THE UTILIZATION OF COMPUTATIONAL LINGUISTIC RESOURCES IN MORPHO-SYNTACTIC DISAMBIGUATION[1]

Gîsca Veronica
Institute of Mathematics and Computer Sciences of Academy of Sciences of Moldova
veronica.gisca@gmail.com

Abstract. *Syntactic structure of language can be defined by context-free grammars. Context-free grammars do not facilitate generation of context sensitive aspects, for example, the agreement between different parts of a sentence. This results in a generation of ambiguous sentences. To solve morpho-syntactic disambiguation we have proposed attribute grammars (AG). AG are the extension of context-free grammars, where attributes are associated with grammar symbols, and semantic rules define values of the attributes. An advantage of using the attribute grammar is in solving the ambiguity problem which constitutes a link between syntax and semantics. This link which is achieved with use attributive grammars widens the volume of information that is used for disambiguation.*

Keywords: *context-free grammar, attribute grammars, morpho-syntactic disambiguation, systems attributes, semantic rules.*

I. Introduction

Natural language modeling, natural language processing is a rather lengthy process that involves detailed analysis of basic rules of communication.

A brequently encountered problem is the one of ambiguity. While people easily solve the problem of disambiguation, computational techniques are not sophisticated enough.

Computer operates strictly embodied elements, with algorithms and mathematical models well determined. For this reason, attempts to represent natural language by formalisms understood by the computer are made. To solve the disambiguation at morpho-syntactic level the formalism of attribute grammars (AG) is proposed.

Attributive grammars combine organic with context-free grammar, unlike the unifying mechanism (PATR), which actually uses only synthesized semantic attributes, attributive grammars can be used and inherited attributes. Using inherited attributes allow implementation contextual dependencies.

II. Morpho-syntactic disambiguation

To formalize the study of the syntactic structure of a sentence, we need two concepts: grammar - a construction specific formal language structure, and analytical techniques which will allow one to determine the correctness of a sentence in accordance with the rules of grammar.

Grammar consists of a set of rules (productions) which provides a formal description of possible syntactic structures in language that describes it.

For morpho-syntactic disambiguation method we build a context-free grammar that supports simple sentences in Romanian.

[1] This article is carried out as part of the project ref. nr. 12.819.18.09A supported by Supreme Council for Science and Technological Development from Republic of Moldova.

Context-free grammar G = (V_N, V_T, P, S), where:
V_N ={ S, NP, VP};
V_T = { art, pron, num, n, v, adj};
P={S→NP VP; NP→ art n; NP→n; NP→pron; NP→num; NP→nadj; VP→v; VP→v NP};
S (sentence) – axiom.

Context-free grammar rules (productions) are not a solution for ambiguities elimination that is encountered in the process of a sentence analysis.

Context-free grammar is extended by attaching a set of attributes to each node (i.e. each word). The associated attributes will be inherited by the unfinished nodes according to the semantic rules that accompany grammar productions.

AG are the extension of context-free grammars, where attributes are associated with grammar symbols, and semantic rules define the values of the attributes.

Thus, certain aspects of natural language such as agreements between words, subcategories, etc. can be easily shaped.

In an attribute grammar, a set of attributes is attached to each symbol. The attribute values are calculated according to the rules attached to grammar productions, called *semantic rules*. A semantic rule defines computation of an attribute in the left side of production – and then the attribute is called *synthesized* – or an attribute of a symbol from the right side of production – and then the attribute is called *inherited* [2].

So, in formal terms the attribute grammar is defined as follows:
Definition: $GA = (V_T, V_N, V_S, A, P, S)$,
where V_N *– nonterminal alphabet symbols,*
V_T *– terminal alphabet symbols,*
A – set of attributes,
V_S *– set of semantic rules,*
P – set of productions of type $A \to \alpha$, *where* $\alpha \in (V_T \cup V_N)^*$;
S – axiom.

