Realisation of Syntactic Parser for Inflectional Language using XML and Regular Expressions*

Marek Trabalka and Mária Bieliková

Slovak University of Technology
Department of Computer Science and Engineering
Ilkovičova 3, 812 19 Bratislava, Slovakia
{trabalka,bielik}@dcs.elf.stuba.sk

Abstract. In the paper we present a method of syntactic parsing for inflectional language. Method consists of several steps including morphological and syntactical levels of analysis. We proposed bottom-up model of syntactic analysis of the sentence. Its advantage is in the case of ill-formed sentence because the analyser is still able to parse at least parts of the sentence. We describe also experimental implementation of the proposed method, which is based on the use of XML and regular expressions.

1 Introduction

Syntactic parsing plays an important role in natural language processing. Most of today's research and practice in syntactic parsing is concerned with the English language. The concentration on English has resulted in advances in solving of its linguistic problems. However parsing text written in other languages often requires solving different problems. Performance of syntactic parsers for the English language is often not satisfactory in particular for inflectional languages [4].

Inflectional languages (e.g., Slavonic languages such as the Slovak, Czech, or Russian languages) use more of morphology than English does. Words in these languages have usually several different morphological forms that are created by changing a suffix. Each form represents some of morphological attributes like case, number or gender. Morphology in inflectional language strongly influences syntax. In the English sentence, meaning is determined by order of words while e.g., in Slovak it depends primarily on the word form and word order is not so strict. On the other hand, syntactical analysis of the sentence for inflectional language is after successful morphological analysis simpler task than in English.

In the paper we present a method of syntactic parsing for inflectional language. Method consists of several steps including morphological and syntactical levels of analysis. We proposed a bottom-up model of syntactic analysis of the sentence. This approach is suitable in particular for parsing ill-formed sentence

* The work reported here was partially supported by Slovak Science Grant Agency, grant No. 95/5195/605 and project INCO Copernicus, No. 977069, European Research Network for Intelligent Support of Electromyographic Studies: EMG-Net.

because the analyser is still able to parse at least parts of the sentence. It can be also used for parsing correct text. One of the positive features of our approach is that it does not heavily depend on a morphological lexicon. It can be used also with existing approaches for morphological analysis (e.g., with existing spell-checkers to improve grammar checking that should consider the context of the whole sentence). The proposed method is suitable for use in a situation where unknown word form has to be analysed for its morphological category. We describe an experimental implementation of the proposed method, which is based on the use of XML and regular expressions.

2 Bottom-up syntactic analysis

Relations between words and clauses in a sentence written in inflectional language are primarily expressed by an agreement on the equality of some morphological categories. For example a noun phrase in the Slovak language can be formed by a noun, preceded by one or more adjectives where these adjectives have the same gender, case and number. Such rules regarding regular phrases and sentences influence connections between words and disambiguate their alternative forms suggested by morphology.

For languages with a strong word order usage of top-down parser is highly recommended because of its prediction features. On the other hand the order of sentence constituents in the inflectional language is not strict and often is driven more by human intuition than by firmly given regulations specified by linguists [4]. Therefore, design of the top-down syntactic parser for inflectional language requires consideration of many possible combinations of words in the sentence. In this case, bottom-up syntactic parsing is more suitable.

In the inflectional language it is possible to determine role of the word even at the centre of the sentence when no analysis of previous words has been made. We use morphological categories of words and rules for their connecting into the phrases to built larger and larger phrases and finally combine them into a sentence. Words are not processed from the left to the right (such as in [2]), but processing depends on their potential role and priority of a syntactic rule used.

In the process of syntactic analysis a parsing graph is created. A sentence is represented by the graph of connected words. At the beginning of the process, all possible morphological categories of each word are found. After this analysis, an initial parsing graph is created: all morphological categories become nodes, and edges are created between all forms of two neighbourhood words. Next, a new node is created using syntactic rules about connection forms into a phrase. Each new node is connected to all predecessors of the first node of the created phrase and also to all successors of the last node of this phrase. The parser has to remember application of each rule on each combination of words and finish when there is no other applicable rule.

The process of analysis can be finished also when such a top node is created that has neither predecessors nor successors, i.e. the node representing the whole sentence is created. In this case the parsing process was successful.

