PROCESSING CLINICAL NARRATIVES IN HUNGARIAN

Gábor Prószéky
National Educational Library and Museum
Computer Department
Honvéd u. 19.
H-1055 Budapest
HUNGARY

ABSTRACT

This paper describes a system that extracts information from Hungarian descriptive texts of medical domain. Texts of clinical narratives define a sublanguage that uses limited syntax but holds the main characteristics of the language, namely free word order and rich morphology. We offer a fairly general parsing method for free word order languages and the way how to use it for parsing Hungarian clinical texts. The system can handle simple cases of ellipses, anaphora, unknown words and typical abbreviations of clinical practice. The system translates texts of anamneses, patient visits, laboratory tests, medical examinations and discharge summaries into an information format usable for a medical expert system. Similarly to this expert system, the information formatting program has been written in MPROLOG language and its experimental version runs on PROPER-16, a Hungarian made (IBM-XT compatible) microcomputer.

1. OVERVIEW

In the past few years we have developed a computational system to analyze Hungarian texts using a morphological analyzer (Prószéky et al 1982) and a general parsing program called ANAGRAMMA (=ANAlytic GRAMMAr) (Prószéky 1984). The whole system for information formatting is based on these modules and consists of five consequent parts: (i) morphological analysis, (ii) normalization, (iii) parsing, (iv) evaluation, (v) mapping into the information format. The last block is an operation that converts the output of ANAGRAMMA, which is (ii)+(iii)+(iv), to a format that can be used by a medical expert system. The approach leads to a structure shown by Figure 1.

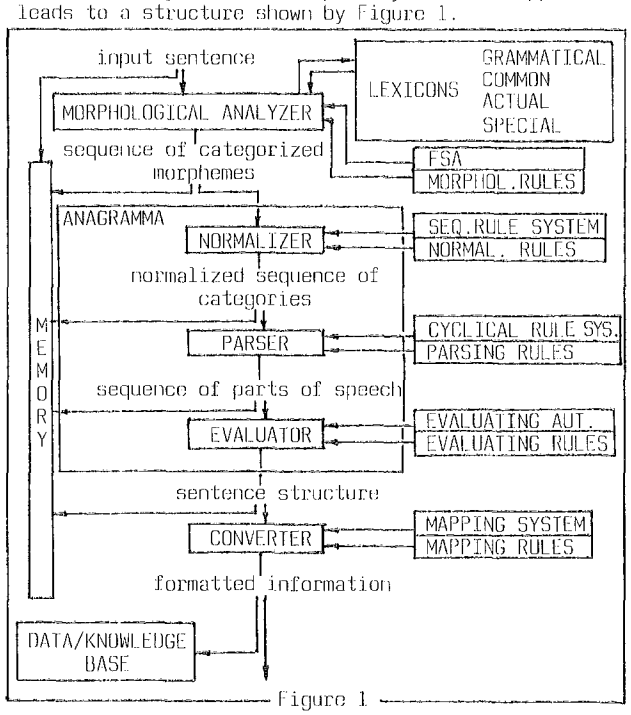

Figure 1

2. MORPHOLOGICAL ANALYSIS

The first phase is the morphological analysis of word forms. Hungarian is a free word order language, therefore the role of suffixes is very important from the viewpoint of identifying phrasal constituents. A lot of syntactic and semantic information (number, person, possession, case, tense, mood etc.) are carried by these elements. The concatenation of stem and suffixes is sometimes rather complex: there are suffixes that have different stem--dependent forms and stems that have different suffix-dependent forms. Thus the lexicon must contain all the possible variants of the stems as independent entries or we have to define an algorithm for constructing the real stems from the archiphonemes of the lexicon. We have chosen the former alternative.

The lexicon consists of four parts but only conceptually. From the point of view of the algorithm, it is an integral whole. The reasons why we distinguish its parts are as follows:

(i) All the NL processing programs of an agglutinative language must know all the grammatical morphemes of the language.

(ii) The dictionary of common expressions is not necessary but it is a useful part of all NL systems. This module can be enlarged by the user.

(iii) The actual lexicon contains more or less all the lexical elements that is needed for the actual type of application (DB querying, updating, information extraction, translation etc.).

(iv) The special lexicon contains terms of the actual application field (in our case the terms of medical science). This module can, of course, be enlarged by the user.

After updating the lexicon, entries will be arranged in alphabetical order.

