GOAL ORIENTED PARSING: IMPROVING THE EFFICIENCY OF NATURAL LANGUAGE ACCESS TO RELATIONAL DATA BASES

Giovanni Guida

Milan Polytechnic Artificial Intelligence Project
Politecnico di Milano
Milan, Italy

This paper is devoted to present a new approach to natural language understanding which is called here goal-oriented parsing. The interaction in natural language with artificial systems (robots, data base systems, program generators, question-answering systems, etc.) does not require in most cases of actual interest a full (human like) comprehension of natural language in all its details and nuances. A partial understanding is often enough ,wich extracts from the natural language expressions the only significant information which is necessary to construct a correct formal input for the target system. In such a model of comprehension the same meaning is assigned to several different natural language expressions, thus defining a many-to-one mapping between natural language sentences and corresponding formal representations. We argue that a bounded scope, restricted, goal-oriented understanding of natural language may greatly increase the efficency of representation models and parsing algorithms, thus allowing the construction of effective systems. This claim is supported by the design and implementation of a natural language interface to a relational data base called NLI and developed at the Milan Polytechnic Artificial Intelligence Project. In the paper the architecture of the system, the linguistic models, and the parsing algorithms are presented and illustrated through selected examples. Promising directions for future research are outlined as well.

1.Introduction

The development of natural language understanding systems has received in the last years a growing interest, both in the area of computational linguistics and artificial intelligence. In this field a lot of running systems have been implemented which are based on different linguistic models and parsing algorithms.

This paper is devoted to illustrate NLI, a natural language understanding system which has been developed at the Milan Polytechnic Artificial Intelligence Project for the inquiry in Italian of a relational data base. NLI is based on a new approach to natural language understanding called here goal-oriented comprehension. It is claimed that the understanding activity may be correctly defined only if the purpose or goal of the comprehension is clearly specified, and that the same sentence may have several different meanings depending on the purpose for which it is considered. It follows that the complexity of the natural language understanding task depends, in addition to the inner linguistic or structural complexity of the sentence which is considered, on the complexity and extent of the scope of the understanding activity. Therefore, we can argue that, if one confines the understanding activity to well defined and bounded goals (as it is generally possible in most of the natural language processing applications), it should be possible to develop ad hoc linguistic models and parsing algorithms which are particulary fitting and efficent. Such models and algorithms will obviously strictly depend on the particular goal which is considered and will generally not hold for very different scopes. Let us outline that goal-oriented understanding implies a bounded scope, incomplete,and in some way diagonal comprehension, in which any detail and nuance which is not relevant to the goal is ignored.

In the design of NLI the classical topic of data base inquiry has been chosen as application domain and the goal of the understanding process has been defined as that one of translating the user's natural language queries into the formal query language QBE (Query-By-Example[15]). The adopted liguistic model is strongly semantics based and doesn't take into account,as much as possible, the syntactic aspects of the sentence to be analysed[1]. The linguistic information needed for the parsing is stored in a lexicon and a base of the heuristics,and the parsing algorithms are strictly dependent on the particular goal which is considered.NLI has been so far developed in two versions,NLI-1 and NLI-2; the latter one, which is presently in an advanced development stage, is the subject of this work.

The paper is organized in the following way: in section two the version NLI-1 is shortly illustrated and the design specifications and criteria for NLI-2 are discussed; section three is devoted to present the overall architecture of the system and the parsing algorithms which have been implemented; section four shows some selected examples of comprehension; in section five some promising directions for future research are illustrated together with some conclusive remarks.

2.Previous work and experimental activity

In this section the first prototype version of the system NLI(NLI-1) is shortly illustrated. The functional requirements and the basic technical decisions are then presented, on which a second version of NLI (NLI-2) is presently being developed.

NLI-1 has been designed and implemented at the Milan Polytechnic Artificial Intelligence Project in the years 1977-78[4,5] with the following aims:

- to evaluate through a sample application the new model of natural language understanding proposed and the designed algorithms for semantic-based parsing[1];
- to study in a particular case the most critical aspects involved by the develepent of actual applications,such as efficiency, memory occupations, creation and management of the dictionaries, evaluation of the understanding capabilities, etc.

