МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2005
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2005
Proceedings of the Thirty Fourth Spring Conference of
the Union of Bulgarian Mathematicians
Borovets, April 6–9, 2005

AUTOMATIC PARSING: A PROBABILISTIC APPROACH FOR BULGARIAN

Hristo D. Krushkov, Atanas G. Chanev

Natural Language Processing (NLP) is one of the most challenging fields in AI. Machine translators and other practical tools have been implemented recently for wide spread languages like English, German etc. The natural languages are quite different from each other and that makes the usage of models for English in systems, processing less spoken languages a difficult task. Although there are few models for Bulgarian syntax, the analysis of that level of language has not given the expected results. In this paper a model for parsing sentences in Bulgarian has been described. In addition, a system has been implemented and tested using stochastic grammars for Bulgarian, extracted from a minicorpus of 5331 tokens.

1. Introduction. There are not many attempts a formal grammar for Bulgarian to be obtained or generated. The grammars, described in most of the textbooks and literature, such as in [7] are not formalized. These grammars cannot be used in models for automatic processing of Bulgarian language, mainly because of the ambiguities they generate. Bulgarian is an inflectional language and the word order is not fixed as in analytic languages like English. The free word order is an additional cause for ambiguities, which are difficult to solve [9].

Context-free grammars (CFG) are representing the main features of the Bulgarian language. Nevertheless they cannot describe more specific phenomena, such as the probabilistic aspect of language [2]. In this document we describe a top down parser for Bulgarian, which we have implemented, a stochastic CFG, that we have extracted from a minicorpus, and the difficulties we came upon while trying to fully parse sentences in Bulgarian.

2. Algorithm for parsing sentences. For the implementation of our system for parsing the top down approach had been chosen. As the stochastic nature of the language should be considered, we chose SCFGs (Stochastic Context-Free Grammars) for our model. This type of combined approach (top down, both rule based and statistically based) is chosen in order whole sentences to be parsed. The algorithm tries to satisfy the full parsing needs instead of giving partial parses, which does the chunker of CLaRK System [3]. Morphological information [8] about the words of a sentence is a prerequisite for this method of parsing.

The implemented algorithm uses a table with rules for processing of a sentence. The rules, which should be appended to the table, are called records. The analysis begins

top down. Initially, all the rules from the grammar, which could be added before the first word from the sentence has been read, are appended to the table. The first word is read afterwards. The algorithm tries to read the first nonterminals from the right-hand sides of the rules in the table. Afterwards rules from the grammar, which could be added before the second word from the sentence has been read, are looked for again. All these steps repeat until all the words from the sentence have been read and the algorithm cannot read more nonterminals from the right-hand sides of the rules from the table. A detailed description of the algorithm could be found in [1].

The Earley algorithm is not able to solve ambiguities. An extension of the Earley algorithm, which uses a stochastic grammar has been presented by Stolcke [6]. Each rule from the grammar has a probability and it is needed in order the probability of the record, which is going to be appended to be calculated. The advantage of this approach is that if the probability of a record is under a pre-defined limit, it is not added in the table and this way less probable solutions are blocked.

Feature structures and an unification mechanism are used in order sentences, which are not admissible, because of disagreement of different parts of speech, to be avoided [2]. A simplified unification mechanism is used. Simov describes a full HPSG (Head-driven Phrase Structure Grammar) annotation scheme in [4] where full unification is the core of the automatic chunking. The approach in this paper is constituent structures oriented and the limited unification is used only as an agreement constraint.

3. Corpus and grammar. In order the system for parsing to be tested, a small corpus of newspaper articles (5331 tokens) was annotated.

After the annotation, a stochastic grammar in the newspaper articles register for Bulgarian was obtained. The probability of each rule was computed as the number of the occurrences of the rule in the corpus was divided to the number of the occurrences of all the rules from the corpus, which had the same nonterminal at their left hand side. The grammar consists of 215 rules.

In difference with other corpora, such as the corpus, described in [3] and [5], all the pronouns are annotated as NP (Noun Phrase) structures. Thus, it is clear, that there are referents from the real world for them and a pronominal anaphora resolution algorithm could be applied afterwards.

In addition another grammar was prepared. It differed from the first only in the method for obtaining the probabilities of the rules. The probabilities were calculated using the following formulae:

$$X\,(\text{rule}) = \frac{\text{count}\,(\text{rule})^2}{\text{count}\,(\text{same}\,(\text{LHS})) * \text{count}\,(\text{allTheRules})};$$

$$P\,(\text{rule}) = \frac{X\,(\text{rule})}{\sum\limits_{i=\text{same(LHS)}} X\,(\text{rule}_\text{i})}.$$

4. Results. The algorithm is able to find complete sentence trees in XML (eXtensible Markup Language) format [10]. Thus there are not any attributes with grammatical information, such as in [3]. The algorithm represents the syntax of the language in plain constituent structures instead of HPSG graphs. A full parse tree is always guaranteed.

After a single article was parsed, precision of 72.73% and recall of 61.11% were ob-

tained [10]. After making tests on 100 sentences from articles from different authors the precision fell to 42.19% and the recall – to 64%. Many sentences had wrong parses, so another grammar, which had different rule probabilities was prepared. The reason we chose to change the model of calculating the probabilities was that comparatively rare rules could have bigger probabilities than frequently used ones only because the nonterminal in their LHS (Left-Hand Side) generates a smaller number of possible RHSs (Right-Hand Sides.) After tests were made on the same 100 sentences the precision increased up to 42.42% and the recall – up to 66%.

