NATURAL LANGUAGE PROCESSING AND THE AUTOMATIC ACQUISITION OF KNOWLEDGE: A SIMULATIVE APPROACH

Danilo FUM

Laboratorio di Psicologia E.E. - Università di Trieste
via Tigor 22, I - 34124 Trieste (Italy)

ABSTRACT

The paper presents the general design and the first results of a research project whose long term goal is to develop and implement ALICE, an experimental system capable of augmenting its knowledge base by processing natural language texts. ALICE (an acronym for Automatic Learning and Inference Computerized Engine) is an attempt to model the cognitive processes that occur in humans when they learn a series of descriptive texts and reason about what they have learned. In the paper a general overview of the system is given with the description of its specifics, basic methodologies, and general architecture. How parsing is performed in ALICE is illustrated by following the analysis of a sample text.

1. INTRODUCTION.

The capability to learn is one of the central features of intelligent behavior, and learning constitutes one of the current hot topics in artificial intelligence (Michalski, Carbonnell, and Mitchell, 1983). Much of the work on this field has dealt with induction, rule discovery, and learning by analogy or from examples, whereas much less effort has been dedicated to building systems able to learn by processing natural language texts. As Norton (1983: 308) remarked, the general agreed-upon assumption was that "such a capability is not 'learning' at all but merely (?) the conversion of knowledge from one representation to another". Acquiring new knowledge via prose comprehension is, on the contrary, a complex activity which relies on understanding the linguistic input, storing the extracted information in memory, and integrating it with prior knowledge for effective use. As far as psychology is concerned, learning from written texts has often aroused the interest of cognitive and educational psychologists. Due to the limitations of the experimental approach which has been generally adopted, however, this topic has seldom been dealt with in its entirety. Lots of experiments have been carried on focusing on restricted arguments and specific phenomena whose explanations too often look suspiciously ad hoc. Unfortunately, those who addressed the full problem of 'meaningful verbal learning' (e.g. Ausubel, 1963) stated their theories so vaguely that it is almost impossible to express them in form of effective procedures and to implement them in computer programs.

In the last few years the situation has changed and several projects (Frey, Reyle, and Rohrer, 1983; Haas and Hendrix, 1983; Nishida, Kosaka, and Doshita, 1983; Norton, 1983) are now devoted to develop computer systems which could automatically extract information from written texts. Practical applications, besides theoretical interest, motivates this kind of research. In the expert system technology, for example, the process of discovering what is known to the experts of the field in which the program must perform requires tedious and costly interactions between the knowledge engineer and those experts. Automatic acquisition of knowledge by text understanding could represent a way to partially reduce the labor and fatigue involved in the transfer of expertise.

The paper presents the general design and the first results of a research project whose long term goal is to develop and implement ALICE, an experimental system capable of augmenting its knowledge base by processing natural language texts and reasoning about them. Particular attention is given to the simulative aspects of the project. ALICE (an acronym for Automatic Learning and Inference Computerized Engine) is an attempt to model the cognitive processes that occur in humans when they learn a series of descriptive texts and reason about what they have learned. Comparisons with what is known about human cognitive behavior are therefore explicitly taken into account in devising algorithms and data structures for the system. In the next section a general overview of the system is provided with the description of its specifics, basic methodologies, and general architecture. The third section briefly describes the parser used in

ALICE, and how parsing is performed is illustrated in section four by following the analysis of a small sample text. Section five concludes the paper by giving a summary of the main ideas and some implementational details.

2. ALICE: A GENERAL OVERVIEW

2.1 Specifics

The main goal of the ALICE project is to examine how it is possible to build a machine which could, in a psychologically plausible way, learn new facts about a given domain by analysing natural language texts. ALICE can operate according to two different ways: in learning mode and in consult mode. In learning mode ALICE is given in input a series of sentences in Italian forming simple introductory scientific passages. The domains chosen for the initial experimentation are elementary chemistry and electronics. The system understands the input texts and integrates the information extracted from them with that previously stored in its knowledge base. For checking purposes the system outputs the sentence-by-sentence internal representation that is added to the knowledge base. When working in consult mode, ALICE receives in input a question concerning the processed texts and returns the portion of the knowledge base containing the information needed to answer it. It should be noted that the system has no generation capabilities; it does not output natural language sentences but only the internal representation of a small part of its knowledge base. Another limitation of the system is that it can deal with questions only in a piece-meal fashion. ALICE, in other words, lacks the dialogic capabilities needed to build a graceful man-machine interface. User modelling, mixed-initiative dialogue, co-operative behavior etc. are simply outside the scope of the project.