The inquiry in natural language(Italian) of a toy relational data base representing the catalogue of a library has been chosen as an application domain for NLI-1. The architecture of the system has been structured in three sequential modules. The first module scanns the input sentence, searches in the vocabulary each word which has been recognized through a simplified lexical analysis, and generates an internal representation of the sentence in which the meaning of each word and elementary construct is embedded. The second module is devoted to reduce the ambiguities and to check the correctness of the structure proposed (through an algorithm based on set intersection[1]); it produces a semantic tree which represents the formal internal representation of the input sentence. The third module translates the semantic three into an equivalent sentence expressed in a formal query language for relational data bases (SEQUEL).

NLI-1 has been implemented in FORTRAN first on an UNIVAC 1100 and, later, on a Digital PDP-11/34 computer. It requires about 15 Kbytes memory with a lexicon of 200 words. The parsing of a simple sentence(a single query of about 10-15 words) needs a few seconds and more complex sentences need up to 20-30 seconds.

The research activity done with NLI-1 has highlighted some relevant aspects concerning the development and actual implementation of NLI, which will be further considered in the design of NLI-2. We list some of these below:

- the necessity of developing an high-level diagnostic subsystem to be used by the application designer(often,the end-user himself) in the tuning activity;
- the need for an interactive system for the incremental definition and for the management of the vocabulary(to be utilized also in the design of a new application versions of NLI);
- the necessity of refining the parsing algorithms and of defining a base of the heuristics to be utilized in the most complex cases of comprehension;
- the problem of the optimal content of the vocabulary (increasing the content of the vocabulary doesn't always improve the understanding capabilities of the system!);
- the problem concerning the need of inserting in the vocabulary all the words which constitute the content of the data base.

For greater details about NLI-1 the reader is referred to the literature[5].

The new version NLI-2 has been designed in 1979 on the base of the experience acquired through NLI-1. The main methodological assumptions, funcitonal specifications,and technical decisions on which NLI-2 is based are illustrated below:

- goal-oriented understanding;
- semantics directed parsing;
- hierarchical organization of the linguistic information supplied to the system in two modules:the lexicon and the base of the heuristics;
- development of a generalized parsing algorithm independent of the content of the lexicon and of the base of the heuristics, of the structure of the data base, and of the particular natural language which is used;
- developent of an high-level diagnostic module;
- design of a fully interactive system for the incremental definition and dynamic management of the content of the lexicon and of the base of the heuristics and for the implementation of new versions of the system;
- implementation of the appropriate human engineering features needed to improving the usability of the system.

The natural language used for the inquiry is Italian and the output of the system is the translation of the query in QBE (Query-By-Example[14]). The data base adopted for the development of the sample application of NLI-2 is concerned with a department store description and is just the same used for the illustration of QBE (a complete definition of the data base is reported in the Appendix). In designing NLI-2 it has been decided of not inserting anything about the content of the data base in the lexicon of the system, in order to keep its dimensions as small as possible. This choice implies that the user indicates explicitely(through a pair of special symbols") in the input sentence the words which denote values of domains of the data base. The adequacy and effectiveness of this decision will be evaluated during the experimental activity to be done with NLI-2 and could be removed if considered unappropriate.

The next section will be entirely devoted to the illustration of the global structure of NLI-2 and of the parsing algorithms which have been developed.

3.System architecture and parsing algorithms

NLI-2 is based on a modular structure in which there is a distinct division between the data(vocabulary,data base model, formal query language structure) and the programs(analyser,generator, vocabulary management). The system architecture is illustrated in Figure 1.

The monitor is devoted to open the job(it informs the user about the system capabilities and present to him an appropriate menu of different options), to interprete the user's commands, and to manage the activities of the lower modules. The user's control messages accepted by the monitor are:

- $HELP : detailed informations and instructions on the operation of the system are supplied;
- $U : the understanding cycle of the natural language queries is activated; the system is ready to run in U-mode;
- $V : the module for the management of the vocabulary is activated; the system can be used for the definition,updating,and tuning of the vocabulary in the V-mode;
- $STOP : returns the control to the monitor for U-mode and V-mode operations;
- $END : closes the job.