The morphological analyzer is a finite state automaton (FSA). It takes a word to be analyzed from the input sequence of words and searches the dictionary in order to find the input word. If the left part of the word matches a dictionary entry, the entry's informational part must be copied to the working buffer. The content of this buffer will be the input to the syntactical analyzer. Then the automaton begins to work from right to left. Its output is the sequence of the informational parts of the grammatical morphemes standing after the stem we identified a short while ago. If the information of the stem and the suffixes are not compatible or there remained an unprocessed part in the word, the algorithm tries to analyze the word as a compound once more and if this process fails then it asks the user what to do. Figure 2 is an illustration of this process. (The "origin" of the entries is marked by G, C, A and S, that is grammatical, common, actual and special lexicon, respectively.)

3. PROBLEMS OF PARSING

The well known methods generally utilized for parsing NLs are not convenient for treating languages like Hungarian, Finnish, Esthonian or Japanese, cf. (Nelimarkka et al 1984),(Tsujii et al 1984),(Prószéky 1984). In these languages, the suffixes carry out most of the

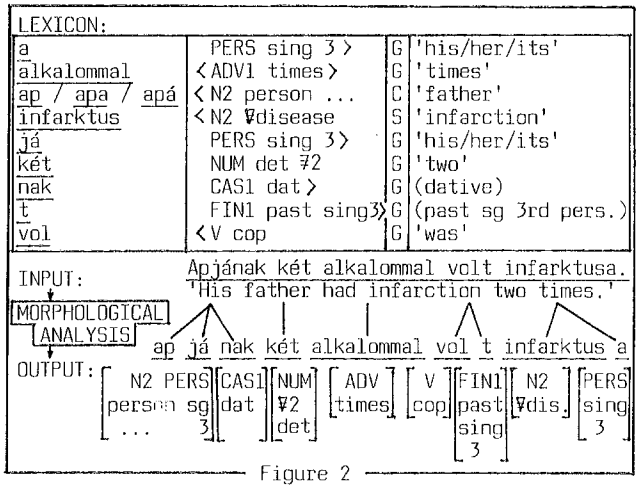

Figure 2

task of marking grammatical function, therefore, the word order -- strictly speaking, the phrase order -- will be relatively free. So we must turn our attention to (i) the internal structure of the phrases and (ii) the order of phrases (and the intonation, of course only in speech) that plays an important role in expressing communicative functions.

The basic idea of the strategy we propose builds on the invariants of the sentence structure of free word order languages, that is, (i) the first thing to do is to recognize the internal structure of the parts of speech and (ii) the second is to interpret their relative order. This order is connected with the communicative roles (topic, focus etc.) of the structure.

The syntactic analysis of free word order sentences is based upon the morphemes identified by morphological analysis. The lexicon cannot help us to give the actual functional role of a morpheme because of two reasons:

(i) All possible functional roles of a morpheme cannot be listed.

(ii) If there were several possible roles in the description of morphemes nobody would know which of them to use actually.

4. UNKNOWN ELEMENTS

The problems of the unknown elements can arise not only in the case of computational analysis, since people may read/hear morphemes never read/heard before, yet they can identify the actual syntactic role of them without any knowledge of any previous syntactical categorization. The category or word class of a word is statistical information about its occurrence in particular syntactic positions. For example, the word 'beteg' can be a noun ('patient') or an adjective ('sick','ill') in Hungarian. It is an adjective in adjectival use, that is without inflections or before adjectival suffixes:

'Előzőleg soha nem volt beteg.'
('He has never been ill before.')
'Hat napja fekszik betegen.'
('He has been laid up since six days.')

The same morpheme can, however, be a noun before nominal suffixes:

'A betegnek nem volt infarktusa.'
('The patient has had no infarctions.')

Although we consider categorization as a syntactic generalization, we do not claim that there are no independent syntactical categories. In agglutinative languages such categories are e.g. the nominal suffixes just mentioned. These categories are not arbitrary, because one cannot introduce a new suffix to the language, but can, however, use new stems in the sentence. If the parser knows these regularities, then lexical categories will be used for control only.

5. SENTENCE STRUCTURE IN AGGLUTINATIVE LANGUAGES

Below we will make use of Hungarian examples to show the most important properties of a typical agglutinative language. In a simple sentence there can be only one finite verbal suffix. If we have a sentence containing two of them, then we have to do with co-ordinate clauses or one sentence with a subordinate clause. Naturally, the finite suffix is immediately preceded by a verbal stem. If the sentence has no finite verbal suffixes, (i) it contains a 0-copula that is rather frequent, not only in medical texts but also in the every-day Hungarian or (ii) there is ellipsis in the sentence. The non-finite verbal suffixes are also preceded by a verbal stem. These elements can behave differently according to whether or not they influence the word order of other elements.