ALICE cannot obviously understand all the sentences that is possible to express in a given language. Unrestricted language comprehension is currently beyond our capabilities. As work in artificial intelligence and computational linguistics has taught us, it is very difficult to build programs that could successfully cope with linguistic materials. This is due to the fact that language is essentially a knowledge-based process. In understanding natural language it is necessary to make a heavy reliance on world knowledge even to do very elementary operations: disambiguate the meaning of a word, identify an anaphoric referent, capture the syntactic structure of a sentence. Paradoxically it has been said that one cannot learn anythingt unless (s)he almost knows it already. In order to avoid the danger of being stuck in a loop (i.e., text understanding requires a rich stock of knowledge, but in order to acquire such a knowledge it is necessary to understand textual material), the passages given in input, derived from programmed instruction textbooks, were kept relatively simple from the linguistic point of view.

As an automatic knowledge acquisition system, ALICE differs from other natural language processors in that, by definition, its knowledge base is incomplete. This means that, at the beginning, not only its conceptual coverage but also its linguistic (particularly lexical) capabilities are quite limited. A great deal of work in learning a new subject is constituted by mastering new concepts and the terminology needed to refer to them. When the system encounters a word for which it has no definition in its dictionary, it should be able to learn this new word and guess at its meaning. Doing this can be easy when the new word is explicitly defined in the text but it can require non-trivial inferential processes if the new word is implicitly introduced by relating it with other concepts whose meaning is already known.

ALICE comes preprogrammed with a fixed set of rules enabling it to cover a small subset of Italian. It also comes with seed concepts and a seed vocabulary which are to be extended as the system learns about the new domain. ALICE acquires new knowledge by integrating the information extracted from the input texts with that previously stored in its knowledge base. As a result of its operation, ALICE's conceptual coverage increases with the number of passages in a given domain which have been understood. ALICE is thus capable of understanding more complex texts since its encyclopedic knowledge can be brought to be bear in the comprehension process. A necessary prerequisite to this accomplishment is that parsing input texts should not be considered as a separate activity but it must be integrated with the remaining operations performed by the system.

2.2 Knowledge Representation Methods

An important point in the design of every artificial intelligence program is constituted by deciding how to represent knowledge. A good formalism should be able to express all the knowledge needed in a given application domain, and should facilitate the process of acquiring new information. ALICE adopts a clear distinction between declarative and procedural knowledge. This is a critical, and not at all obvious, choice.

Norton (1983), for example, adopts as the target representational formalism for his system statements in the PROLOG language which can be interpreted both declaratively and procedurally. From a psychological point of view, however, there are strong reasons for maintaining the distinction between these two kinds of knowledge (Anderson, 1976:116-119):
- the declarative knowledge seems possessed in all-or-none manner whereas it is possible to possess procedural knowledge only partially;
- the declarative knowledge is acquired suddenly by being told whereas the procedural knowledge can be acquired only gradually by performing a skill;
- it is possible to communicate verbally the declarative but not the procedural knowledge.

In ALICE the declarative knowledge is constituted by the information that the system is able to derive from the texts. It is represented through the BLR propositional language (Fum, Guida, and Tasso, 1984), a formalism derived by augmenting the representation used in psychological setting by Kintsch (Kintsch, 1974; Kintsch and van Dijk, 1978) with the features necessary to make it computationally tractable. The procedural knowledge represents the knowledge necessary to the system operation. It is expressed in form of production systems which operate on the propositions contained in the knowledge base. There are several motives that make the use of productions systems particularly interesting to model human cognitive processing. Productions systems provide a unifying formalism to deal with the different kinds of processes that occur in knowledge acquisition through text comprehension. Moreover, they are especially suitable to support the strategic approach on which the system operation is grounded.

2.3 Basic Methodologies

The strategic approach to text understanding, and reasoning with linguistic materials, can be fruitfully contrasted with the algorithmic one. Examples of the algorithmic approach in the field of natural language processing can be found, for example, in the use of grammars which produce structural descriptions of sentences by syntactic parsing rules. In the field of inferential processes this approach is represented by theorem provers based on resolution mechanisms which, granting that a theorem could be derived from a given set of axioms, are able to discover its proof. These processes can be complex, long and tedious but they guarantee success as long as the algorithm is correct and it is correctly applied. The strategic approach does not guarantee a priori success. It is based on a set of heuristics, expressed as production rules, which constitute some working hypotheses about how to discover the correct meaning of a fragment of text or the way by which a certain inference could be drawn. Strategies are rules of thumb which are applied to analyse, understand, and reason about natural language texts. Humans differ in their cognitive functioning according to the amount and the kind of strategies they have at their disposal, and according to the way in which these strategies are applied. Experimental evidence for the strategic approach has been gathered since a long time. Clark and Clark (1977) reviewed some of the strategies utilized in sentence comprehension; van Dijk and Kintsch (1983) wrote a whole book to examine the strategies employed in discourse understanding, and Anderson (1976) examined the strategies his subjects adopted to perform formal deductions in syllogistic reasoning tasks.