The analyser, which is activated by the monitor when running in the U-mode,accepts the natural language query in input and generates an appropriate internal representation or, if some step of the parsing fails, a diagnostic message to be displayed to the user. The analyser operates in a cyclic way: when the parsing of a sen-

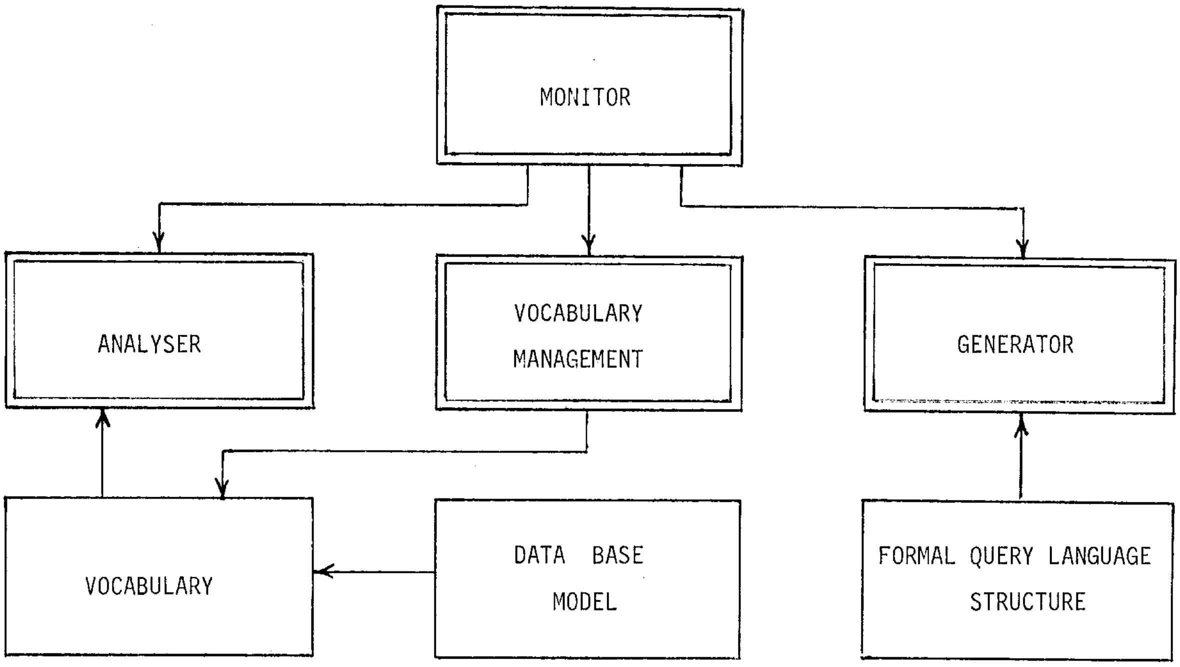

Figure 1 - NLI-2 architecture

tence is concluded it is automatically reset and is ready to accept a new query. Queries are not stored by the parser, so that the user is not allowed to carry on a dialog with the system in which new queries can refer to old ones(or to their answers). The module for the vocabulary management can be activated by the monitor or directly by the analyser when a fail occurs. It supplies the user with some basic functions(some of them mainly relating to text editing) for a flexible management of the vocabulary(search, display,delate,add,update,etc.).The generator receives in input the internal representation produced by the analyser and translates it into the formal query language QBE.

Let us describe now the internal structure of the vocabulary. It is composed of two parts: the lexicon, which contains the words and the elementary construct referring to the application domain in which the system operates,and the base of the heuristics , which embeds further linguistic information(sometimes also of syntactic nature) necessary to understand complex and often ambiguous constructs.

The lexicon is organized in 26 alphabetic groups,each one of them contains all the words beginning by the same letter. Each word (or elementary construct) is bound to all its possible meanings,thus yelding a record of the lexicon. A record is in fact composed of two parts:

- the left part contains a word or a (short) sequence of words representing a simple construct; morphology is (generally)not taken into account in the parsing,but the wordsare represented in such a way to be recognized in all possible forms(conjugated verbs,inflected nouns, ect.);
- the right part embeds the representation of the semantics of the word within the application domain which is considered.

