Constituency Parsing of Bulgarian: Word- vs. Class-based Parsing

Masood Ghayoomi[†] **Kiril Simov**[‡] **Petya Osenova**[‡]

[†]Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[‡]Linguistic Modelling Department, IICT-BAS, Sofia, Bulgaria

masood.ghayoomi@fu-berlin.de kivs@bultreebank.org petya@bultreebank.org

Abstract

In this paper, we report the obtained results of two constituency parsers trained with BulTreeBank, an HPSG-based treebank for Bulgarian. To reduce the data sparsity problem, we propose using the Brown word clustering to do an off-line clustering and map the words in the treebank to create a class-based treebank. The observations show that when the classes outnumber the POS tags, the results are better. Since this approach adds on another dimension of abstraction (in comparison to the lemma), its coarse-grained representation can be used further for training statistical parsers.

Keywords: Constituency Parsing, Word Clustering, the Bulgarian Language, Treebanking

1. Introduction

One of the primary steps in natural language understanding is syntactic analyses of sentences. This task can be done in two steps: (a) assigning the part-of-speech (POS) tag to each word for defining the syntactic behavior of the word, (b) bracketing the words that construct a constituent for determining the role of the words that interact with each other. The former problem is greatly succeeded almost for all languages, but it is not the case for the latter step.

Rule-based and statistical methods can be used for bracketing sentences automatically. Using either of these approaches has its own advantages and disadvantages. Rule-based parsers are language dependent and mostly the defined grammar rules in one language are not reusable for another language, while statistical parsers are language independent and the system can be adapted for a new language. The advantage of rule-based approaches is that no prior annotated data is required, but in statistical methods a huge amount of annotated data (such as, treebanks) is required for a model to be built.

In this paper, we describe the employment of statistical constituency parsers, such as the Stanford and Berkeley parsers, for Bulgarian. Furthermore, to reduce data sparsity and create more coarse-grained model, we propose a class-based parsing.

The structure of this paper is as follows: the previous studies on parsing Bulgarian and related works are described in Section 2. The data source to train the parsers is introduced in Section 3. The clustering approach used for the class-based parsing is explained in Section 4. The setup of experiments and the obtained results are discussed in Section 5. Finally, the paper content is summarized in Section 6.

2. Related Works

2..1 Parsing Bulgarian

In this section, we introduce the related works in several directions. First, we present results from parsing Bulgarian. A lot of work has been done already on dependency parsing: (Marinov and Nivre, 2005), (Chanev et al., 2006), (Chanev et al., 2007), papers from CoNLL-X 2006 Shared Task (Buchholz and Marsi, 2006), and others. The best result, mentioned in the literature, is 93.5% unlabeled attachment accuracy (Martins et al., 2011). The best model for Bulgarian which is available for us has obtained the following results: 89.6% for labeled attachment accuracy and 92.5% for unlabeled attachment accuracy. Not many experiments, however, have been done on Bulgarian constituent parsing. We are aware of only one work done by Chanev et al. (2007). The reported results of the F-measure for unlabeled evaluation and labeled evaluation are 80.4% and 80.2%, respectively.

There are two reasons that motivated us for the current study on the constituency parsing of Bulgarian: (1) our setup is richer than the one reported in previous studies, because we keep the full range of POS tags and also encode the information for the discontinuous constituents which is missing in other treebank conversions; (2) our goal in this paper is not to train the best parser for Bulgarian, but to study the impact of the semantic information (provided as cluster names) on constituent parsing.

2..2 Parsing Persian

PerTreeBank is an HPSG-based treebank developed for Persian (Ghayoomi, 2012a) which contains only 1,028 trees. The annotation scheme used in PerTreeBank is relatively similar to BulTreeBank, the Bulgarian treebank (Osenova and Simov, 2007). Due to the small size of this treebank, Ghayoomi (2012b) proposed a class-based parsing model to reduce the data sparsity problem. Ghayoomi used the Brown word clustering algorithm (Brown et al., 1992) for the experiments. Since short vowels are not written in Persian, Ghayoomi proposed an extension to the normal word clustering algorithm to deal with homographs. In this extension, the main POS tags of the words are attached to the word forms to make the homographs distinct. As an example, in this new data format, the homo-

graph 'نبرد' in Persian is represented as 'نبرد-N' or 'نبرد-V' when the word is used as either a noun (/nabard/ 'fight') or a verb (/nabarad/ 'not taking'; /nabord/ 'did not take'; /naborad/ 'does not cut'), respectively. For comparison, in Bulgarian this problem also exists, thus the same solution has been applied. For example, the homograph 'син' is represented as either a noun (/sin/ 'son') or an adjective (/sin/ 'blue').

