Panayotis Stamatopoulos, Maria Katzouraki
NRC "DEMOCRITOS", Athens

George Philokyprou
University of Athens

A NATURAL LANGUAGE SYSTEM FOR DATA BASE QUERIES

In this paper a natural language system is presented for processing types of sentences to be used on a data base environment. The system is implemented in PROLOG using Metamorphosis Grammar rules and introduces a verb oriented logical structure for the output representation, contrary to existing natural language processors, which employ a quantifier based approach. As for the analysis stage of the implementation, efficiency is obtained by employing a two-pass processing. The implementation is adapted to a Greek language subset.

1. Introduction

Interacting directly with data bases can be achieved either by special structured query languages (SQL,QBE,SQUARE etc.) or with natural languages which conform to the user's native language. It is with this latter approach that we are dealing in this paper presenting the description of a natural language processor adapted to Greek for data base queries. Following some recent developments in natural language analysis [2,4,5,8,9], we have employed logic to deal both with the underlying formalism and the programming environment. Specifically the system is implemented entirely in Prolog [1,7], a powerful programming language based in logic. For these purposes a specially oriented Prolog system has been developed (Interpreter) implemented on a HP 21 MX minicomputer of our laboratory [6].

The general structure of the Natural Language Processor is shown in Fig.1. The processor handles restricted natural Greek language using rules of Metamorphosis Grammar (MG) formalism [3] under normal form. The application domain, a geographical data base, is only for demonstration purposes given that by no means restricts its adaptation to a variety of applications. There are mainly two ways in which the proposed system differs from previous work [2,4,5,8,9] and compares favourably with them:

a) The way the natural language analysis is performed in two phases rather than one [2,5] or three [9].

b) The logical structure of the output tree representation which is "verb" rather than "quantifier" oriented.

Besides, it represents the first adaptation of Logic Natural Language framework to Greek.

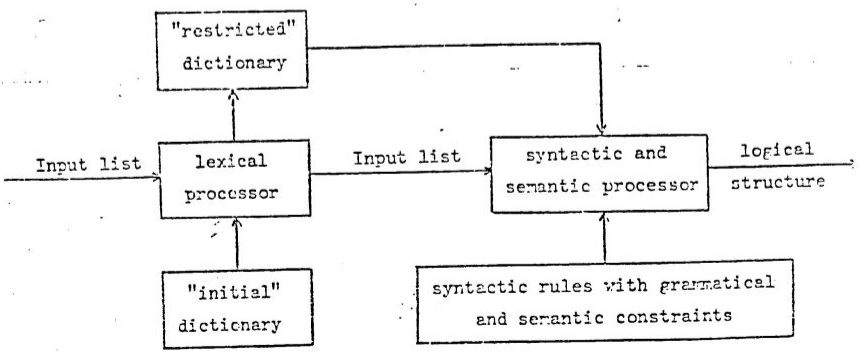

Fig.1 Natural Language Processor

2. The Natural Language World

2.1. Linguistic Coverage

The natural language subset which the system correctly interprets is determined by the Vocabulary and the Syntactic Structures it accepts. We present below the vocabulary scope and the types of sentences which are within the scope of our natural language subset.

2.1.1. Vocabulary

The vocabulary includes both general and application dependent words. Each word is classified according to its part of speech class, which in turn determines a combination of grammatical, syntactic and semantic feature-categories characteristic of that class. Below we present some of the vocabulary classes together with some examples. To this end we use the following abbreviations: g:gender, n:number, c:case, df:domain field, sot:standard output type, x:variable, ls:logical structure, d:degree. It is understood that certain of the above categories (e.g. gender, number, case) assume several values (e.g. masculine, feminine, neutral/singular, plural/nominative, genitive, accusative).

Class	Categories
a) Definite Articles	(g,n,c)
e.g. the (O)	(masculine,singular,nominative)
b) Relative Pronouns	(g,n,c)
e.g. which (OPOIA)	(feminine,singular,nominative)
c) Prepositions	(ϕ)
e.g. with (ME)	(ϕ)
d) Nouns	(g,n,c,df,x,ls)
e.g. country (XWRA)	(feminine,singular,accusative,country,x,country(x))
e) Proper Nouns	(g,n,c,df,sot)
e.g. Danube (DOYNABHS)	(masculine,singular,nominative,river,Danube)
f) Comparative Adjectives	(g,n,c,df,d)
e.g. biggest (MEGALYTERH)	(feminine,singular,accusative,country,max)
g) Verbs	(n,df_s,ls_s,df_i,ls_i,preposition$_i$,c_i,ls_v)
e.g. borders (SYNOREYEI)	(singular,country,ls_s,country,ls_i,ME,accusative, border(ls_s,ls_i))

Remarks: In the case of verbs, besides the domain field and the logical structure of the subject (df_s,ls_s), the domain fields (df_i), logical structures (ls_i), prepositions and cases (c_i) for each complement constitute distinct categories. Finally, ls_v stands for the logical structure of the sentence.