To demonstrate the proposed method, a simple grammar was constructed with V_T*={ v (verb), n (noun), adj (adjectiv), pron (pronoun), num (numeral), adv (adverb), art (article), pp (preposition), interj (interjection), conj (conjunction)};* V_N*={NP (noun phrase), VP (verb phrase), ADJP (adjectival phrase), PP (propositional phrase), ADVP (adverb phrase)* }[3]. The set of attributes is defined as: *A={ number, gender, case, person}*. For the rule *NP*→*n adj*, for example, one of the semantic functions is:

```
if n. numer = adj. number & n.gender = adj.gender then
 NP. number = n. number, NP.gender = n.gender; NP.case = n.case.
```

Using attribute grammar more information can be formalized, which then can be used to solve problems encountered in natural language processing. One of the most difficult problems encountered in natural language processing is the ambiguity that is possibility to give two or more interpretations for a construction or its component. Often, these multiple interpretations are completely different, and in a particular context the speaker needs to choose the appropriate meaning of a word. This process is called *disambiguation* [9].

Morpho-syntactic ambiguity is characterized by a word belonging to the same or different parts of speech.

One word, however, can have multiple entries for different parts of speech, as having a different semantics, for example, the Romanian verb *a acorda* can be translated in the legal field – to make an agreement, and *a acorda* – in the music industry – to adjust the tone settings [5].

Therefore, the first step in morpho-syntactic disambiguation method is the annotation of each word from the sentence with lexical morphological attributes. To define the set of attributes we have

used the lexicon RRTLN[2] ***(Reusable Resources for the Romanian Language Technology***) developed at the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova. The lexicon consists of words and their information about the morphological categories and possible syntactic functions.

RRTLN contains a database with word-level linguistic information. Lexicon gives information about the morphological categories of speech and syntactic function.

The information associated with each word is lexical and morphological and it also contains some aspects of syntax.

RRTLN allowed the establishment of systems attributes. In addition, this computational linguistics resource has been the main source of information that formed the basis for the algorithm to achieve subject-predicate agreement [10].

For example, for the input word ***merge*** is displayed following the information from the RRLTN database represented in Figure 1.

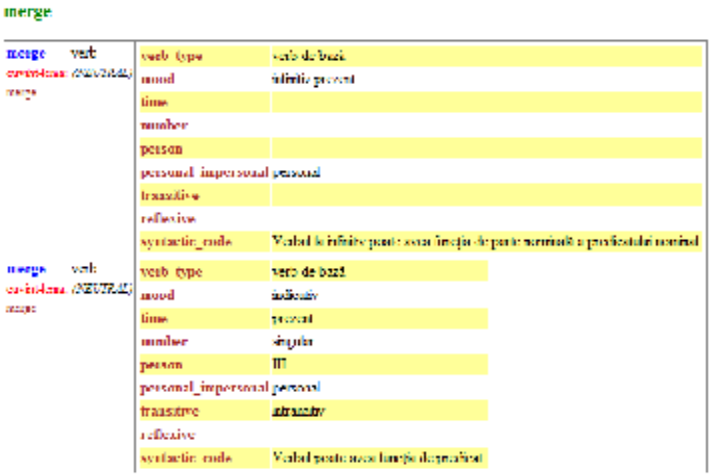

Figure 1 The information for the word ***merge***

A set of programs is developed to find the necessary information from the database RRTLN.

Semantic rules represent attribute values, calculated according to the rules attached to grammar productions.

For example, NP construction: ***un creioane*** is not correct, because the indefinite article ***un*** is singular and the plural noun is ***creioane***. It is said that the agreement does not satisfy constituents NP number of the Romanian language. There are many other agreements, such as subject-predicate agreement, pronoun gender agreement, and others. To check these phenomena of language, grammar formalism is extended by adding attributes and semantic rules.

For example, you can define the attribute number that can take two values: singular (the singular) or plural (the plural) on it, a rule might be:

NP→Art N only if ***number1*** is in agreement with ***number2***

Its meaning is: a noun phrase consists of an article followed by a noun, provided that the two words are in agreement relative to the number.

This production is equivalent to two context-free rules that will use separate terminals for codifying these singular and plural noun phrases:

NP→Art N if Art.number=NP.number then accept(Art, NP)

The agreement requires the presence of formal correspondence between two or more words,

[2] Lexiconul se conține pe site-ul http://imi201.math.md/elrr/

establishing a relationship between dependency, usually within a sentence. The phenomenon occurs in the combination the agreement of verb and noun or pronoun as subject, but in groups, in the center representing around a noun group, arrange one or more adjectives. In both cases, the agreement emphasizes the link between constituents and a set of attributes.