There is still one drawback in this extension, and that is related to the homographs which have similar POS tags, such as the word 'نبرد' when it functions as a verb in Persian. This word form is created from two different lemmas. For these cases, attaching the POS tags will not help much to make them distinct and to put them in different clusters, but the lemma information can play a role. The lemma of the verbs /nabarad/ and /nabord/ is 'بردن' /bordan/ 'to take'; and the lemma of the verb /naborad/ is 'بریدن' /boridan/ 'to cut'.

In this paper, we try to tackle this problem which is described in Section 4. We apply this model on clustering the Bulgarian words and use the clusters for the parsing application.

3. Bulgarian Treebank

The annotation schema behind the Bulgarian treebank, called BulTreeBank (BTB), (Osenova and Simov, 2007) generally follows the HPSG linguistic model by Pollard and Sag (1994). It incorporates the universal principles, such as Head Feature Principle, Valence principle, etc. In addition, it follows the hierarchical approach when attaching dependents to their heads. First, the complements are attached, then the subject being an external argument, and finally the adjuncts. It should be noted that the complements are attached together, by one operation only. Additionally, in BTB the constituent structure is separated from the word order. It means that the topic-focus layer is not distinguished. In such a paradigm, crossing branches are allowed, and three types of discontinuity are envisaged (scrambling, topicalization, and mixed). The implementation is based on XML, where the XML tree structure is exploited to represent the constituent structure as much as possible with encoding of crossing branches via ID and IDREF attributes. The visualization takes the form of the XML tree and represents it as close as possible to the canonical syntactic trees. The dependency relations are also encoded into the syntactic labels. For example, VPC means verbal phrase with a complement.

Apart from the phrase level, another level has been introduced, the functional level. This level handles the various types of clauses (CLR, CLDA, CLQ, and CL), coordination, co-referenced pro-dropness, etc. BTB takes into account the types of named entities (person, organization, location and other), various co-references within the sentence as well as the ellipses. The layers in BTB are modeled separately. Morphological analyses come first. The ambiguous ones have been disambiguated manually. Then, chunks have been analyzed, and finally full analyses with handling the specific attachments, discontinuities and cases of ellipsis. Non-local dependences are handled by the discontinuity markers only.

BTB introduces phrase structures and dependency relations, but lacks feature structures as well as a separate semantic layer of representation. The semantics can be derived as follows: (1) the predicate structure through the dependency labels (arity) and co-references (control, pro-dropness); (2) the relations through the functional labels (nominalizations, subordinate clauses among others) and co-references (possession, control, etc.). The scope of quantification is present only in the selected interpretation by the annotator. Additionally, the analysis of names shows the semantically correct analysis with respect to subject and complement selection.

The tree analysis of Example 1 from BTB is presented in Figure 1. The determiner /Nikoya/ 'nobody' is viewed as an adjunct within the NPA. The phrase is also a subject to an intransitive verb.

(1) Nikoya kotka ne laeshe.
Nobody cat no was-barking
'No cat was barking.'

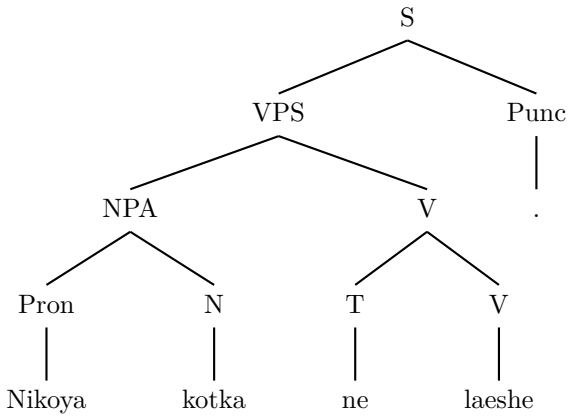

Figure 1: Sample tree of Example 1 from BTB

4. Class-based Parsing

Statistical parsers require a huge amount of annotated data for creating an accurate grammar model. Due to the infinite nature of languages, the model suffers from data sparsity. In order to reduce the amount of data sparsity, we propose to use a more coarse-grained level of the lexicon rather than the actual words used in the original data. To this end, first the words in a corpus are clustered according to a clustering criterion; then the words in the treebank are mapped to their relevant clusters. The newly developed class-based treebank will be used for training a parser. The consequence of this parsing model is that it provides another dimension of generalization over the treebank data, in addition to lemmas. We call this method 'class-based parsing'.