We consider the noun as an element that stands before a nominal suffix group. Sometimes the lexicon does not categorize theis morpheme as a noun. We consider this situation as a case of a missing noun. Regeneration of missing elements is important because of identifying elliptical constructions. For example, Hungarian adjectives can have nominal endings when no noun occurs in the structure.

As it seems, most of the morphemes do not have a fixed lexical category, because their positions in the sentence actually define their functional role. But we have some important lexical features:

(i) Stems . They are closed morphologically to the left and open to the right (formally: <stem). "Open" means an ability to join other elements. In the case of noun-like ones these "other" elements are, for example, the case suffixes.

(ii) Suffixes. They are closed morphologically to the right and open to the left (suffix>), e.g. the case endings.

(iii) Open endings. They are open morphologically on both sides (open), e.g. the morphemes marking plurality or possessivity.

(iv) Closed elemnts. They are closed on both sides (<closed>), e.g. adjectives, numerals, adverbials.

So, if a closed side immediately precedes an open one or an open one a closed one, the parser has to correct the "wrong" sequence inserting an empty morpheme:

(a) <stem <closed> → <stem suffix><closed>

(b) <closed> suffix>→<closed><stem suffix>

Instance (a) can be, for example, a genitive case-insertion (as this case ending can sometimes have an empty form in Hungarian) and instance (b) can be a noun insertion between an adjectival stem and a nominal suffix.

6. PARTS OF SPEECH

The surface scheme of a Hungarian sentence is the following:

(<A>*<S N>*<V NF>*)*<V F> (<A>*<S N>*<V NF>*)*

where A stands for adverbials, S for nominal and V for verbal stems, N for nominal case endings, F for finite and NF for non-finite verbal suffixes. Hence the types of the constituents are as follows:

(i) independent adverbials (without any suffix),

(ii) non-finite verbs (e.g. infinitive, gerund),

(iii) nominal groups with case ending,

(iv) a verb plus a finite suffix (the main verb of the sentence).

Having made clear the internal structure of the constituents, the parser can deal with the formal evaluation of the connections between the constituents (e.g. verb and complements, possessives and possessors

etc.). In the first part of the parsing we do not need any S-symbols, more precisely, any string over a particular set, the parts of speech, can serve as S-symbol. The main parts of speech can be described with a help of the schemes (i)-(iv), but in fact, only (ii) and (iii) are important. Adverbials of type (i) usually consist of one element and every sentence has one and only one structure of type (iv). Sentences rather frequently consist of more than two constituents, but in a free word order language there is no typical primary order of these constituents. Our method is based on this observation. We do not describe the structural relations in the sentence sequentially from the left to the right end of the sentence. Our rules form blocks and these blocks are used in an order depending on the elements of the actual sentence.

7. ANAGRAMMA

Between the morphological analysis and the parsing we need a normalization procedure that inserts the missing morphemes on the basis of the formal lexical properties of the elements of the input string. If the input string includes interjective signs or words and (i) this sign means subordination, then it seems to be obvious to take the embedded string out and handle it like an independent sentence, or (ii) this sign or word means co-ordination, then we will elaborate the co-ordinate structures parallelly.

So, to analyze a simple sentence of Hungarian, the ANAGRAMMA parser would begin with the question 'Is there any substring of the sentence to be parsed that has the form of the first rule's left hand side?'. If the answer is 'yes', the right side of the same rule is substituted as many times as the substring occurs in the sentence. For example:

```
The rule:          ADJ N2 --> N1
The sentence
to be parsed:  DET ADJ N2 CAS1 DET ADJ N2 CAS1 V FIN1
The result:    DET  N1   CAS1 DET  N1   CAS1 V FIN1
```

If the sentence does not contain the substring, the next rule follows. In this way all rules can be applied once only, although we would probably have to use them more than once. The repeated use of the rules can be realized with the help of cycles:

```
    3:      ...
    4:  CAS1 V ADJ1 --> ADJ
    5:       ADJ N2 --> N1
    6:       DET N1 --> N
    7:       N CAS1 --> CAS
    8:     ADJ CAS1 --> CAS
    9:       V FIN1 --> FIN
   10:      ...
```

Figure 3

The kernel of the cycle is a sequential rule package and its condition is the quantity of rules applied at the last pass over the cycle. If it is not 0, then the algorithm continues at the first rule of the package. If it is 0, that is, there were no such applications, the rule of the next number has to be applied.