The strategic approach is inextricably linked with other assumptions concerning text understanding and learning. The goal of the human understanding activity (and of the systems aimed at modelling human cognitive processing) is not the discovery of the syntactic structure of a sentence but of its meaning. This does not mean that syntax is of no use in text understanding. Syntactic information, however, constitutes only one among the different knowledge sources utilized to capture the meaning of a piece of text, and syntactic analysis represents neither a separate phase nor a prerequisite for comprehension activity. The construction of the meaning representation takes place more or less at the same time of the data input. Humans do not wait until an entire sentence is uttered before they begin to interpret what has been said. They may have expectations about what sentences look like, and these expectations may facilitate the understanding process. As words are being received people try to build a possible semantic interpretation for them. Additional words are used to confirm or disconfirm that interpretation. In the latter case, a new interpretation is build and it is checked against the new data. There is no fixed order between input data and their interpretation: interpretations may be data driven or they may be constructed in absence of external evidence and only later be matched with data.

Language understanding is a multifaceted activity and several kinds of competence are needed to perform it. ALICE relies on a series of specialists which co-operate in performing the variuos operations (i.e., parsing, inferencing, memory management) which are required to acquire new knowledge by text comprehension.

2.3 General Architecture

ALICE is composed (see fig. 1) of the following modules:

- the parser
- the inference engine
- the memory manager
- the monitor

which can utilize, in order to perform their activity, two data structures: the knowledge base and the working memory.

The knowledge base can be considered as the long term memory of the system. Information extracted from the texts received in input is represented in declarative form in such a structure. The knowledge base is constituted by a huge amount of BLR propositions linked to form a cohesion graph. Unlike semantic networks, a cohesion graph only indicates the fact that some concepts and propositions of the knowledge base are connected; all the information concerning the kind of relationship existing among them is to be found in the BLR propositions. The knowledge base is concept indexed; it can be accessed through one or more concepts that become thus activated. From these concepts activation spreads, through the the different kinds of arcs - irrespective of their direction - to the propositions in which they are contained and to other concepts connected to them. This mechanism of spreading activation, similar to that described in Quillian (1969), Collins and Loftus (1975) and Anderson (1976), makes it possible to selectively access the information contained in the knowledge base.

The working memory represents the short term memory of the system. It is a memory of limited capacity which represents the portion of the knowledge base which can be accessed and operated upon by the different productions. To utilize a piece of knowledge, it is necessary to activate it, i.e. it must be present in the working memory. The working memory stores generally only the information connected to the sentence that is currently being processed plus some information necessary to understand the sentence (information needed to draw an inference, to establish coreferential links and coherence, to exactly quantify an expression etc.).

The system modules do not communicate directly with each other but they can exchange information only through the working memory which serves as a "blackboard" for the whole system. There are some important differences, however, between the use of