The procedure of parsing consists of the following steps:

1. Perform morphological analysis on each word from the sentence (base form and grammatical categories of each word are determined; result can be ambiguous).
2. Create list of nodes (a node represents base form together with grammatical categories; for each morphological alternative the separated node is created). Create oriented edges that concatenate nodes representing neighbouring words in the sentence.
3. Put all syntactic rules into the set of applicable rules and sort them by defined priority.
4. If the set of applicable rules is empty, go to the step 9.
5. Select the first applicable rule. Let's its length is n.
6. Try to unify each n-tuple of concatenated nodes with the right side of the selected syntactic rule. Try only that n-tuples that were not tested with that rule before.
7. If unification is successful, create a new node(s) using the left side of the rule with bound variables. Create oriented edges from each predecessor of the first node of the original n-tuple that ends in the first node of the new m-tuple. Similarly create also edges from the last node of the new m-tuple to all successors of the original n-tuple.
8. If there were some successful applications of the rule, continue by the step 3., otherwise follow the next step.
9. If there is a node without neither predecessors nor successors, parsing was successful. Such node represents full sentence and parsing tree can be created by analysing history of node creations. If there is no such node, parsing was unsuccessful.

Figure 1 illustrates the parsing graph of the Slovak sentence *"Pekný veľký strom rastie"* ("A nice big tree grows"). Based on the syntactic rule, the node `NOUNPHRASE(N,sg,m)` from the nodes `ADJ(sg,m)` (word *veľký*) and `NOUN(N,sg,m)` (word *strom*) is created. Next, this phrase is connected with the adverb *pekný* (its form `ADV(N,sg,m)`) and another noun phrase is created. Finally this phrase is connected to the verb *rastie* in the form `VERB(3,sg)` and the node represented the whole sentence `SENTENCE()` is created. The sentence was parsed successfuly.

The main advantage of the proposed bottom-up syntactic parser is minimising necessary processing of rules over different groups of nodes (words in a sentence). Efficiency of the analysis is influenced by the quality of morphological analysis and its performance.

3 Morphological analysis

Syntax of inflectional languages strongly depends on morphology, i.e. word form influences potential syntactic role of word in a sentence. Morphology in such languages is based on suffixes that combine several morphological categories, e.g. gender, case or number. Our approach to morphological analysis is based

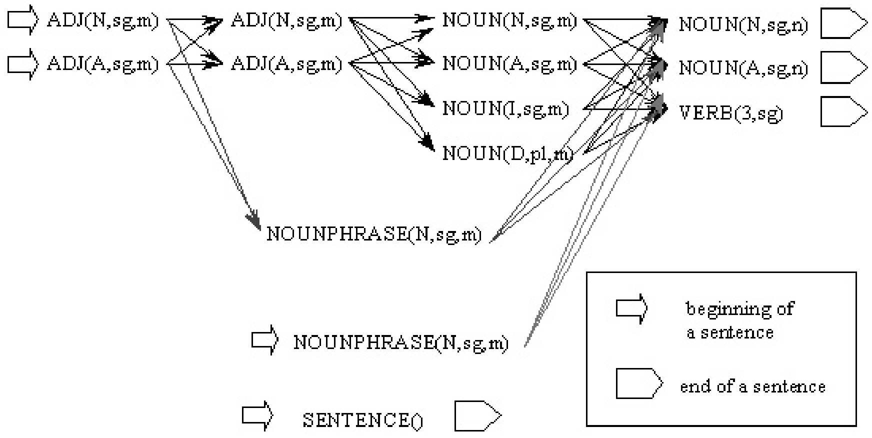

Fig. 1. An example of the parsing graph.

on subsequent refining partial results. Our aim was to support the analysis of unknown words.

We proposed two levels of the morphological analysis. The first level is based on suffix analysis and the second level on the definition of paradigms, which represent different word forms. To automate the process of binding an unknown word to appropriate paradigm we devised a method of adaptive morphological analysis [6]. The method uses paradigms and heuristics to compute an acceptance probability of the particular word. The acceptance probability expresses a degree of belief in the correct determination of paradigm for the word.

Input to the method is a word, which is subject to the morphological analysis. Output is determination of possible forms for this word (morphological tags) with an acceptance probabilities for each tag alternatives of the analysed word. These alternatives and their acceptance probabilities are computed by means of linguistic knowledge. Linguistic knowledge is either filled in advance by an expert, or learned by the system during the previous analysis.

For purposes of syntactic analysis we finally do not need a base form of the word. To improve effectiveness of a paradigm check we proposed an advanced suffix analysis, which precede our adaptive morphological analysis. It is used when the analysed word is not found in the morphological lexicon. Suffix analysis is not a completely new idea. Many authors use such approach to increase performance of lemmatisation, but they usually check only grammatical suffix. However, in this case there is high ambiguity of the analysis.

In fact, many words have typical endings, which determine some of their morphological categories. However, these typical endings often contain also some letters before the actual grammatical suffix. Our approach is applicable also for checking letters before the grammatical suffix. For example, the (nongrammatical) suffix *-pcia* is in the Slovak language typical for famine nouns in singular nominative. On the other hand, the grammatical suffix *-a* can determine verb, noun, or adjective in various numbers, cases and genders.