Osenova and Simov describe the PP (Prepositional Phrase) attachment problem for their shallow parser in [3]. In the algorithm, described in this paper a different approach has been implemented. All the PPs, which modify the verb, are considered to be adverbial phrases. For the purpose, the separate constituent AdvP has been presented.

One of the reasons for the comparatively poor results could be found in the constant, which limits the addition of rules with lower probabilities. In sentences, which contain more words, the probabilities of the rules get smaller after every scanned word, so that at the end of such sentences all the rules are blocked due to low probabilities. A possible solution is the mentioned constant to be calculated as a function of the number of words in the sentence. The disadvantage of such an approach is that the smaller constants won't block less probable paths of parsing at the beginning stage of the parsing. A more reasonable approach could be the calculation of the constant – now a variable after each scanning step. Thus it could be bigger enough at the beginning of the parse and smaller enough at its end.

Another reason could be the unification mechanism, which is not precise enough. Sometimes it blocks correct rules and sometimes it allows wrong rules to be added to the table. The preparation of an effective unification processor for Bulgarian could be very labour intensive.

There are some sentences, which have their wrong parses due to the stochastic model and the top down approach. The more rules the algorithm adds to the table, the smaller probabilities are calculated for the sentence. Thus the shallow trees have bigger probabilities than the deeper ones.

Wrong parses are obtained if the sentence contains phrases or idioms. A possible solution is the preparation of dictionaries, which contain typical phrases or idioms, such that the algorithm could scan them as a single word.

In case of a more difficult to determine sentence, the program finds the correct parse. The sentence tree, which has a smaller probability but it is still close to the probability of the correct parse is exactly the same tree, which could make a human annotator be in doubt. That phenomenon is observed in many of the cases of successfully found correct answer.

The obtained results lead to the following conclusion: A deeper understanding of language should be integrated in the algorithm through a better unification processor and various dictionaries. Although the results of the tests are not satisfactory enough, the obtaining of right parses for ambiguous sentences makes the usage of the top down approach, combined with SCFGs promising.

REFERENCES

[1] J. Earley. An Efficient Context-free Parsing Algorithm. *Communications of the ACM*, **6**, 1970, No 8, 451–455.
[2] D. Jurafsky, J. H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing. Computational Linguistics and Speech Recognition, Prentice Hall, New Jersey, 2001.
[3] P. Osenova, K. Simov. Between Chunk Ideology and Full Parsing Needs. In: Proceedings of the Shallow Processing of Large Corpora (SProLaC 2003) Workshop, Lancaster, UK, 78–87.
[4] K. Simov. HPSG-Based Annotation Scheme for Corpora Development and Parsing Evaluation. In: Proceedings of the RANLP 2003 Conference, Borovets, Bulgaria, 2003, 432–439.
[5] K. Simov, P. Osenova, A. Simov, K. Ivanova, I. Grigorov, H. Ganev. Creation of a Tagged Corpus for Less-Processed Languages with CLaRK System. In: Proceedings of SALTMIL Workshop at LREC 2004, First Steps in Language Documentation for Minority Languages, Lisbon, Portugal, 2004, 80–83.
[6] A. Stolcke. An Efficient Probabilistic Context-free Parsing Algorithm That Computes Prefix Probabilities. Technical Report TR-93-065, International Computer Science Institute, Berkeley, CA, 1993, Revised 1994.
[7] Т. Бояджиев, И. Куцаров, Й. Пенчев. Съвременен български език. Петър Берон, София, 1999.
[8] Хр. Крушков. Моделиране и изграждане на машинни речници и морфологични процесори. Дисертация за присъждане на образователна и научна степен "доктор", ПУ "Паисий Хилендарски", Пловдив, 1997.
[9] Хр. Танев. Автоматичен анализ на текстове и решаване на многозначности в българския език. Дисертация за присъждане на образователна и научна степен "доктор", ПУ "Паисий Хилендарски", Пловдив, 2001.
[10] А. Чанев. Автоматичен анализ на текстове на български език. Дипломна работа, ПУ "Паисий Хилендарски", Пловдив, 2004.

Hristo Dimitrov Krushkov
Paisii Hilendarski Plovdiv Unuversity
Fakulty of mathematics and informatics
24, Tzar Asen Str.
4000 Plovdiv, Bulgaria
e-mail: hdk@pu.acad.bg

Atanas Georgiev Chanev
31, Ladjene Str.
4000 Plovdiv, Bulgaria
e-mail: artanisz@mail.bg

АВТОМАТИЧЕН СИНТАКТИЧЕН АНАЛИЗ: ЕДИН ВЕРОЯТНОСТЕН ПОДХОД ЗА БЪЛГАРСКИЯ ЕЗИК

Христо Д. Крушков, Атанас Г. Чанев

Автоматичната обработка на естествения език е една от най-предизвикателните области на Изкуствения интелект. Машинният превод и други практически средства за такава обработка се прилагат в много езици като английски, немски и др. Използваните за тази цел синтактични модели не могат директно да се прилагат за българския език. Съществуващите наши такива не са подходящи за компютърни приложения. В настоящата статия е предложен един вероятностен подход за автоматичен синтактичен анализ на български текст. Реализирана и тествана е програмна система, която използва стохастична граматика, извлечена от малък корпус, състоящ се от 5331 езикови единици (tokens).