2.1.2. Syntactic Structures

The system implemented recognizes affirmative sentences of the non-elliptic type, whose nounsets can be either proper nouns or nouns, which in turn can be determined by comperative adjectives, quantifiers or relative sentences. Quanti-

fiers are the class of definite and indefinite articles, indefinite pronouns, numerals or structure forms of the kind "all the".

2.2. Application Domain

The application domain chosen is that of a wider district geography, specifically that of the Balkan States. It is essentially a relational data base organization with all information coded in the form of Prolog facts and rules, as shown in Appendix I. We believe that it should be relatively easy to adapt to different applications.

3. The Natural Language Processor

3.1. The Lexical Processor

According to the previous section, all the information regarding the categories for each word must be included in the Prolog system to build an "initial" dictionary. But Greek language (and other languages too) have the characteristic that the values assumed by some linguistic categories for many word classes are too many to be efficiently manipulated as a whole. Indicatevely we note that in Greek nouns can assume 6 different values, proper nouns 3, adjectives 18 etc. The space problems resulting from such an implementation are faced by employing the Lexical Processor, which preprocesses the sentence to be analyzed consulting the "initial" dictionary and giving as output a "restricted" dictionary with entries which are relevant only to the words of the specific sentence. The "initial" dictionary entries are Prolog facts with the same predicate, taking the form:

+Dictionary(Word Class,Word,[Category$_1$,Category$_2$,.....,Category$_n$]) (1)

In Appendix II a small part of the "initial" dictionary is shown. The Lexical Processor assigns to each word of the sentence to be analyzed all the facts of the form (1) whose second argument coincides with that word. In this way a "restricted" dictionary is produced with entries facts of the form:

+Word Class(Word,Category$_1$,Category$_2$,.....,Category$_n$) (2)

The above predication form is easily amenable to the syntactic and semantic processing of the next stage. The Prolog program implementing the Lexical Processor together with the "restricted" dictionary for the following sentence are given in Appendix III and IV respectively.

Sentence: O DOYNABHS DIARREI THN MEGALYTERH XWRA H OPOIA SYNOREYEI ME THN ELLAD

i.e. Danube flows through the biggest country which borders Greece.

Given that the linguistic categories of a word can be singled out solely on the basis of its appearance, even the "restricted" dictionary will include facts which will be of no further use.

3.2. The Syntactic and Semantic Processor

Metamorphosis grammar rules under normal form, directly translated into Prolog are the rules employed by the Syntactic and Semantic Processor for defining some