A trace of an ANAGRAMMA parsing:

```
    DET DET ADJ N2 CAS1 V ADJ1 N2 CAS1 DET ADJ N2 CAS1 V FIN1 .
5:  DET DET  N1    CAS1 V ADJ1 N2 CAS1 DET  N1    CAS1 V FIN1 .
6:  DET  N   N1    CAS1 V ADJ1 N2 CAS1  N   N1    CAS1 V FIN1 .
7:  DET CAS  N1    CAS1 V ADJ1 N2 CAS1 CAS  N1    CAS1 V FIN1 .
4:  DET CAS ADJ    CAS1 V ADJ1 N2 CAS1 CAS ADJ    CAS1 V FIN1 .
5:  DET CAS  N1    CAS1 V ADJ1 N2 CAS1 CAS  N1    CAS1 V FIN1 .
6:  DET  N   N1    CAS1 V ADJ1 N2 CAS1  N   N1    CAS1 V FIN1 .
7:  DET CAS  N1    CAS1 V ADJ1 N2 CAS1 CAS  N1    CAS1 V FIN1 .
9:  DET CAS  N1    CAS1 V FIN1 N2 CAS1 CAS  N1    CAS1 V FIN1 .
```

Figure 4

The parsing is over if (i) all elements of the actual string to be parsed are from the distinguished set (e.g. CAS and FIN in the above example), or (ii) the algorithm is after the last rule and there is no acceptable cycle-end after this rule. We say that the algorithm cannot interpret the sentence if there have remained other than distinguished elements. The parser can operate more quickly if the rules in the same package give the description of the same grammatical phenomenon. Such modules consist of rules the left sides of which are similar. If a package contains only rules whose left side does not contain any element of the sentence to be parsed, then it can be omitted. We can use this method of simplification without much ado, owing to an X-like formalism that guarantees that no new symbols can be born as a result of application of the rewriting rules. We use decreasing bar levels alike the formal derivation process does with exponents.

8. EVALUATION

The evaluation module is essentially a pattern matching algorithm that identifies the link between (i) the predicates and their arguments, (ii) the anaphoric elements and their antecedents, and (iii) the "parallel" structures separated by the normalizer. The lexical forms of predicates contain the surface case endings and the semantic role of the needed constituents, therefore the algorithm has to look for these constituents and order the new features given to them by the predicate. The identification of the antecedents of anaphoric elements is similar, but antecedents often occur in previous sentences. Therefore the evaluator can set up a connection with the analyzed form of the same paragraph.

9. MAPPING INTO INFORMATION FORMAT

After some consultation with physicians it was possible to establish the specialized concept classes and the patterns of the concept class co-occurrence from which the information format could be defined. The nouns in the lexicon are subcategorized by their membership in these classes. Most classes are mapped into the appropriate slot, because the names of the classes are the labels of the slots of the frames used by the expert system. Figure 5 shows the form of the formatted text:

```
ANAMNÉZIS
 GENETIKUS-FAKTOR   FOK          apa
                    BETEGSÉG     ischaemias szívbetegség
 KÓRELŐZMÉNY        MI           szorító fájdalom
                    HOL          mellkas
                    MIKOR        fizikai terhelésre
                    GYAKORISÁGA
 KEZELÉS-ELŐZMÉNY               nitrát beta-blokkoló
```

Figure 5

10. REFERENCES

Kálmán,L.– G.Prószéky 'FMR Grammar' Work.Pap.of Inst.of Ling. Nr.1., 31-41 (1985).

Nelimarkka,E.– H.Jäppinen – A.Lehtola 'Parsing an Inflectional Free Word Order Language' Proc. ECAI-84, Pisa, 167-176 (1984).

Prószéky,G. 'ANAGRAMMA: A Parsing Strategy and Grammar for Agglutinating Languages' Abstr. AIMSA-84, Varna (1984).

Prószéky,G.– Z.Kiss – L.Tóth 'Morphological and Morphonological Analysis of Hungarian Word Forms by Computer' Computational Linguistics and Computer Languages Vol.15.,195-228 (1982).

Sager,N. 'Natural Language Processing' Addison-Wesley, Reading, Mass. (1981).

Tsujii,J. – J.Nakamura – M.Nagao 'Analysis Grammar of Japanese in the MU Project' Proc. COLING-84, Stanford, 267-274 (1984).

Yang,Y. – T.Nishida – Sh.Doshita 'Use of Heuristic Knowledge in Chinese Language Analysis' Proc. COLING-84,Stanford,222-225 (1984).