a verb, a noun phrase requires a noun, etc, and many structures involve discontinuities such as "shika -- nai", "kara --- made", "to --- to --- to", etc.

In our system, look-ahead conditions are written manually. A compiler which can automatically attach look-ahead conditions at source ATN is being considered for a future system.

The look-ahead parsing also works very effectively for scientific and technical documents since these documents are more complicated and longer than communication dialogues (see 3.2)[Nogami 1987].

Acknowledgements

We would like to thank Harold Somers (the Centre for Computational Linguistics, University of Manchester Institute of Science and Technology, England) for his comments on an earlier version of this paper.

References

[Aho 1972] Aho, A. V. and Ullman, J. D.: The Theory of Parsing, Translation, and Compiling, Volume 1: Parsing. Prentice-Hall, Inc, Englewood Cliffs, N. J., 1972.

[Amano 1986] AMANO shin-ya (1986). The Toshiba Machine Translation system. Japan Computer Quarterly 64 "Machine Translation - Threat or Tool" (Japan Information Processing Development Center, Tokyo), 32-35

[Amano 1987] AMANO Shin-ya, Hideki HIRAKAWA & Yoshinao TSUTSUMI (1987). TAURAS: The Toshiba Machine Translation System. Machine Translation Summit, Manuscripts & Program, Tokyo: Japan Electric Industry Development Association (JEIDA), 15-23.

[Amano 1988a] AMANO shin-ya, Kimihito TAKEDA, Koichi HASEBE & Hideki HIRAKAWA (1988). Experiment of Automatic Translation Typing Phone (in Japanese). Information Processing Society of Japan (1988. 3. 16-18)

[Amano 1988b] AMANO Shin-ya, Hiroyasu NOGAMI & Seiji MIIKE (1988). A step towards Telecommunication with Machine Interpreter. Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, June 12-14, 1988 (CMU)

[Asahioka 1988] ASAHIOKA Yoshimi, Yumiko YOSHIMURA, Seiji MIIKE & Hiroyasu NOGAMI (1988). Analysis of the Translated Dialogue by Automatic Translation Typing Phone (in Japanese). Information Processing Society of Japan (1988. 3. 16-18)

[Marcus 1980] Marcus, M.: A Theory of Syntactic Recognition for Natural Language, MIT Press, 1980

[Miike 1988] MIIKE Seiji, Koichi HASEBE, Harold SOMERS & Shin-ya AMANO (1988). Experiences with an on-line translating

dialogue system. 26th Annual Meeting of the ACL

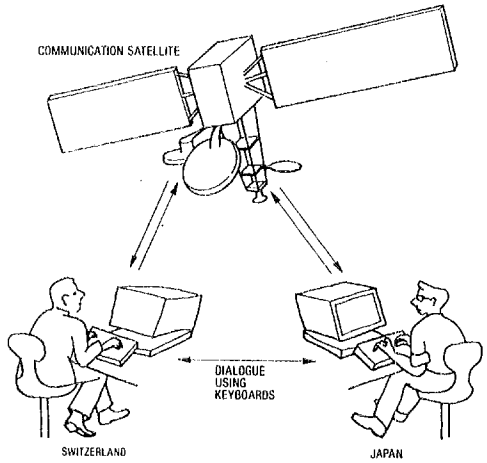
[Nogami 1987] NOGAMI Hiroyasu, Yumiko SUGIURA, Hideki HIRAKAWA & Shin-ya AMANO. English Analysis in an English-Japanese Machine Translation System (in Japanese). Information Processing Society of Japan 61-4 (1987. 5. 22)

[Pereira 1980] Pereira, F. C. N. and Warren, H. D.: Definite Clause Grammars for Language Analysis, A Survey of the Formalism and a Comparison with Augmented Transition Networks. Artificial Intelligence, 13, 1980

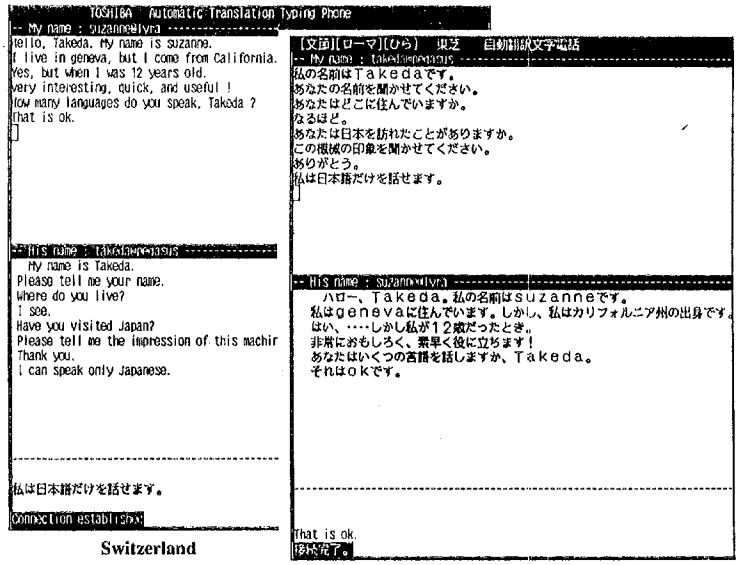
[Takeda 1988] TAKEDA Kimihito, Koichi HASEBE & Shin-ya AMANO (1988). System Configuration of Automatic Translation Typing Phone (in Japanese). Information Processing Society of Japan (1988. 3. 16-18)

[Woods 1970] Woods, W. A.: Transition Network Grammars for Natural Language Analysis, CACM Vol. 13, pp. 591-606, 1970

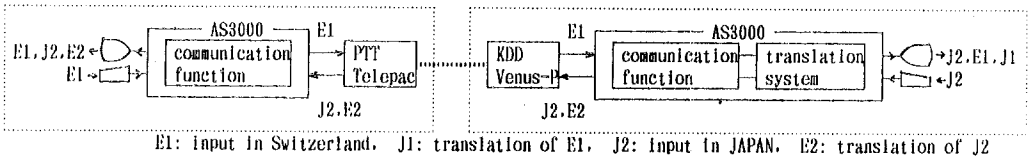
Appendix



General set-up



Screen display



E1: input in Switzerland. J1: translation of E1. J2: input in JAPAN. E2: translation of J2

Switzerland Japan

System configuration