Considering our ultimate goal for using the clustered data (thereafter we call it Model A), we use the Brown word clustering method (Brown et al., 1992) which is a hard clustering, i.e. in this clustering each word is put in only one cluster. This clustering approach uses Mutual Information (MI) as the clustering criterion.

$$MI(C_w, C_{w'}) = \log \frac{P(C_w, C_{w'})}{P(C_w) * P(C_{w'})} \quad (1)$$

The shortcoming of this clustering approach is treating homographs equally. Thus, it might be better to use a soft clustering approach. But Dhillon et al. (2002) have shown experimentally that the overall performance of the hard clustering is higher than the performance of the soft clustering.

To handle the shortcoming of the Brown word clustering and the extended model proposed by Ghayoomi (2012b) to treat homographs distinctly, we first add the relevant information of lemmas and the POS tags of the words to the word forms, and then start clustering (thereafter we call it Model B). In this data format, we add morphological (lemma) and syntactic (POS tag) information to the word forms. Since the equation (1) computes the degree of semantic information that two words share, we can conclude that in our proposed clustering model all morphology, syntactic, and semantic information is taken into consideration to cluster the words more accurately.

5. Evaluation

5..1 Experimental Setup

For our experiments, we train two constituency parsers with our treebank, the Stanford parser (Klein and Manning, 2003) and the Berkeley parser (Petrov et al., 2006). The difference between these two parsers is that the Stanford parser is a lexicalized parser, whereas the Berkeley parser is an unlexicalized parser. To adapt the Stanford parser for Bulgarian, a head finder table is provided for the parser.

To prepare the data and train the parsers, we converted automatically the original XML format of BTB into the Penn treebank style, and we used the gold POS tags to reduce the interference of POS tagging on parsing. During this conversion process, we keep the discontinuous information via modification of the constituent labels. Figure 2 represents the Penn style format of the tree analysis of Example 1.

```
(S
  (VPS
    (NPA
      (Pne-os-f  Nikoya)
      (Ncfsi  kotka))
    (V
      (Tn  ne)
      (Vpitf-m3s  laeshe)))
  (.  .))
```

Figure 2: Penn style tree analysis of Example 1

To create the class-based treebank, we needed to map our data to a clustered data. To create the clustered data, we used the SRILM toolkit (Stolcke, 2002) which has the implementation of the Brown algorithm. The Brown word clustering algorithm requires a predefined number of clusters. For our experiments, we clustered the words into 100, 500, 1,000, and 1,500 classes. The data that we used for clustering is from the Bulgarian National Reference Corpus[1] which is (automatically) annotated at the morphological level, and also lemmatized.

To evaluate the performance of our parsing models, we used the standard PARSEVAL metrics as well as the Leaf-Ancestor criterion proposed by Sampson (2000). The latter metric computes the cost for converting a false label into the correct one. This metric compares the similarity of the path to link each leaf (word) of a sentence to the root node in both the gold standard and the candidate trees, and then computes the overall average of the correct paths.

10-fold cross-validation is used for evaluating the performance of both the word- and class-based parsings models. A set of 12,855 trees is used as the training data, and 1,428 sentences as the test data.

5..2 Results

For the experiments, we first evaluate the word-based parsing model as the baseline. Table 1 summarizes the obtained results of the two parsers. As it can be seen, the Berkeley parser outperforms the Stanford parser according to all evaluation metrics.

Table 1: Performance of constituency parsers trained with the word-based model

Parser	F-measure	Precision	Recall	Exact Match	Leaf Ancestor
Stanford	64.65	65.00	64.30	13.714	86.09
Berkeley	71.03	70.29	71.78	14.871	88.25

Tables 2 and 3 report the obtained results of the class-based model, for Models A and B. Comparing the overall performance of class-based parsing with the word-based model, the class-based models of the Stanford parser outperformed its baseline, while the class-based models of the Berkeley parser has a slightly better or worse performance than the baseline. Moreover, as it can be seen in the tables, the performance between different numbers of clusters, which is to some extent uniformed, is not statistically significant.

Comparing the performance of the parsers in Models A and B, Model B has a slightly better performance in both parsers. Cluster 1000 of Model B performs the best for both the Stanford and Berkeley parsers. The difference between this model and the word-based parsing baseline is statistically significant according the two tailed *t*-test ($p < 0.05$).

Comparing the exact match rate in the word-based parsing and Models A and B, we can observe significant improvement. This shows how effective the words are on the performance of the parsers in such a way that a more coarse-grained lexicon can improve the performance. Moreover, the class-based models relatively reduced the cost for converting a false label into the correct one. This reduction is more surprising for the Stanford parser which is lexicalized.