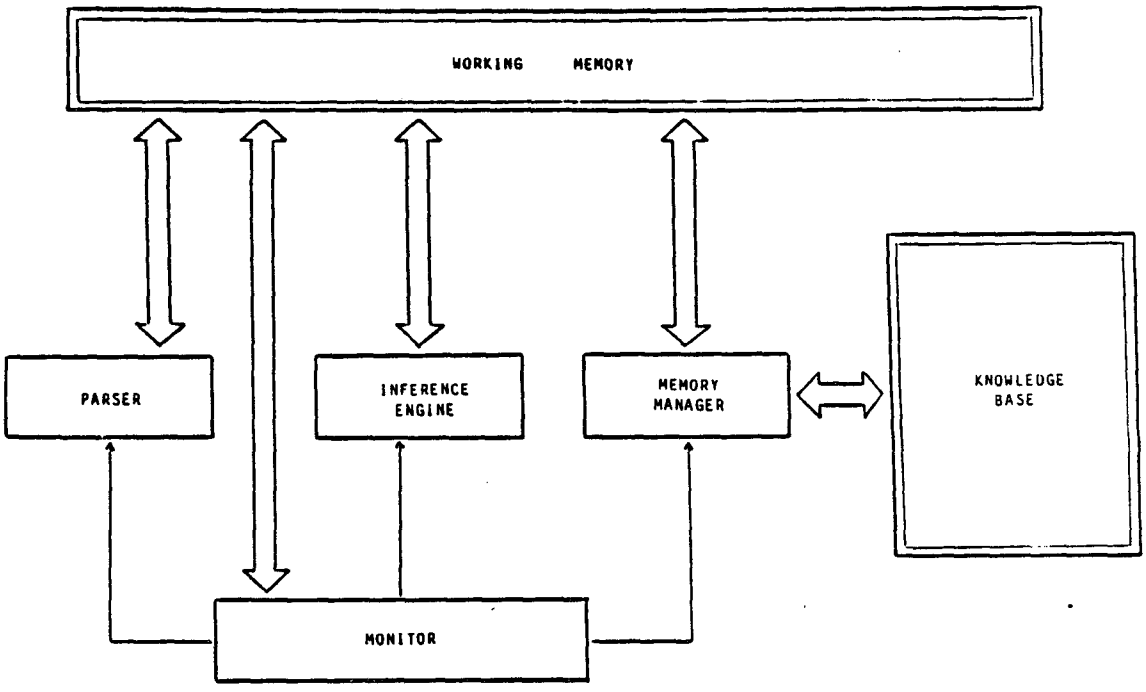

Fig.1: The General Architecture

the working memory in ALICE and other blackboard-based system like HEARSAY-II (Lesser and Herman, 1977; see also: Cullingford, 1981)). First, in HEARSAY-II each specialist expresses its hypotheses on the blackboard in its own representation language. In ALICE, BLR is the common language for representing all the information provided by the specialists. Second, the control of the specialist activity is decentralized in HEARSAY while in ALICE the control information is explicitly present rather than diffused through a large database. The activity of the different modules does not depend only from the content of the blackboard but is directly controlled by the monitor which disciplines the operation of the different modules.

The parser is devoted to translate a natural language expression (a sentence to be processed in learning mode or a query to be answered in consult mode) into the BLR representation. This activity is performed through the collaboration of a number of parsing specialists which are supposed to be competent in each of the several domains involved in language understanding, and to cover the wide spectrum of different capabilities required to build up the text representation. Parsing is strictly integrated with the other operations performed by the system: inferencing and memory management (i.e., retrieving old information to be utilized in text understanding, and integrating new information in the knowledge base).

The inference engine is the module devoted to perform the inferences required to understand a piece of text or to answer a question. Its task is to go beyond the information given and to discover new information to be supplied to the system. Different kinds of inferences are performed by this module: propositional, pragmatic, and formal deductions. Propositional inferences are based on linguistic features of predicates. They are necessarily true and can be directly derived from the semantic content of the propositions. Pragmatic inferences are derived from knowledge sources beyond the explicit, linguistic input. They are not necessarily true but only plausible. Pragmatic inferences, however, are often drawn in processing natural language to establish, for example, the coherence of seemingly separate segments of texts, to understand referential expressions, to build "bridging implicatures", etc. Formal deductions are often required to understand scientific passages. Humans, however, are different from theorem provers in that they are neither sound nor complete inferential engines. They sometimes reason in contrast with the dictates of logic; they do not draw every possible consequence from a set of premises but only those that appear sensible and interesting; finally, they perform in a reasonably efficient manner. The inference engine module is an attempt to simulate human inferential processes in dealing with scientific texts.

The memory manager is the only module which interacts directly with the knowledge base. It is devoted to retrieve some information necessary to the system operation, to match the information extracted from the current text with that contained in the knowledge base, to upgrade it by integrating the new knowledge. The memory manager implements a multiple-access, parallel search assumption concerning the way the knowledge based is searched for information. This means that the system memory can be accessed from all the concepts contained in the linguistic input and that the concepts spread their activation in parallel among the links departing from them. When the minimum length path between two concepts is discovered the propositions standing on it are returned as being relevant to the current input. Through the memory manager it is possible to simulate certain process that are known to occur in human memory, for example propositional fan and interference effects.

3. TOWARDS A MENTAL PARSER

In accordance with the general simulative approach of the ALICE project, the main criterion to follow in designing and evaluating a parser is that of how well its operation corresponds to the way humans understand language. Unfortunately, in spite of lots of psycholinguistic studies, we are far from knowing how the mind works. Experimental evidence, at most, can help us to put some constraints on the specifics of a 'mental parser'. It is apparent, for example, that human parsing does not occur entirely top-down or bottom-up but uses some combination of these strategies. It is almost certain, moreover, that humans do not use backtracking or looking ahead in order to cope with nondeterminism (Johnson-Laird, 1983).

The most important preliminary question to be dealt with in the design of a mental parser, however, is that of what mechanisms people use in understanding. Linguists hold that people rely on formal rules and that they have implicit knowledge of the grammar they apply in analysing a sentence. Some of the rule systems that linguists use to parse sentences are implausible as psychological models: the resources they demand and the computations involved simply exceed the human processing limitations (see, for instance Anderson's critique of ATN formalisms: Anderson,

1976).

