Linguistic Processing Pipeline for Bulgarian

Aleksandar Savkov[i,ii], Laska Laskova[i], Stanislava Kancheva[i], Petya Osenova[i], Kiril Simov[i]

[i]Language Modelling Department, IICT-BAS, [ii]Department of Informatics, University of Sussex
[i]Acad. G.Bonchev St. 25A, 1113 Sofia, Bulgaria, [ii]Chichester 1 Room 214, Falmer, Brighton BN1 9QJ, UK
a.savkov@sussex.ac.uk, {laska|stanislava|petya|kivs}@bultreebank.org

Abstract

This paper presents a linguistic processing pipeline for Bulgarian including morphological analysis, lemmatization and syntactic analysis of Bulgarian texts. The morphological analysis is performed by three modules — two statistical-based and one rule-based. The combination of these modules achieves the best result for morphological tagging of Bulgarian over a rich tagset (680 tags). The lemmatization is based on rules, generated from a large morphological lexicon of Bulgarian. The syntactic analysis is implemented via MaltParser. The two statistical morphological taggers and MaltParser are trained on datasets constructed within BulTreeBank project. The processing pipeline includes also a sentence splitter and a tokenizer. All tools in the pipeline are packed in modules that can also perform separately. The whole pipeline is designed to be able to serve as a back-end of a web service oriented interface, but it also supports the user tasks with a command-line interface. The processing pipeline is compatible with the Text Corpus Format, which allows it to delegate the management of the components to the WebLicht platform.

Keywords: HLT Services, Tagging, Lemmatization, Parsing

1. Introduction

In the recent years, the Natural Language Processing (NLP) community focuses on two perspectives: (1) integration of existing tools and resources for various languages, and (2) making them publicly available on the web. Even though many such tools and resources already exist, they are often not accessible or hard to integrate into usable application architectures. For that reason, the pan-European CLARIN initiative put as its main goal the *communication* among all differing resources as well as their *applicability* in the area of humanities. This paper describes a processing pipeline for Bulgarian including language technology services that are designed to be widely accessible and reusable through the Internet. The modules are run as separate units and as a whole pipeline.

The Linguistic Processing Pipeline for Bulgarian (BTB-LPP[1]) was developed as part of the *EuroMatrixPlus* project whose goal is to create a Bulgarian-English machine translation system. Our project set out to realize this goal by chaining a number of linguistic processing tools and resources together in order to generate semantic analysis of Bulgarian that enables more accurate translation into English. The processing pipeline currently makes use of several processing modules: (1) a Finite State Transducers (FST) based tokenizer and sentence splitter module, (2) several modules implemented in CLaRK system (Simov et al., 2001) making use of a large morphological dictionary, manually crafted disambiguation rules and lemmatization rules, (3) a guided learning POS-tagging system — GTagger (Georgiev et al., 2012), (4) a POS-tagging statistical model for the SVMTool tagger (Gimenez and Marquez, 2004), (5) and a statistical parsing model for the MaltParser (Nivre and Hall, 2005). These modules can be accessed both — through a command line and web service interfaces. The web service interface allows each of them to be easily integrable and interchangeable with alternative analysis modules on their respective levels. Each of the presented modules will be made available on the web as a free service.

The rest of the paper is organized as follows: Section 2. gives an insight of the ideas and goals of BTB-LPP architecture; Section 3. presents the datasets and the linguistic resources that were used in the different processing modules; Section 4. describes the modules in their usual order of usage; The last section concludes the paper and gives some directions for future work.

2. BTB-LPP Architecture

In the context of eScience, researchers want not only to share their resources and technologies, but also to minimize the work needed to reuse them. One of the current major problems is that many technologies are incompatible with each other. Although some have chosen to implement general data-encoding standards like TEI (Burnard and Bauman, 2007), many linguistic tools and resources develop their own operational annotation formats and very few of them choose to implement common interfaces, which impedes the interoperability of language technologies. To ensure that our processing modules can be shared and reused properly in the context of the CLARIN linguistic infrastructure project (Váradi et al., 2008), we decided to adopt some of the underlying ideas of the project D-SPIN – the German-based linguistic resources infrastructure project from the preparational phase of CLARIN. Its main product, the platform WebLicht, is a web-based service environment that allows the users to integrate and exploit various language resources and tools directly through web (Hinrichs et al., 2010). It allows them to upload resources and share tools in one place with common operation and annotation formats, thus improving their collaboration.

[1] The pipeline is developed on the basis of the language resources, created within BulTreeBank project. The prefix BTB stands for BulTreeBank.

Our processing pipeline is built in accordance with the WebLicht standards implementing the Text Corpus Format (Hinrichs et al., 2010) and thus it should be compatible with the infrastructure that is being developed under the CLARIN project. It is our intention to make BTB-LPP available both — through WebLicht platform and through an independent web interface.