The main advantage of the proposed approach is that the process of suffix analysis is straightforward and fast. It can also disambiguate some alternatives in very early stage of analysis. However, such analysis can produce more results (possible word categories), because not all suffixes are unambiguous. In order to perform suffix analysis we checked a stock of language words and defined meaningful suffixes into a lexicon.

4 XML and regular expressions

The proposed method of syntactic analysis is realised using two technologies: XML and regular expressions. XML (eXtensible Markup Language) was developed as a universal tool for describing structures and documents [1]. We use XML to describe state of the process of analysis, lexicons and all necessary static structures. Regular expressions are used to represent rules, i.e. dynamic character of the parser. Combination of these two approaches provides interesting and robust features. It enables the unified representation of static and dynamic aspects of the natural language processing.

Static part (XML) is represented mainly by lexicons (morphological lexicon, paradigms definitions, suffix lexicon) and texts. It can be considered as knowledge base together with working memory of the knowledge-based system. There are many ways how different authors solve the storage of these data and knowledge. They often use some self-defined text formats or database tables (see e.g. how word can be tagged [3]).

Dynamic part (regular expressions) determines the way of change, manipulation, or transformation of the static part. We exploit the fact that the arbitrary change can be considered as transformation of strings. Such dynamic part is meaningful only if there exists a text form of processed entities. XML is an ideal candidate for this purpose.

Bellow we illustrate several examples. Each word form in a node of the parsing graph is represented by XML string, e.g.

```
<NOUN paradigm="otec" case="4" number="sg" gender="m">mládenca</NOUN>
```

Rules are described using regular expression substitutions. They are represented by a tag `RULE` which consist from two parts:

```
<RULE> s{left-hand-side}{right-hand side} </RULE>
```

The left hand side of the rule determines text to be substituted, the right hand side specifies the text used for substitution. An example bellow illustrates a syntactic rule that combines attribute and nounphrase into the nounphrase:

```
<RULE cf="8.5">
 s{ <ATTRIBUTE
        (animate="\w*") (case="\d*") (number="\w*")(gender="\w*")>(\r*)
    </ATTRIBUTE>
```

```
    <NOUNPHRASE \1 \2 \3 \4 (person="\d*")
      (compoundnumber="\w*")>(\r*)
    </NOUNPHRASE>
  }
  { <NOUNPHRASE $1 $2 $3 $4 $6 $7>$5 $8</NOUNPHRASE>
  }m
</RULE>
```

Rules are exploited on all levels of the analysis. On the morphological level, rules are used for paradigm specification. In this case a rule specifies transformation between different forms of the word. On the syntactical level rules describe transformation of morphological categories of words into phrases and phrases into the sentence.

5 Conclusions

The proposed syntactic parsing method is aimed to use for the Slovak language. It can be also used for other languages with a rich morphology. We emphasise differences against languages with a weak morphology where different algorithms and methods are used.

We implemented the proposed method of bottom-up syntactic parsing for performing grammar checking of a sentence [5]. The system combines words into phrases and when it finally achieves only one top node, parsing is successful and sentence is correct. On the other hand, when there are still more nodes after analysing all combinations, sentence is probably incorrect.

Our approach takes advantage of refined processing of the sentence in several adjoin phases. The analysis exploits the morphological information of particular suffixes. Suffix analysis often produces ambiguous results although in the particular context the word has often only one meaningful form. This limitation is resolved by including the adaptive morphological analysis [6] and the syntactic analysis that reduce the number of possible alternatives.

References

1. Bradley. N. The XML companion. Addison Wesley, Harlow, England, 1998.
2. Hausser, R.: Newcat: Parsing Natural Language using Left-Associative Grammar. Berlin, Springer-Verlag, 1986.
3. Nenadic, G., Vitas, D. Using Local Grammars for Agreement Modeling in Highly Inflective Languages. In Proc. of TSD'98, P. Sojka, V. Matoušek, K. Pala, I. Kopeček, I. (Eds.), Masaryk University Press, pp.91-96, 1998.
4. Smrž, P., Horák, A. Implementation of Efficient and Portable Parser for Czech. In Proc. of TSD'99, V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (Eds.), Springer Verlag, LNAI 1692, pp. 105-108, 1999.
5. Trabalka, M. Grammar-checker for the Slovak Language. Diploma Thesis. Slovak University of Technology. 1999. (in Slovak)
6. Trabalka, M., Bieliková, M. Performing Adaptive Morphological Analysis Using Internet Resources. In Proc. of TSD'99, V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (Eds.), Springer Verlag, LNAI 1692, pp. 66-71, 1999.