The parser that has been designed for ALICE relies on the strategic approach (van Dijk and Kintsch, 1983) implemented through production systems and constitutes a first step toward the construction of a psychologically viable mental parser. The parsing process is organized around a set of parsing specialists. The monitor is in charge of controlling the overall parsing activity and of directing the operation of the specialists towards the construction of the BLR. It utilizes a set of construction rules which represent knowledge about the BLR, about the use of the specialists, and about the use of the information supplied by the specialists for the construction and validation of the BLR. The specialists are devoted to analise the input text and to supply the information necessary to the monitor. The general philosophy of the parser is to exploit any and all available knowledge whenever helpful. The specialists are therefore supposed to be competent in each of the severals domains which are involved in the comprehension activity and to cover the wide spectrum of different capabilities required to build up the BLR.

The following specialists are used:

- morpholexical specialist
- syntactic specialist
- semantic specialist
- quantification specialist
- reference specialist
- time specialist.

The morpholexical specialist analyzes the words contained in the natural language sentences. It is the specialist which performs the segmentation of words into morphemes and which looks up the dictionary for their definition. In case the processed word is unknown, the specialist provides some hypotheses about its morpholexical features (gender, number, lexical class, etc.) which will be used for guessing, in collaboration with the other specialists, the meaning of the new word. The syntactic specialist tries to discover the surface structure of each sentence, and to recognize its functional organization. The rules it utilizes do not represent a 'grammar' for the language but only some hypotheses concerning the role of word order in the determination of meaning. The semantic specialist is aimed at proposing a first tentative interpretation of the natural language sentences as a series of BLR propositions. It recognizes the predicates which will be used in the construction of propositions and checks that such predicates will be instantiated with the correct arguments. The quantification specialist is used to discover how the arguments of the propositions could be quantified. The reference specialist is devoted to examine if each concept conveyed by the input text represent a unique token or if it refers to other concepts known by the system. The time specialist examines the time specifications contained in the text which are implicit in the tense of verbs or explicitly stated through the use of temporal adverbs or time expressions.

4. AN EXAMPLE

This section gives an idea of the parser operation by following in some detail the analysis of a small sample text. Let us consider the following sentence:

"La materia e' composta da molte sostanze differenti."
(The matter is composed of many different substances.)

As mentioned above, ALICE works under the control of the monitor which directs and coordinates the activity of the specialists. The monitor starts by examining the first word of the sentence and puts the following information into the working memory:

```
10 EQUAL ($1, "LA")
20 EQUAL ($PROC-WORD, $1).
```

BLR constitutes in ALICE the common language through which the specialists can exchange information and communicate with each other. The only difference between the standard BLR (as described in Fum, Guida & Tasso, 1984) and the formalism here utilized is the introduction of linguistic variables (identified by the $ sign) used exclusively in the parsing activity. The $ sign can be followed by an index which indicates the word to which the variable refers. The index can be constituted by:
- an integer, for example: $1, $2, $3, in which case the variable refers to the first, second, third word of the sentence, respectively;
- a letter, for example $x, $y, in which case the variable refers to a generic word of the sentence;
- an expression indicating a fixed displacement in relation to a given word. So, for instance, $x-1, $x+1, $y+2, $3+2 refer respectively to the word that immediately precedes that indicated by the $x variable, to the word that follows it, to the word that comes two positions in the sentence after that referred to by $y, and to the fifth word of the sentence;
- an expression indicating a generic displacement in relation to a given word. $x+n, $5-n therefore indicate a word that generically follows the xth

word of the sentence, and a word that generically precedes the fifth word of the sentence.

The main variables utilized in the present example are:

- $.PROC-WORD, which represents the word the system is currently processing;
- $(index).CLASS, $(index).GENDER, $(index).NUMBER, $(index).FUNCTION, which represent the lexical class, the gender, the number, and the syntactic function of the (index)th word of the sentence, respectively;
- $(index).CONCEPT, which represents the concept to which the (index)th word refers and into which it is mapped in the course of the parsing activity.

The predicate EQUAL is used to indicate that its arguments can be considered as the same thing and can therefore be utilized interchangeably. Proposition 10 then asserts that the variable $1 has the value "La", that is "La" is the first word of the sentence. Proposition 20 states that $1 (i.e. "La") is the word that is currently processed. This information triggers the activity of the specialist that performs the morpholexical analysis. Looking at its dictionary, the specialist finds that "La" can be a a definite (feminine, singular) article or a (feminine, singular) pronoun that is used only as object. The specialist returns the following propositions:

```
30 EQUAL ($1.GENDER, FEMININE)
40 EQUAL ($1.NUMBER, SINGULAR)
50 XOR (60, 70)
60 ?EQUAL ($1.CLASS, DEF-ARTICLE)
70 ?AND (80, 90)
80 ?EQUAL ($1.CLASS, PRONOUN)
90 ?EQUAL ($1.FUNCTION, OBJECT)
```

These propositions give the complete morphological analysis of the word "La". Proposition 50 states an alternative and indicates that only one of its arguments is true:
- either the current word is a definite article, or
- both of the following facts hold: (i) the current word is a pronoun and (ii) it appears as the object of the current sentence.
Propositions preceded by the ? sign represent expectations the system has or conditions that must be fulfilled by the content of the working memory.