The right part of each record has a different structure depending on the semantic type to which the word stored in the left part belongs. In our application a word can denote three different types of information concerning:

- the identification of the relations and of the domains involved by the query;
- the logical connection between relations and domains;
- the specification of the required output.

Therefore, in relation to the three above presented activities, we define the following semantic types: object, connective, and function , respectively.

The general structure of an object record is:

P	O	$N_1^1.M_1^1=N_2^1.M_2^1=\dots$	$N_1^2.M_1^2=N_2^2.M_2^2=\dots$	---

where:

- P denotes a word or a simple costruct;
- O denotes that the record is of type object;
- each pair $N_i^j.M_i^j$ denotes that the word P refers to the domain N_i^j in the relation M_i^j (relations and domains are represented by positive integers); the = symbol separates equal domains belonging to different relations; if P refers to a relation without specifying the domain, N_i^j is replaced by the special symbol ∅;each field of the record contains a different meaning of the object P.

The structure of a connective record is:

P	C	$X_1 : Y_1$	$X_2 : Y_2$	---

where:

- P denotes a word or a simple costruct;
- C denotes that the record is of type connective;
- each pair $X_i : Y_i$ denotes a possible meaning of the connective P; namely, X_i represents the pattern in which P may appear(position of P and type of the adjacent words),Y_i denotes(through a pointer) the function which must be applied or the action which must be performed during the parsing to take correctly into account the meaning of the connective P.

The structure of a function record is:

P	F	Z_1	Z_2	---

where :

- P denotes a word or a simple construct;
- F denotes that the record is of type function;
- each Z_i denotes a possible meaning of the function P and represents (through a pointer) an action which must be performed in the parsing.

The base of the heuristics is constituted by a bag of heuristic rules(of the type precondition-action), which allow to represent linguistic informations which are not comtained in the lexicon but are still needed for understanding complex sentences and for the resolution of ambiguities. The heuristic rules are selected and activated during the parsing,whenever a critical situation occures,on the base of a pattern directed invocation algorithm.

Let us outline that the vocabulary(both its structure and content) is strctly dependent on the particular application to which the system is devoted. This feature,which represents a quite rigid bound to the flexibility of the system, is, on the other hand, a straightforward consequence of the concept of goal-oriented understanding. The issue of designing system architectures which allow a flexible handling of the purpose and domain of the comprehension is considered as a promising and ambitious topic for future research,as it will be illustrated in the conclusions.

Let us illustrate now the activity of the analyser. It accept as input a natural language query and supplies an internal formal representation of it which is not far from a QBE expression; in such away the role of the generator is confined, within NLI-2, to a simple editing activity. The activity of the analyser can be splitted in four steps:

1. scanning of the natural language input sentence, search in the lexicon,and generation of a first-level internal representation;
2. parsing,(partial or full) resolution of the ambiguities, and generation of the second-level internal representation;
3. intersection,i.e. verification of the consistency of the proposed structure and resolution of the possibly remaining ambiguities;
4. generation of the correct formal representation or, if any fail has occured, of the appropriate diagnostics.

In step 1. the input sentence is first scanned and an internal representation of it is constructed(first-level internal representation).The searching of the words in the lexicon is indexed sequencial(with a one-level index); if a word or elementary construct is found in the lexicon it is replaced by all the record to which it refers; on the other hand, if a word is not recognized, it is enclosed between brackets in the first-level internal representation and it is successively ignored in the parsing. The words which denote values of domains,i.e. which appear in the content of the data base, are preceded and followed by the special symbol" in the input sentence and remain unaltered in the internal representation. The first-level internal representation reflects the ordering of the words in the input sentence and embeds all the linguistic information which can be obtained from the lexicon.