[1] http://www.webclark.org/Clark.html

Table 2: Performance of class-based models of the Stanford parser

Parser	Class	F-measure	Precision	Recall	Exact Match	Leaf Ancestor
Stanford (Model A)	100	67.69	66.86	68.54	17.106	87.73
	500	66.90	66.45	67.37	16.302	87.53
	1000	67.83	67.00	68.68	16.876	87.77
	1500	67.86	67.08	68.65	16.848	87.82
Stanford (Model B)	100	67.74	66.95	68.55	16.925	87.75
	500	67.88	66.07	68.70	16.989	87.71
	1000	**67.94**	67.11	68.78	16.981	87.80
	1500	67.90	67.14	68.68	16.965	87.74

Table 3: Performance of class-based models of the Berkeley parser

Parser	Class	F-measure	Precision	Recall	Exact Match	Leaf Ancestor
Berkeley (Model A)	100	70.84	71.16	70.51	17.004	88.73
	500	71.16	71.40	70.92	17.778	88.75
	1000	71.10	71.26	70.95	17.643	88.75
	1500	71.42	71.65	71.19	17.248	89.00
Berkeley (Model B)	100	70.33	70.74	69.94	16.652	88.86
	500	70.78	71.08	70.47	16.702	88.70
	1000	**71.62**	71.82	71.43	17.652	89.15
	1500	71.05	71.28	70.82	17.484	88.75

6. Summary

In this paper, we described the adaptation of two constituency parsers for Bulgarian in order to balance the dependency-based mainstream for this language, and to enhance future work on performance improvement. Furthermore, we proposed using word clustering for the parsing task. To this end, the Brown word clustering was used. The shortcoming of this clustering algorithm is making no distinctions between homographs. However, we proposed an extension to this model to use richer information for clustering. Based on the experiments, we can conclude that: first, we succeeded to train and compare two constituency parsers for Bulgarian. Second, we obtained a relatively better performance using a class-based model, which shows a positive impact of semantics on the parsing results.

7. Acknowledgements

This research has received partial funding from the EC's FP7 (FP7/2007-2013) under grant agreement number 610516: "QTLeap: Quality Translation by Deep Language Engineering Approaches".

References

Brown, Peter F., deSouza, Peter V., Mercer, Robert L., Pietra, Vincent J. Della, and Lai, Jenifer C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18:467--479.

Buchholz, Sabine and Marsi, Erwin. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149--164, New York City, June. Association for Computational Linguistics.

Chanev, Atanas, Simov, Kiril, Osenova, Petya, and Marinov, Svetoslav. (2006). Dependency conversion and parsing of the BulTreeBank. In *Proceedings of the LREC workshop Merging and Layering Linguistic Information*, pages 16--23.

Chanev, Atanas, Simov, Kiril, Osenova, Petya, and Marinov, Svetoslav. (2007). The BulTreeBank: Parsing and conversion. In *Proceedings of the Recent Advances in Natural Language Processing Conference*, pages 114--120.

Dhillon, Inderjit S., Mallela, Subramanyam, and Kumar, Raul. (2002). Enhanced word clustering for hierarchical text classification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 191--200.

Ghayoomi, Masood. (2012a). Bootstrapping the development of an HPSG-based treebank for Persian. *Linguistic Issues in Language Technology*, 7(1).

Ghayoomi, Masood. (2012b). Word clustering for Persian statistical parsing. In Isahara, Hitoshi and Kanzaki, Kyoko, editors, *Advances in Natural Language Processing*, volume 7614 of *Lecture Notes in Computer Science: JapTAL '12: Proceedings of the 8th International Conference on Advances in Natural Language Processing*, pages 126--137. Springer Berlin Heidelberg.

Klein, Dan and Manning, Christopher D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of*

the Association for Computational Linguistics, pages 423--430.

Marinov, Svetoslav and Nivre, Joakim. (2005). A data-driven parser for bulgarian. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 89--100.

Martins, Andre, Smith, Noah, Figueiredo, Mario, and Aguiar, Pedro. (2011). Dual decomposition with many overlapping components. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 238--249, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Osenova, Petya and Simov, Kiril. (2007). *Formal Grammar of Bulgarian (in Bulgarian).* IPP-BAS, Sofia, Bulgaria.

Petrov, Slav, Barrett, Leon, Thibaux, Romain, and Klein, Dan. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the Association for Computational Linguistics*, pages 433--440, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pollard, Carl J. and Sag, Ivan A. (1994). *Head-driven Phrase Structure Grammar.* University of Chicago Press.

Sampson, Geoffrey. (2000). A proposal for improving the measurement of parse accuracy. *International Journal of Corpus Linguistics*, 5:53--68.

Stolcke, Andreas. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing.*