Since propositions 10 and 20 cannot activate other specialists, the control returns to the monitor which tries to determine the truth value of propositions 60-90. There is not enough information in the working memory to allow performing this activity and the monitor, therefore, starts another processing step. In the next cycle the activity of the syntactic and reference specialists can be triggered since the condition part of some of their productions match the information contained in the working memory. In particular, the syntactic specialist has in its rule base the following productions:

```
IF      EQUAL ($x.CLASS, DEF-ARTICLE)
THEN    XOR (P, Q)
     P ?EQUAL ($x+1.CLASS, NOUN)
     Q ?EQUAL ($x+1.CLASS, ADJECTIVE)
```

and

```
IF      EQUAL ($x.CLASS, DEF-ARTICLE)
        EQUAL ($x.GENDER, g)
        EQUAL ($x.NUMBER, n)
THEN    EQUAL ($x+1.GENDER, g)
        EQUAL ($x+1.NUMBER, n)
```

i.e., if a word of a sentence is a definite article it has to be followed by a noun or an adjective which must agree with its gender and number. The former production is triggered by proposition 60 which represents only a plausible alternative and states an assertion whose truth value must still be determined. This fact represents a typical case of conditional matching which is taken into account by the monitor which subordinates the execution of the action part of such production to the truth of proposition 60. As a result, the following propositions are generated:

```
100 IMPLY (60, 110)
110 ?XOR (120, 130)
120 ?EQUAL ($2.CLASS, NOUN)
130 ?EQUAL ($2.CLASS, ADJECTIVE)
```

The latter production, after matching (conditionally) the first clause with proposition 60, and matching the second and third with propositions 30 and 40, respectively, generates:

```
140 IMPLY (60, 150)
150 ?AND (160,170)
160 ?EQUAL ($2.GENDER, FEMININE)
170 ?EQUAL ($2.NUMBER, SINGULAR).
```

The syntactic specialist knows also that, if a pronoun appears as the object of a sentence, the following constituent orders are feasible in Italian: SOV, OVS, VOS, i.e, the pronoun must be preceded or followed by a verb. This information is represented in the following production which is triggered in the same cycle:

```
IF      EQUAL ($x.CLASS, PRONOUN)
        EQUAL ($x.FUNCTION, OBJECT)
THEN    XOR (P, Q)
     P ?EQUAL ($x-1.CLASS, VERB)
     Q ?EQUAL ($x+1.CLASS, VERB).
```

This production is triggered by propositions 80 and 90 which must be both true in order to allow considering proposition 70 - which represents a plausible alternative and whose truth value must be still determined - also true. This case of conditional matching is taken into account by the monitor too and what results is:

```
180 IMPLY (70, 190)
190 ?XOR (200, 210)
200 ?EQUAL ($0.CLASS, VERB)
210 ?EQUAL ($2.CLASS, VERB).
```

In the same cycle, the reference specialist is triggered which uses the heuristic:

"IF a determiner has been identified
THEN look for a noun that specifies the header of the noun phrase."

This general heuristic is implemented in this particular case by the following production:

```
IF      EQUAL ($x.CLASS, DEF-ARTICLE)
THEN    EQUAL ($x+n.CLASS, NOUN)
        EQUAL ($x+n.CONCEPT, HEADER)
```

and the following information is returned:

```
220 IMPLY (60, 230)
230 ?AND (240, 250)
240 ?EQUAL ($1+n.CLASS, NOUN)
250 ?EQUAL ($1+n.CONCEPT, HEADER)
```

These propositions state that one the of next words of the sentence should be syntactically classified as a noun and that the concept to which this noun refers shoud be considered the header of the noun phrase.

Another heuristic utilized by the reference specialist is the following:

"IF a pronoun has been identified,
THEN look for the referent among the nouns wich have the same gender and number.".