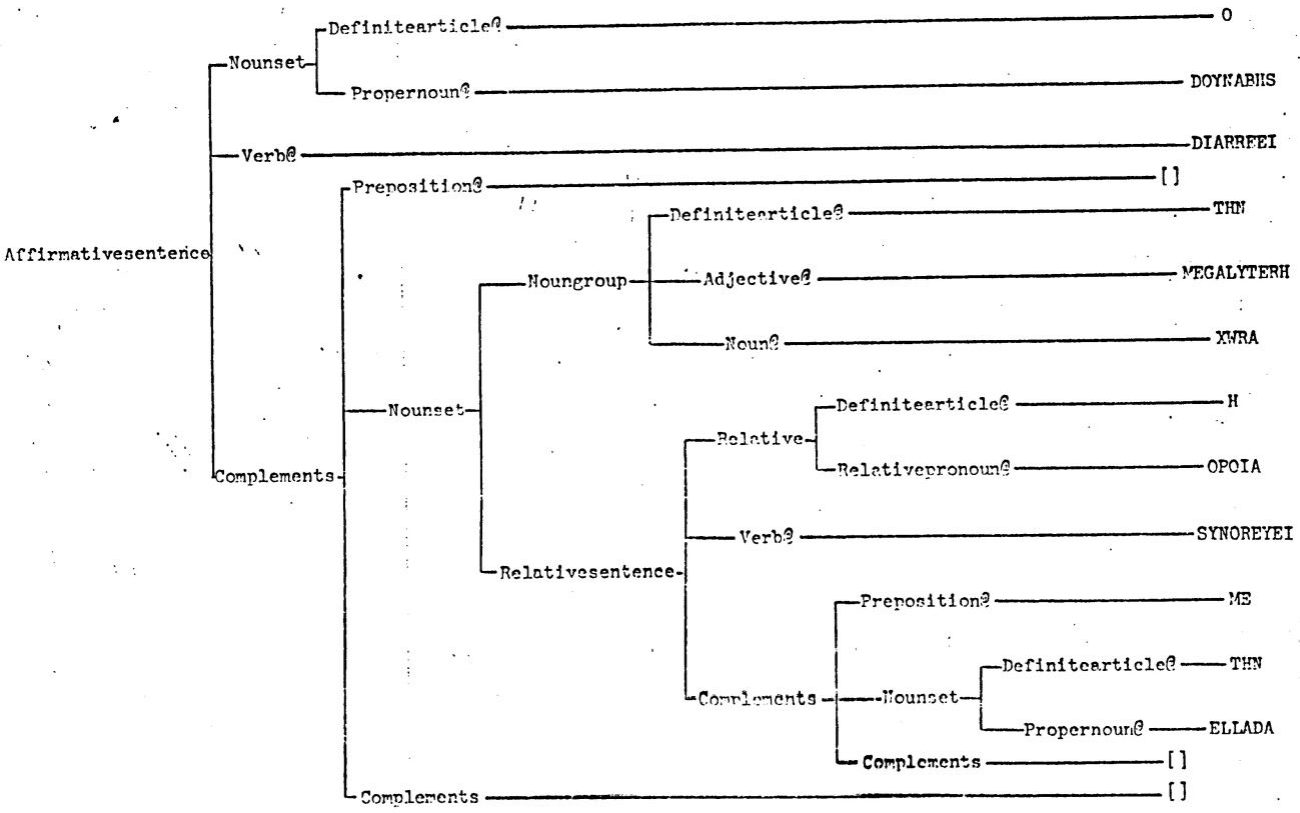

Fig.2 Deep structure tree

formal properties of the natural language. The Syntactic and Semantic Processor, consulting the "restricted" dictionary output of the Lexical Processor, performs syntactic analysis while verifying validation of the natural language grammar rules and of the semantic features of the application domain. Thus, it verifies, for example, agreement between several linguistic categories (gender, number, case) among an adjective and a noun or(and) coincidence among a verb and its subject as far as the number category is concerned etc. Metamorphosis grammar rules under normal form can be directly translated into Prolog rules as it is shown below for the case of an affirmative sentence:

Affirmative sentence(*Structure1) → Nounset(*Number,nominative,*Structure2), Verb0(*Number,*structure2,*Structure3,*Structure1),Complements(*Structure3)

+Affirmative sentence(*List1,*List4,*Structure1)-Nounset(*List1,*List2,*Number, nominative,*Structure2)-Verb0(*List2,*List3,*Number,*Structure2,*Structure3, *Structure1)-Complements(*List3,*List4,*Structure3)

Within the above rule syntactic agreement restrictions and semantic constraints can be satisfied. In Appendix V we state all the rules which are necessary for analyzing the above mentioned sentence, while the corresponding deep structure tree is shown in Fig.2. Similar rules will enforce the analysis of other types of sentences.

3.2.1. Logical Structure

Confirming with the example-case of an affirmative sentence, the output logical structure after the application of the rules is of the form:

$P(Set_s, Set_1, \ldots, Set_n)$ where P denotes the predicate associated with the main verb of the sentence, Set_s denotes the set describing the subject and $Set_1, \ldots, Set_n$ the sets describing the complements of the verb, in case they exist. The above sets can be either a list of proper nouns, a structure of the form set(variable,quantifier,property), or finally a variable previously introduced by some structure. The "quantifier" argument can be of the form the(number,howmany,comparative), where howmany ⟹ ϕ,1,2,3,..... and comparative ⟹ ϕ,max,min or some(number) or n(howmany) or all. Finally the "property" argument constitutes a set of properties associated with "and", which are satisfied by the variable introduced by the quantifier. Thus, the output structure of the example-case will take the form:

flow([Danube],set(x,the(sin,ϕ,max),and(country(x),border(x,[Greece]))))

Appendix I

+Area(Ugoslavia,255804)

+Area(Greece,131944)

+Length(Danube,2850)

+Border0(Greece,Ugoslavia)

```
+Flow(Danube,Ugoslavia)
+River(*X)-Length(*X,*Y)
+Country(*X)-Area(*X,*Y)
+Border(*X,*Y)-Border?(*X,*Y)
+Border(*X,*Y)-Border?(*Y,*X)
```

Appendix II

```
+Dictionary(Definitearticle,O,[mas,sin,nom])
+Dictionary(Relativepronoun,OPOIA,[fem,sin,nom])
+Dictionary(Propernoun,ELLADA,[fem,sin,acc,country,Greece])
+Dictionary(Noun,XWRA,[fem,sin,acc,country,*X,country(*X)])
+Dictionary(Adjective,MEGALYTERH,[fem,sin,acc,country,max])
+Dictionary(Verb,SYNOREYEI,[sin,dom(country,*S1),[agr(dom(country,*S2),ME,acc)],
            border(*S1,*S2)])
```