In step 2. the relations involved by the user's query are first recognized through a detailed analysis of the words of type object. This activity cannot yeld, in general, a definite and unambiguous result since some objects may refer to different relations. A pattern directed invocation of appropriate heuristics may contribute in eliminating some (or also all)of the ambiguities. After the correct relations are individuated,their logical schemata are extracted from the data base model and a new phase starts aiming to individuate the domains referred to in the input sentence. The possible ambiguities may be resolved through an appropriate use of the heuristics. The domains which have a role in the user's query are then labelled in order to be further processed in the following steps of the parsing. The words enclosed between pairs of " symbols are then considered in order to find the appropriate assignement of these values to the relating domains. Different criteria may be used to perform such an assignement(e.g., the contiguity of a value and an object),but, in any case, the assignement is not a definitive one until it is confirmed by an heuristic or by the following step 3.(intersection). A tentative interpretation of the input sentence(second-level internal representation) is then produced,which will be further refined and completed in the following steps of the parsing.

The analysis of the connectives constitutes the kernel of step 3.;it is organized in two phases:

- singling out the correct semantics of a connective on the base of its position in the sentence and of the type of the words to which it refers(appropriate heuristics have to be utilized to resolve possible ambiguities);
- verification of the proposed interpretation of the sentence segment to which the connective belongs(this activity is performed through set intersection algorithms[1], what gives the reason for the name assigned to step 3.).

In step 4. the functions,which possibly appear in the input sentence, are considered and the domains to which they apply are individuated.

If the parsing terminates correctly the formal representation of the sentence is generated which will be later translated in QBE by the generator.

In the following section some complete examples of comprehension are illustrated.

4.Parsing sample sentences

In this section we are going to present some examples of parsing extracted from the sample application in which NLI-2 is presently working (the department store data base which is described in the Appendix).

Example 1.

Input sentence:

< TROVA I REPARTI CHE VENDONO PRODOTTI FORNITI

DA "PARKER"> (1)

The first-level internal representation,after step 1. of the parsing is concluded,results:

< TROVA/F/PV/ (I) REPARTI/O/1.4=1.2 (CHE) VENDONO

/C/$X_1 \$ X_2$,$X_1$=O,$X_2$=V:PA2.2/$Y_1 \$ Y_2$,$Y_1$=O,$Y_2$=O:PR2

PRODOTTI/O/2.2=1.3=1.4 FORNITI DA /C/$X_1 \$ X_2$,$X_1$=O,

X_2=V:PA2.3/$Y_1 \$ Y_2$,$Y_1$=O,$Y_2$=C:PR3 "PARKER">

In step 2. the relations involved by the user's request are first recognized,

<VENDITE (REPARTO,ARTICOLO) >

< FORNITORI (ARTICOLO,FORNITORE)>

and, then, the domains referred to in the input sentence are labelled through the letters A and B:

< VENDITE (REPARTO:A,ARTICOLO:B)>

< FORNITORI (ARTICOLO:B,FORNITORE)>

The second-level internal representation (proposing a tentative assignement of values to the labelled domains) results:

< VENDITE (REPARTO:A,ARTICOLO:"PARKER") >

< FORNITORI (ARTICOLO:"PARKER",FORNITORE)>

After step 3. we get:

< VENDITE (REPARTO:A,ARTICOLO:B) >

< FORNITORI (ARTICOLO:B FORNITORE:"PARKER") >

and, after step 4., the correct formal representation in QBE results:

VENDITE (REPARTO: $\underline{\text{P}}$, ARTICOLO:B)

FORNITORI (ARTICOLO:B, FORNITORE: "PARKER")

PV is a pointer to a routine which puts the special symbol $\underline{\text{P}}$ in the domain which follows the function TROVA. PA is a pointer to a routine which assigns X_2 to domain 2 of relation 2 (in the case VENDONO),or to domain 2 of the relation 3 (in the case FORNITI DA); PA is activated if the connective to which it is bound connects an object (O) to a particular value of a domain(V), as it arrives in both cases VENDONO and FORNITI DA. The second meaning of these two connectives, represented by the pointer PR, is not considered in this parsing.

Example 2.

