A PROGRESS REPORT ON THE USE OF ENGLISH IN INFORMATION RETRIEVAL

J. A. MOYNE

IBM Corporation and Harvard University

Boston Programming Center
IBM Corporation
545 Technology Square
Cambridge, Mass. 02139
U.S.A.

ABSTRACT

Progress is reported in the further development of an already working model for communicating in English with a computer about the contents of a library. The revised grammar of this model combines the phrase structure and transformational rules of the underlying grammar into a single efficient component. Problems of implementation and ambiguity resolution are discussed.

During the academic year 1966-1967 a system, Proto-RELADES, was designed and implemented at Boston Programming Center, IBM Corporation, for communication with a computer (System/360, Models 40 and 50). This system has been operational since June 1967[1]. It permits the user to communicate with the computer in English about the contents of the library at the Center[2]. The underlying grammar in this system is a recognition grammar based on the generative approach in linguistic theory. The pioneering work for a recognizer for a generative grammar was done by Petrick (1965). Among the transformational grammars

---

1. This system was reported in Moyne (1967a) and a detailed specification of it is included in Moyne (1967b).

2. One can type English sentences at a computer terminal making queries, giving commands and, in general, asking for the retrieval of any pertinent data about the content of the library.

developed for computer application two stand out for their historical impact on this approach: The Mitre (1964) grammar developed by a number of M.I.T. scholars, and the so-called IBM Core Grammar[3]. A lucid and informative discussion of the implications of the use of natural languages in computers is given in Kuno (1967).

The theoretical and historical significance of these grammars notwithstanding, they all have serious practical disadvantages in that tney generate all the possible syntactic analyses for every ambiguous sentence but have no practical way of selecting in a fast and efficient manner the sense of the sentence either intended by the user or inherent in the nature of the discourse. In Proto-RELADES, we tried to avoid this difficulty by restricting the discourse to a highly-specialized field and thus reduced most of the ambiguities to the lexical level. In his important work on semantics for question answering systems. Woods (1967) adopts the same approach, but he stipulates that the ultimate solution for resolving ambiguities in a more general system is in interaction with the user. This is, of course, the most general solution. If one can generate all the possible analyses of a sentence and let the user select the analysis which reflects his sense of the sentence, one would delegate the choice of understanding to the user and will satisfy nim as long as the user knows what he is talking

---

3. Rosenbaum and Lochak (1966). For the latest version of this grammar, see Rosenbaum (1968).

about. However, this approach is also unsatisfactory for practical reasons, even if an easy way to build such an interactive system were known. Under a time-sharing environment, which is the only practical environment for on-line systems of this kind, every interruption and interaction will cost time, and the total effect will make the system so slow and cumbersome to make it impractical.

In this paper, we will propose some additional devices for the automatic resolution of ambiguities. These devices are now being studied and implemented at the IBM Boston Programming Center. Ideally, one should not have to arbitrarily restrict the types of sentences which the user of the system may input to the grammar, i.e., the grammar should be able to parse any sentence of any length. Implementation of this ideal goal is, however, presently untenable. We will outline here our efforts to approach this goal to the extent which is possible under the present state of the art.

The grammar of Proto-RELADES was a standard recognition grammar with separate phrase structure and transformational components; that is, phrase structure rules would apply to the input sentence and produce a surface structure. The latter would then be the input to the transformational component and the output of this component would be the deep structure of the sentence. Our new experimental grammar combines these two components into one integrated system of rules. To understand the implication of this, we must look at the form and

nature of the rules in this grammar. Each rule in this grammar has the following format:

(1) L_i: A·BC ↑ D·E ⟶ F X @Y@ *** /L_n

This rule has a label L_i and a GOTO instruction L_n. The function of the rule can be paraphrased as follows: Check to see that the elements ABC are to the left of the pointer ↑ in the input sentence and that the elements D and E are to the right of it (there is no upper limit to the number of the elements to the left and right of the pointer; there must be at least one element to left of the horizontal arrow ⟶ .) If this is the case, then if condition X is satisfied, perform action Y and create a node F to dominate over the symbols between the two dots (·) on the left of the arrow (X and Y can be null). Next, move the pointer to the right according to the number of the stars (*) at the tail end of the rule and go to the rule labeled L_n. If this rule does not apply, the control will pass on to the next rule in the sequence, i.e., to L_{i+1}.

We see at once that this rule format permits one to write context sensitive rules constrained by some conditioning factors and also build local transformations in the Y part of the rule. The traffic in the rule application is controlled by the GOTO label L_n. Underlying this system of rules is the "reductions analysis" (RA) recognizer which reads the rules and applies them to the input sentence resulting in a tree structure (P-marker) representing the deep structure of the sentence.

The RA in our system is an extension of the model proposed by Cheatham (1968).

Culicover (1969) and Lewis (1969) have written and implemented a grammar which uses these rules with exclusively local transformations. The net result of this grammar is that a canonical deep structure is produced for the input sentence without the generation of the intermediate surface structure. In terms of computer efficiency and speed, this is a significant step. The theoretical significance of such a recognition grammar has yet to be studied.

The ambiguities can be resolved by the following interactions, all of which are automatic internal and, therefore, fast interactions, except the last one. In a fully-generalized system, all these interactions must be implemented in a manner that they will tradeoff against each other for reducing the complexity and increasing the speed.