Appendix III

```
+Processsentence-Consult(Dictionary)-Line-Write("Give input sentence:")-Line
   -Readsentence(*L)-Searchforwords(*L)
+Readsentence([*X|*L])-Read(*X)-Readsentence(*L)
+Readsentence([])
+Searchforwords([])
+Searchforwords([*X|*L])-Process(*X)-Searchforwords(*L)
+Process(*X)-Dictionary(*P,*X,*L)-Univ(*Y,[*P,*X|*L])-Not(*Y)-Assertz(*Y)-Fail
+Process(*X)
+Not(*X)-*X-/-Fail
+Not(*X)
```

Appendix IV

```
+Definitearticle(O,mas,sin,nom)
+Definitearticle(THN,fem,sin,acc)
+Definitearticle(H,fem,sin,nom)
+Relativepronoun(OPOIA,fem,sin,nom)
+Relativepronoun(OPOIA,neut,plur,acc)
+Preposition(ME)
+Propernoun(DOYNABHS,mas,sin,nom,river,Danube)
+Propernoun(ELLADA,fem,sin,acc,country,Greece)
+Noun(XWRA,fem,sin,acc,country,*X,country(*X))
+Adjective(MEGALYTERH,fem,sin,acc,country,max)
+Verb(SYNOREYEI,sin,dom(country,*S1),[agr(dom(country,*S2),ME,acc)],border(*S1,*S
+Verb(DIABREEI,sin,dom(river,*S1),[agr(dom(country,*S2),ϕ,acc)],flow(*S1,*S2))
```

Appendix V

```
+Affirmativesentence(*L1,*L4,*S1)-Nounset(*L1,*L2,*N,nom,*S2)-Verb?(*L2,*L3,*N,
   *S2,*S3,*S1)-Complements(*L3,*L4,*S3)
+Nounset(*L1,*L3,*N,*C,dom(*D,[*W]))-Definitearticle?(*L1,*L2,*G,*N,*C)
   -Propernoun?(*L2,*L3,*G,*N,*C,*D,*W)
+Nounset(*L1,*L3,*N,*C,*S1)-Noungroup(*L1,*L2,*G,*N,*C,*D,*X,*Q,*S2)
   -Relativesentence(*L2,*L3,*G,*N,*D,*X,*Q,*S1,*S2)
+Complements(*L1,*L1,[])
+Complements(*L1,*L4,[agr(*S,*P,*C),*R])-Preposition?(*L1,*L2,*P)-Nounset(*L2,
   *L3,*N,*C,*S)-Complements(*L3,*L4,*R)
+Noungroup(*L1,*L4,*G,*N,*C,*D,*X,the(*N,ϕ,*N),*S)-Definitearticle?(*L1,*L2,*G,
   *N,*C)-Adjective?(*L2,*L3,*G,*N,*C,*D,*N)-Noun?(*L3,*L4,*G,*N,*C,*D,*X,*S)
+Relativesentence(*L1,*L4,*G,*N,*D,*X,*Q,dom(*D,set(*X,*Q,and(*S1,*S2))),*S1)
   -Relative(*L1,*L2,*G,*N,nom)-Verb?(*L2,*L3,*N,dom(*D,*X),*S3,*S2)
   -Complements(*L3,*L4,*S3)
+Relative(*L1,*L3,*G,*N,*C)-Definitearticle?(*L1,*L2,*G,*N,*C)-Relativepronoun?
   (*L2,*L3,*G,*N,*C)
```

References

1. W.Clocksin, C.Mellish, "Programming in Prolog", Springer-Verlag, Berlin, Heildelberg, New York, 1981.
2. H.Coelho, "A program conversing in Portuguese providing a library service", Ph.D. Thesis, University of Edinburgh, 1979.
3. A.Colmerauer, "Les grammairs de metamorphose", Groupe d'Intelligence Artificielle, Université d'Aix-Marseille, 1975.
4. A.Colmerauer, "An interesting subset of natural language", in "Logic Programming", K.Clare, S.-A.Tärnlund (eds.), Academic Press, 1982, pp.45-66.
5. V.Dahl, "Translating Spanish into logic through logic", American Journal of Computational Linguistics, vol.7, num.3, July-September 1981, pp.149-164.
6. M.Katzouraki, M.Gavalas, P.Stamatopoulos, A.Arvillias, "A flexible Prolog interpreter for minicomputers" (in Greek), Proc. of 1st Panhellenic Conference in Informatics, vol.2, Athens, 1984, pp.238-244.
7. P.Roussel, "PROLOG: Manuel de référence et d'utilisation", Groupe d'Intelligence Artificielle, Universite d'Aix-Marseille, 1975.
8. S.Szpakowicz, "Syntactic analysis of written Polish", Iinf UW Report, Nr 64, 1977.
9. D.Warren, F.Pereira, "An efficient easily adaptable system for interpreting natural language queries", DAI Research Report, No.155, University of Edinburgh, 1981.