Input sentence:

< VOGLIO I NOMI DEGLI IMPIEGATI CHE GUADAGNANO

PIU' DEL LORO DIRIGENTE> (2)

First-level internal representation:

< VOGLIO/F/PV (I) NOMI DEGLI IMPIEGATI/O/1.1

(CHE) GUADAGNANO/O/2.1 PIU' DEL/F/PM (LORO)

DIRIGENTE/O/3.1>

Second-level internal representation:

< IMPIEGATI (NOME:A, SALARIO:B, DIRIGENTE:C,

REPARTO) >

After step 3. we get:

< IMPIEGATI (NOME:A, SALARIO:B, DIRIGENTE:C,

REPARTO) >

In step 4., after the analysis of the functions, we have:

< IMPIEGATI (NOME:$\underline{\text{P}}$, SALARIO:B, DIRIGENTE:C,

REPARTO)>

In this representation there are two domains(SALARIO and DIRIGENTE) whose role is not yet understood; this pattern activates a particular heuristics which allows two obtain the following correct formal representation:

IMPIEGATI (NOME:$\underline{\text{P}}$, SALARIO > B, DIRIGENTE:C,

REPARTO)

IMPIEGATI (NOME:C, SALARIO:B DIRIGENTE,

REPARTO)

Example 3.

Input sentence:

< VORREI I NOMI DELGI IMPIEGATI CHE LAVORANO

NELLA DIVISIONE "CANCELLERIA" > (3)

First-level internal representation:

< VORREI/F/PV (I) NOMI DEGLI IMPIEGATI/O/1.1

[1] find the departments which sell products supplied by Parker

[2] I want to know the names of the employees, who earn more than their managers

[3] I would like to know the names of the employees,who work in the writing-materials department

(CHE) (LAVORANO) (NELLA) (DIVISIONE)

"CANCELLERIA">

Second-level internal representation:

< IMPIEGATI (NOME:"CANCELLERIA", SALARIO,

DIRIGENTE, REPARTO) >

After step 3. we get:

< IMPIEGATI (NOME: "CANCELLERIA", SALARIO,

DIRIGENTE, REPARTO) >

Step 4. fails since the assignement of the special symbol P to the domain NOME, which is required by the function VORREI (pointer PV), can not be executed, being the value "CANCELLERIA" already present in the domain NOME. The following diagnostic message is therefore generated:

>analisi fallita nella fase 4

>impossiblità di effettuare un corretto assegna-

>mento del valore "CANCELLERIA" ad un opportuno
 dominio

>termini ignorati nell'analisi:
 I, CHE, LAVORANO, NELLA, DIVISIONE

>rappresentazioni interne generate: (4)
 (vedi sopra)

Let us note that,if we insert in the lexicon the new record DIVISIONE/O/1.4=1.2, it is possible to obtain the following correct parsing:

IMPIEGATI (NOME:P, SALARIO,DIRIGENTE, REPARTO:

"CANCELLERIA")

Example 4.

Input sentence:

< DAMMI I NOMI DEI FORNITORI DI "OROLOGI"> (5)

First-level internal representation:

< DAMMI/F/PV (I) (NOMI) (DEI) FORNITORI/O/2.3

(DI) "OROLOGI">

Second-level internal representation:

< FORNITORI (ARTICOLO, FORNITORE:"OROLOGI")>

After step 3.we get:

< FORNITORI (ARTICOLO, FORNITORE:"OROLOGI")>

Step 4. fails because of the same reasons as in Example 3. In this case, however,the system can correctly parse the input sentence only by inserting in the lexicon the word OROLOGI which, on the other hand,is part of the content of the data base.

5.Conclusions

In the paper the NLI system has been illustrated, with particular attention to the NLI-2 version. It is devoted to the inquiry in natural language (Italian) of a relational data base. NLI is based on a modular architecture which allows the designer , or even the user, to define his own application in a fully interactive and incremental way. The linguistic model supported by the system is that one of goal-oriented understanding,and the parsing algorithms adopted are mainly semantics directed.

The implementation of NLI-2 is now being completed and the system will be used in the future for developing the following research activities:

- definition of performance evaluation criteria for measuring the understanding capabilities of natural language systems;
- evaluation of the complexity of the parsing algorithms in relation to the complexity of the input sentence and of the scope of the comprehension;
- investigation on the flexibility of the system in relation to small variations in the structure and content of the data base and in the adopted natural language;
- definition of new system architectures allowing a full flexibility in relation to the purpose of the understanding and to the application doamin.