The current architecture of BTB-LPP provides modularity in the spirit of CLARIN web services architecture. This modularity provides a hybrid architecture that combines rule-based and statistical components. Such an approach proves to be the most efficient way for achieving high quality results. Our plan is to identify the weak spots in the results of the statistically trained tools and to correct the errors via rule-based methods.

3. Data Resources

This section describes the linguistic resources used in the pipeline. Some of them have been used for training and testing of the machine learning tools. Others are used in the rule-based components.

3.1. Datasets

In the process of training and testing of the different machine learning tools in the pipeline two interrelated datasets were used: (1) Morphologically annotated dataset of BulTreeBank[2] (Simov et al., 2004a); and (2) Dependency part of BulTreeBank[3]. The second dataset is constructed as a conversion of a part from the original HPSG-based treebank.

The morphologically annotated corpus contains 321 542 tokens in 20 556 sentences. Each token is annotated with its possible morphosyntactic tags. The set of possible tags is taken from the morphological lexicon, described in the next subsection. One of these possible tags is manually selected as correct for the token in the specific context. The number of tags is 680. They are from the BulTreeBank tagset (Simov et al., 2004b). The dataset has been divided into three parts: training part (80% of the sentences), validation part (10%) and test part (10%).

The dependency dataset contains a little more than 196000 tokens in 13 000 sentences. The data is encoded according to the standards in CoNLL Shared Task 2006[4]. The division of the dataset corresponds to the division of the morphologically annotated corpus. In this way, the bias of the data on the machine learning components is avoided.

We observed some sparseness in the training set, namely it lacks 128 word types, which are only found in the evaluation and testing sets.

3.2. Morphological Lexicon

The most important idea of our processing strategy is the use of linguistic knowledge to improve the results achieved by the language models and statistical algorithms. The cornerstone of this knowledge is the extended version of the morphological dictionary published in Popov et al. (1998) and Popov et al. (2003). It comprises 110 000 lemmas that are linked to 1.5 million word forms each of them with a corresponding morphosyntactic tag. The lexicon can be classified as exhaustive with regard to the lemmas it contains. Additionally, we incorporated a set of gazetteers of approximately 40 000 personal names (Osenova and Simov, 2002).

4. Processing Modules

Each of the processing modules performs one or more of the steps of the overall analysis process depending on the subprocess ability to perform as a standalone software. For example, tokenization and sentence splitting are co-dependent processes, so they are executed together.

4.1. Tokenization and Sentence Boundary Detection

The processes of tokenization and sentence boundary detection, as stated above, are critically dependent on each other, especially when they are rule-based as in our implementations. The complexity of the tasks for Bulgarian is moderate and it could be compared to the complexity of the same tasks for English. The latest version of the pipeline supports three different methods to deal with these problems. We have implemented a rule-based program executed through the CLaRK system (Simov et al., 2001), as well as a rule-based approach using the Stuttgart Finite State Transducer (SFST) tools (Schmid, 2005). The processing pipeline also offers a naïve RegEx Java tokenizer as a fail-safe baseline technology.

4.2. Morphological Tagging

The POS tagging in Bulgarian is more complex than the same task in English. Bulgarian is also an analytical language, but with rich word inflection. Although we often refer to this task as POS tagging, it is more accurately for it to be defined as *morphosyntactic annotation* or *morphological tagging*, because of the big variety of grammatical features and their interdependance. To tackle the complexity of the problem in an adequate way we use the full form of 680 tags of the BulTreeBank Morphosyntactic Tagset (BTB-TS) (Simov et al., 2004b), which is the original tagset of the BulTreeBank (Simov et al., 2004a). Its positional encoding of different morphosyntactic features allows us to better train statistical models for tagging and parsing (see Section 4.4.) as it provides us with the most important linguistic features of the word forms.

4.2.1. Guided Learning System — GTagger

The best solution for the POS tagging problem that BTB-LPP offers is the guided learning system described in Georgiev et al. (2012) — GTagger. The authors report accuracy results as high as 97.98 % for their best configuration using BTB-TS. This result can be considered the state of the art for Bulgarian. However, this result is archived when the input to GTagger is already tagged with all possible tags for each token — similarly to the morphological dataset, described above.