This heuristic is implemented through the following production:

```
IF      EQUAL ($x.CLASS, PRONOUN)
        EQUAL ($x.GENDER, g)
        EQUAL ($x.NUMBER, n)
THEN    EQUAL ($x.CONCEPT, $y.CONCEPT)
        EQUAL ($y.CLASS, NOUN)
        EQUAL ($y.GENDER, g)
        EQUAL ($y.NUMBER, n)
```

The first clause of the condition part of the production matches (conditionally) proposition 70 while the second and third clause match propositions 30 and 40, respectively. The production gives raise to the following propositions:

```
260 IMPLY (70, 270)
270 ?AND (280, 290, 300, 310)
280 ?EQUAL ($1.CONCEPT, $y.CONCEPT)
290 ?EQUAL ($y.CLASS, NOUN)
300 ?EQUAL ($y.GENDER, FEMININE)
310 ?EQUAL ($y.NUMBER, SINGULAR).
```

i.e., if "La" is a pronoun it refers to a concept represented in the text by a word which is a feminine, singular, noun.

The information present in the working memory at the beginning of the cycle (propositions 10-90) cannot activate other specialists. After all the productions have fired in a cycle, the results are taken into account by the monitor which checks the results obtained through the work of the specialists. The monitor tries to establish the truth value of the propositions preceded by the ? sign, it tries also to identify the concepts to which variables indexed by a letter or an expression refer and, more generally, it checks the compatibility and consistency of the propositions in the working memory. In our example, the only thing that the monitor can do at this point is to capture the error condition contained in proposition 200 which has among its arguments the variable $0.CLASS, i.e. the variable which refers to the sytactic class of the 0th word of the sentence. Proposition 200 is recognized as stating something that cannot be true and, as a consequence, one of the alternatives stated in proposition 190 is not valid any more. The monitor substitutes the second argument of proposition 180 with 210, while propositions 190 and 200 are deleted. At this point we know a lot about the current word. We know that "La" is an article or a pronoun and in both cases we know what should happen next. If "La" is an article, a noun must follow sooner or later, and the concept referred to by this noun will be the header of the noun phrase. In particular, the next word must be a noun or an adjective, and it must be singular and feminine. If "La" is a pronoun, on the other hand, it must be followed by a verb and its referent must be looked for among the concepts which are

represented in the sentence by feminine singular nouns.

The next word to be processed is "materia". Before the morpholexical specialist could be activated the monitor performs some housekeeping operations on the content of the working memory. It deletes proposition 20 which is not true any more and adds the following propositions to the working memory:

320 EQUAL ($2, "MATERIA")
330 EQUAL ($.PROC-WORD, $2)

The morpho lexical specialist analyses the new word and gives as a result the information that it is a feminine, singular noun. Moreover, the word "materia" corresponds to a concept known by the system, i.e. it is a lexical entry which refers to the concept MATTER. The following propositions result from this analysis:

340 EQUAL ($2.CLASS, NOUN)
350 EQUAL ($2.CONCEPT, MATTER)
360 EQUAL ($2.GENDER, FEMININE)
370 EQUAL ($2.NUMBER, SINGULAR)

In this case we have no problems of semantic ambiguity since MATTER represents the only concept that the system can connect to the word "materia". Generally speaking, however, each word of the sentence may refer to a number of different concepts and it is not always possible to decide which interpretation is appropriate until more of the sentence has been analyzed. The approach taken in ALICE to solve semantic ambiguity is to use more information about the context in which the current sentence appears. Spreading activation is the mechanism used for this purpose. Another classic way to deal with cases of polysemy that is sometimes used in ALICE is to attach to certain interpretations a series of requests or expectations that must be fulfilled by the content of the working memory.

Coming back to our example, the information returned by the morpholexical specialist allows the monitor to perform a series of checks on the content of the working memory concerning the propositions whose truth value must be determined and the expectations the system has. In particular: after a series of deductions for which the help of the inference engine module is requested, the following propositions remain in the working memory:

10 EQUAL ($1, "LA")
30 EQUAL ($1.GENDER, FEMININE)
40 EQUAL ($1.NUMBER, SINGULAR)
60 EQUAL ($1.CLASS, DEF-ARTICLE)
120 EQUAL ($2.CLASS, NOUN)
160 EQUAL ($2.GENDER, FEMININE)
170 EQUAL ($2.NUMBER, SINGULAR)
240 EQUAL ($1+n.CLASS, NOUN)
250 EQUAL ($1+n.CONCEPT, HEADER)
320 EQUAL ($2, "MATERIA")
330 EQUAL ($.PROC-WORD, $2)
350 EQUAL ($2.CONCEPT, MATTER)

This information triggers the activity of the specialists: the syntactic specialist recognizes that the definite article and the noun are part of a noun phrase. This can be complete or, in Italian, one or more adjectives can follow the noun. Proposition 60, 120 and 250 at the same time trigger the activity of the reference and quantification specialists. The reference specialist looks for another occurrence of the supposed header of the noun phrase in the working memory.The quantification specialist tries to find how the header of the noun phrase must be quantified. In this particular case it uses the following heuristic:

"IF the header concept is an individual concept,
AND it has not being previously referred to
THEN quantify it individually"

and as a result it quantifies individually the concept MATTER (Fum, Guida, & Tasso, 1984). The parsing process goes on by identifying the verb of the sentence. The verb "e' composta" is recognized as an instance of the concept COMPOSE which represents the constitutive relation of the following predicate:

COMPOSE (<composer>, <composee>)

The task of the parser becomes now that of figuring out the arguments of this predicate. After discovering that the preposition "da" signals that the verb is in the passive form, that it is in present tense, and after solving some problems posed by the second noun phrases which contains the fuzzy quantifier "molte", the parser has all the elements necessary to build up the BLR . What results in the working memory after the parsing has been completed is the following:

3070 COMPOSE (.VV1, MATTER, P)
3080 *SUBSTANCE (VV1)
3090 MANY (.VV1)
3100 DIFFERENT (VV1, P)

i.e. there exist a subset (= more than one) VV1 of entities which are of the type SUBSTANCE (i.e. each of them ISA SUBSTANCE) that taken together

compose the individual entity MATTER; the cardinality of this subset is MANY, and each of the entities have the property to be DIFFERENT. Propositions 3070-3100 are given as output of the parsing process and are stored in the knowledge base where they can be accessed to answer questions.

5. CONCLUSION

In the paper the general design of ALICE has been presented and an ilustration of the parser used by the system has been given. The main ideas on which such an attempt is grounded are:
- to exploit all of the possible knowledge to aid the system in the parsing activity,
- to parallelize the morphologic, syntactic, and semantic analysis, the determination of referents, quantification, etc, and to pursue them as soon as enough information has been gathered;
- to provide through the use of the production system formalism, an integrate framework into which all the problems posed by the language understanding activity could be dealt with.

A prototype reduced version of the system, implemented in FLISP under NOS 2.2 on a Control Data Cyber 170, is currently running at the University of Trieste and shows the feasibility of this approach. A full system implementation in Common LISP is under development.

REFERENCES

Anderson, J.R. (1976). Language, Memory, and Thought. Hillsdale: N.J., Erlbaum.

Ausubel, D.P. (1963). The Psychology of Meaningful Verbal Learning. New York, N.Y.: Grune & Stratton.

Clark, H.H. and Clark, E.V. (1977). Psychology and Language. New York, N.Y.: Harcourt Brace Jovanovich.

Collins, A.M. and Loftus, E.F. (1975). A Spreading-Activation Theory of Semantic Processing. Psychological Review (82) 407-428.

Cullingford, R. (1981). Integrating Knowledge Sources for Computer "Understanding" Tasks. IEEE Transactions on Systems, Man, and Cybernetics (11) 52- 60.

Frey, W., Reyle, U., and Rohrer, C. (1983). Automatic Construction of a Knowledge Base by Analisyng Texts in Natural Language. Proceedings of the IJCAI-83, Los Altos, CA: Kaufmann.

Fum, D., Guida, G., and Tasso, C. (1984). A Propositional Language for Text Representation, in: B.G. Bara and G. Guida (Eds.), Computational Models of Natural Language Processing, Amsterdam: North-Holland.

Haas, N. and Hendrix, G.G. (1983). Learning by Being Told: Acquiring Knowledge for Information Management, in: R. Michalski, J.G. Carbonnell Jr., and T.M. Mitchell, (Eds.), Machine Learning, Palo Alto,CA: Tioga

Johnson-Laird, P.N. (1983). Mental Models. Cambridge, U.K.: Cambridge University Press.

Kintsch, W. (1974). The Representation of Meaning in Memory. Hillsdale, N.J.: Erlbaum.

Kintsch, W. and van Dijk, T. (1978). Toward a Model of Text Comprehension. Psychological Review (85) 363-394.

Lesser, V.R. and Erman, L.D. (1977). A Retrospective View of Hearsay-II Architecture. Proceedings of the IJCAI-77, Los Altos, CA: Kaufmann.

Michalski, R., Carbonnell, J.G. Jr., and Mitchell, T.M. (Eds.) (1983). Machine Learning, Palo Alto,CA: Tioga

Nishida, T., Kosaka, A., and Doshita, S. (1983). Towards Knowledge Acquisition from Natural Language Documents. Proceedings of the IJCAI-83, Los Altos, CA: Kaufmann.

Norton, L.M. (1983). Automated Analysis of Instructional Texts. Artificial Intelligence (20) 307-344.

Quillian , M.R. (1969). The Teachable Language Comprehender: A simulation program and a theory of language. Communications ACM (12) 459-476.

van Dijk, T. and Kintsch, W. (1983). Strategies of Discourse Comprehension. New York, N.Y.: Academic Press.