Appendix

This Appendix is devoted to present the sample relational data base which has been adopted in the development of NLI-2. This is concerned with the activity of a department store and it is just the same utilized in the definition of QBE[15]. The

(4) parsing failed in step 4
 the value"CANCELLERIA" can not be correctly
 assigned to an appropriate domain
 words ignored in the parsing:
 I, CHE, LAVORANO, NELLA, DIVISIONE
 internal representations generated:
 (see above)

(5) find the names of the supplyers of watches

data base is sketched below through a set of tables in which both the names of relations and domains and their internal representation(pairs of integers) is given.

1.% IMPIEGATO (1.1 NOME, 1.2 SALARIO,
1.3 DIRIGENTE, 1.4 REPARTO)

2.% VENDITE (2.1 REPARTO, 2.2 ARTICOLO)

3.% FORNITORI (3.1 ARTICOLO, 3.2 FORNITORE)

4.% TIPO (4.1 ARTICOLO, 4.2 COLORE, 4.3 TAGLIA)

Acknowledgments

I am indebt to E. Cuccurullo for the implementation of NLI-2 and for stimulating discussions.

References

1. Burger J.F, Leal A., Shoshani A. Semantic-based parsing and a natural language interface for interactive data managelent. Proc. 13th Conf. of the ACL, Boston, 1975
2. Codd E.F. A relational model of data for large shared data banks. Comm.ACM 13,6 (1970), 377-387.
3. Codd E.F. Seven steps to rendezvous with the casual user. IBM Research Report RJ1333, San Josè,Cal. ,1974.
4. Guida G. Ideas about design of natural language interfaces to query systems. Proc. of a Workshop on Natural Language for Interaction with Data Bases, IIASA CP-78-9, Laxemburg, Austria, 1978, 265-279.
5. Guida G. Natural language interfaces to computer system: An experimental project. Alta Frequenza XLVII, 9 (1978), 668-674.
6. Guida G., Somalvico M. A two level modular system for natural language data base query applications. Proc. 6th Int. Joint Conf on Artificial Intelligence, Tokio,Japan, 1979, 345-347.
7. Harris L.R. Experience with ROBOT in 12 commercial natural language data base query applications. Proc.6th Int. Joint Conf. on Artificial Intelligence, Tokio,Japan, 1979, 365-368.
8. Hendrix G.G., Sacerdoti E.D., Sagalowicz D., Slocum J. Developing a natural language interface to complex data. ACM Trans. on Data Base Systems 3,2 (1978), 105-147.
9. Kaplan S.J., Mays E., Joshi A.K. A technique for managing the lexicon in a natural language interface to a changing data base. Proc. 6th Int.Joint Conf. on Artificial Intelligence,Tokio,Japan, 1979, 463-465.
10. Lehmann H.The USL project. Its objectives and status. Proc.Int. Technical Conf. on Relational Data Base Systems, IBM Scientific Center,Bari, Italy, 1976, 7-38.
11. Mylopoulos J., Borgida A., Cohen P.,Roussopoulos H., Tsotsos J., Wong H. TORUS : A step towards bridging the gap between data bases and the casual user. Information Systems 2 (1976), 49-64.
12. Schank R.C., DeJong G. Purposive understanding. Machine Intelligence 9, J.E. Hayes, D.Michie, L.I.Mikulich (Eds.), Ellis Horwood, Chichester, 1979, 459-478.
13. Waltz D.L. Natural language access to a large data base: an engineering approach. Proc. 4th Int. Joint Conf. on Artificial Intelligence,Tbilisi, USSR, 1975, 868-872.
14. Waltz. D.L., Goodman B.A. Writing a natural language data base system. Proc. 5th Int. Joint on Artificial Intelligence, Cambridge, Mass., 1977, 144-150.
15. Zloof M.M. Query by example. Proc. Nat. Computer Conf., AFIPS Press, Vol. 44, 1975, 431-438.