The final interaction on list (2), i.e., human interaction, which is the last resort in this system can be omitted or its use greatly restricted in many practical situations. The interactions are with:

(2) (i) the lexicon
(ii) the date base
(iii) the system
(iv) the human user

Lexical entries have a certain number of features which play a role in the structural analysis of the input sentence. This is based on the already well-known proposal of Chomsky (1965) for syntactic features. A simple example of a semantic feature of a sort is given

below:

(3) John wrote the book on the shelf.

If the word shelf in the lexicon has a feature or features denoting that it is a place for storing books, etc., but normally people do not write on it or reside on it, then in the process of the analysis of (3) the prepositional phrase on the shelf will be recognized as modifying the noun book and not the verb write or the proper noun John. The trouble with this solution is obvious: there will be too many simple and complex features for each entry in the dictionary[4], and we run into severe problems for practical applications. This is why we want to reduce the reliance on the dictionary features to the minimum and tradeoff as far as possible with the other interactions listed under (2) above.

Interaction with the data base will provide the discourse background and may turn out to be the most significant and practical means for resolving ambiguities. For our system, this category of interaction includes looking up in micro-glossaries; that is, specialized glossaries containing the jargon of each narrow field of application. Again, a highly simplified example of interaction with the data base is the following. Suppose that the input sentence was

(4) Do you have any books on paintings by Smith?

Somewhere in the process of the derivation of the underlying structure

---

4. For a fractional grammar of English with partial features specified, see Rosenbaum (1968).

(and the interpretation) of the sentence in (4) it becomes necessary to decide whether the phrase by Smith modifies books or paintings, that is whether the question is about books by Smith or about paintings by Smith. At this point, the system can look into the data base and see, for example, whether Smith occurs under the column for authors or for painters and resolve the ambiguity accordingly.

Interaction with the system is similar to the interaction with the data base except that here we question the capabilities of the underlying system in order to resolve the ambiguity. Consider the following example:

(5) Do you have any documents on computers?

The ambiguity in (5) is, among others, in whether we want documents written about computers or we are referring to piles of documents on the top of computers. Now the underlying system which analyzes and interprets (5) and produces the answer to the question has certain capabilities; for example, it has computer routines for searching lists of titles, authors, etc., printing data, and whatever else there is. However, if the system does not have a facility for "looking" on the top of the computers in search of documents, we can reject that interpretation and adopt one which concerns documents containing information about computers.

The human interaction becomes necessary only when none of the above devices resolve the ambiguity; for example, in the case of the

data base sample in sentence (4) above when the data base has the name Smith under both the author and painter columns. In this case, the system should formulate some sort of simple question to ask the human user before the final interpretation is effected; for example: "Do you mean books by Smith or paintings by Smith or both?" But, as I mentioned above, we have found in practice that, within a specified discourse and with a properly organized lexicon and data base, the need for taking this last resort seldom arises; and that is why systems such as Proto-RELADES and Woods (1967) can have significant practical claims.

In summary, we visualize a restricted but completely practical natural language system for communication with a computer and information retrieval with a general lexicon and specialized micro-glossaries. Certain restrictions in the lexicon and in the micro-glossaries will prevent wild generation of all possible and obscure (or unlikely) analyses but will permit generation of all the reasonable analyses for each input sentence. Interactions with the lexicon, the data base (i.e., the subject of the discourse) and system will further eliminate the various analyses for each sentence until one analysis is left. In such cases when the system is unable to reduce the query to one analysis, the human user is asked to help in clarifying the ambiguity.

I would like to close this paper, however, with a word of caution. No linguist and no serious computational linguist will claim that he knows how to build a system such as outlined above for a completely unrestricted processing of a natural language. The stress throughout this paper has been on practicality. We visualize a restricted natural language system of the sort which is fully practical and useful for many applications in information sciences.

Bibliography

Cheatham, T. E., Jr. (1968) "On the Generation and Implementation of Reductions Analysis Programs for Context Free Grammars," Computer Associates, C/ -6902-2911, Feb. 29, 1968.

Chomsky, Noam (1965) Aspects of the Theory of Syntax, M.I.T. Press, Cambridge, Mass.

Culicover, Peter (1969) "A Discussion of the CUE Grammar" (forthcoming)

Kuno, Susumu (1967) "Computer Analysis of English," Mathematical Aspects of Computer Science, American Mathematical Society, Vol. 19.

Lewis, Clayton (1969) "The RA-DRYAD System for Automated Recognition Grammars" (forthcoming)

Mitre (1964) English Preprocessor Manual, The Mitre Corporation Bedford, Mass.

Moyne, J. A. (1967a) "Toward the Understanding of Natural Languages by Machines," Proceedings of the X International Congress of Linguists, Bucharest.

Moyne, J. A. (1967b) Proto-RELADES: A Restrictive Natural Language System, IBM Corp., BPC Technical Report No. 3, Oct. 3, 1967, Cambridge, Mass.

Petrick, Stanley Roy (1965) A Recognition Procedure for Transformational Grammars, M.I.T. doctoral thesis, Cambridge, Mass.

Rosenbaum, P. and D. Lochak (1966) The IBM Core Grammar of English, Thomas J. Watson Research Center, IBM Corporation, Yorktown Heights, New York.

Rosenbaum, Peter S. (1968) English Grammar II, IBM Research Report R C 2070, Yorktown Heights, New York.

Woods, William A. (1967) Semantics for a Question-Answering System Report No. NSF-19, The Computation Laboratory, Harvard University, Cambridge, Mass.