Table 1 Semantic Rules

Producția	**Regula semantică**
NP→ art n	*If n.determination=inarticulately then* *If art.number=n.number &* *art.gender=n.gender & art.case=n.case then* *NP.number=n.number;* *NP.gender=n.gender; NP.case=n.case;* *Accept(art n)* *End.*
NP→n adj	*If n.number=adj.number & n.gender=adj.gender* *then* *NP.number=n.number;* *NP.gender=n.gender; NP.case=n.case;* *accept(n, adj);* *End.*
VP→v NP	*If v.transitivity=intransitive then* *If v.number=NP.number then* *VP.number=v.number;* *VP.mode=v.mode; VP.time=v.time;* *accept(v, NP);* *End.*

Syntactic analysis techniques are used to automate the analysis of sentences. Syntactic analysis techniques used in natural language processing differ from those used for instruction parsing of programming languages. This difference comes from the fact that programming languages have a deterministically pronounced character, while in natural language the ambiguity is an obvious feature.

Syntax description of simple sentences of the Romanian language using attribute grammars allows the use of formal methods in expanding the parser.

Syntactic analysis, which cannot be a stand-alone application in natural language analysis, is used in the combination with a method of semantic analysis represented by semantic rules. These rules are created to solve some problems related to the agreement between different parts of speech. The analysis process is automated using ascending left to right (LR) analysis techniques.

There are several types of LR parsers differentiated by the structure of parsing tables and used grammars. We will use the LALR (1) parser which consists of: input tape, stack, output tape and parsing tables. Parsing tables constructing is an important step that determines the efficiency of parser, because these tables take an important part of the analysis management [1].

Constructing tables of the analysis is the step that determines the efficiency of LR type analyzer, because it takes an important part of the analysis.

In order to evaluate attributes during syntactic analysis, LALR(1) parser is modified by adding a parallel stack in which attribute values are stored for each terminal and nonterminal symbols. Integration of attributes evaluation with syntactic analysis has led to the use of semantic elements, thus making morpho-syntactic disambiguation.

The using the attribute grammars for Romanian to solve morpho-syntactic ambiguity demonstrates that all attributes are synthesized. This simplifies the analysis by applying the semantic rule corresponding to each of its production.

A syntactic analysis algorithm can be built simply as a procedure trying different ways to combine grammatical forms, in order to achieve a combination on the basis of which to build a derivation tree structure corresponding to input sequences. In the first phase of the construction one will not be interested in the tree, but only answer if the input sequence can be generated or not grammar and semantic rules associated with production.

III. Conclusion

In this paper we presented a morpho-syntactic disambiguation method using attribute grammars. To define the attributes set the computational linguistic resources developed at the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova was used. The process of semantic rules evaluation was integrated with a syntactic ascending LR parser.

The algorithm presented in this paper has been implemented and tested on a small set of preliminary examples. However, the formalism presented in this article proved to be useful in the process of morpho-syntactic disambiguation.

IV. References

1. Knuth D.E. Semantics of context-free languages. *Mathematical Systems Theory.* pp. 127-145.
2. Grigoraş G. *Proiectarea compilatoarelor.* Editura Universității A. I. Cuza din Iaşi, 2007, pp123.
3. Hristea F. *Introducere în procesarea limbajului natural cu aplicații în prolog.* Editura Universității din Bucureşti, 2000, pp 309.
4. Athanasiu I. *Limbaje formale şi translatoare.* Bucuraşti 1991, pp 167.
5. Irimia D. *Gramatica limbii române.* Polirom 2004, pp 543.
6. Ionescu E. Gramaticile generative nontransformationale. *Limba Română în Societatea Informațională - Societatea Cunoaşterii*, pp.41-49.
7. Tufiş D. Dezambiguizarea automată a cuvintelor din corpusuri paralele folosind echivalenții de traducere. *Limba Română în Societatea Informațională - Societatea Cunoaşterii*, pp. 235-267.
8. Barbu A. Teoria HPSG. Studiu de caz: acordul incrucisat. *Limba Română în Societatea Informațională - Societatea Cunoaşterii*, pp.89-109.
9. Radu I., *Metode de dezambiguizare semantică automată. Aplicații pentru limba engleză şi română*, Teză de doctor în informatică. Bucureşti, mai 2007. 158p.
10. M. Petic, V. Gîsca, O. Palade. Multilingual mechanisms in computational derivational morphology. *Proceedings of Workshop on "Language Resources and Tools with Industrial Applications"* Cluj-Napoca, Editura UAIC Iasi, pp. 29-39.