BTB-LPP provides such input for GTagger exploiting the other statistical POS Tagger, based on SVM Tool, and the rule- based algorithm that tags some tokens with a list of

[2] http://www.bultreebank.org/btbmorf/

[3] http://www.bultreebank.org/dpbtb/

[4] http://ilk.uvt.nl/conll/index.html

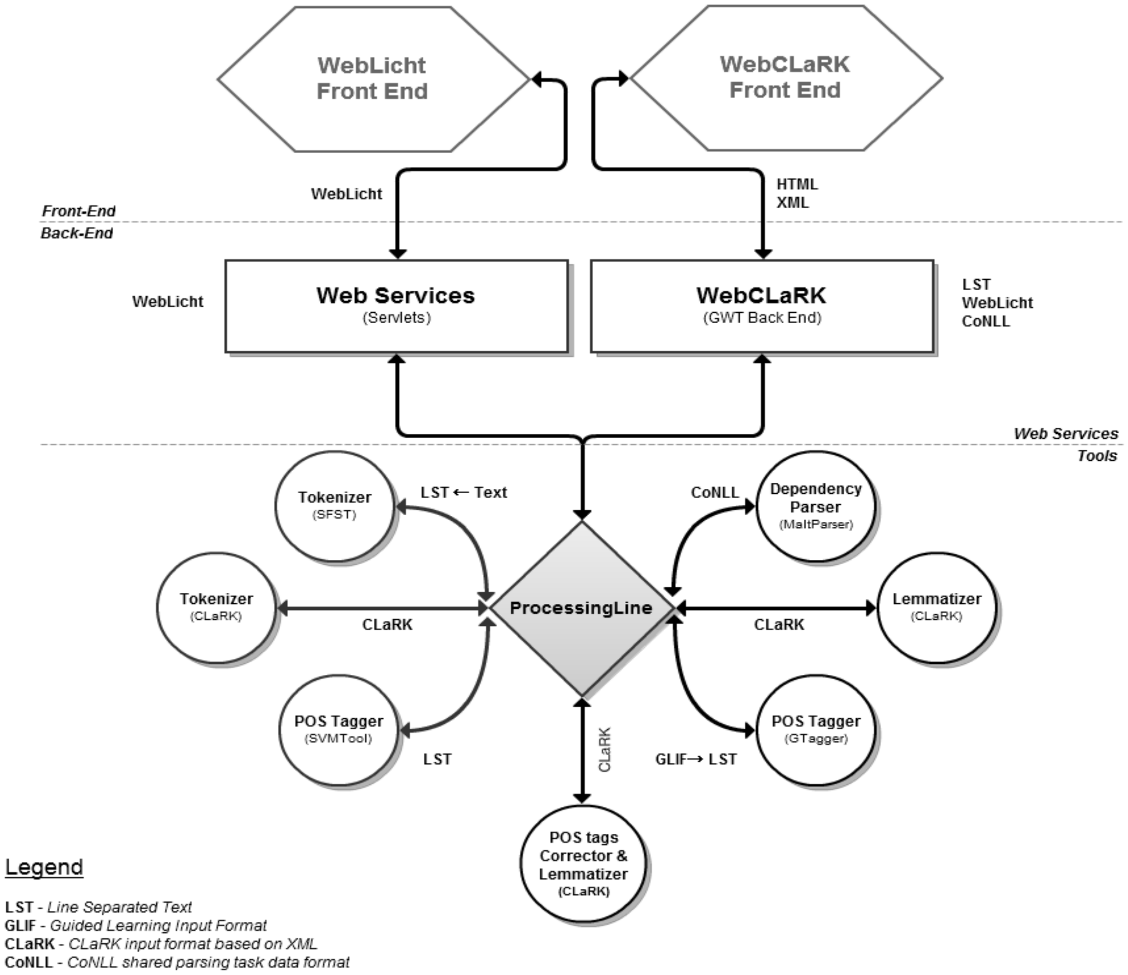

Figure 1: Processing Pipeline Architecture: modules, back-ends and front-ends.

the best possible candidate tags according to Popov et al. (1998) and Popov et al. (2003). This rule-based algorithm is described in the **Preparing Algorithm** part of Section 4.2.3..

The machine learning powered module in the tagging system uses the guided learning framework, which has achieved state-of-the-art results for English (Shen et al., 2007). The framework has also been successfully employed to achieve successful results for Icelandic (Dredze and Wallenberg, 2008), whose morphological complexity is comparable to the one of Bulgarian. The model was trained on the BulTreeBank dataset described in Section 3.1. Its authors use the original feature set as described by Shen et al. (2007), but they have also allowed prefixes and suffixes of length 9 as in Toutanova et al. (2003) and Tsuruoka and Tsujii (2005). They have also extended the features using the set of possible tags proposed by the morphological lexicon described in Section 3.2..

4.2.2. SVM Tagger

The second solution to the POS tagging problem provided in BTB-LPP uses the SVMTool (Gimenez and Marquez, 2004), which is a SVM-based statistical sequential classifier. It is built on top of the SVMLight (Joachims and Schlkopf, 1999) implementation of the Support Vector Machine algorithm (Vapnik, 1999). Its flexibility allows it to be trained for an arbitrary language as long as it is provided with enough annotated data. We used the morphologically annotated dataset as described above. A set of linguistic features was extracted from the BulTreeBank tagset to give an extra edge to the results. The accuracy that was achieved with the optimal training configuration ranged from 89% to 91% depending on the text genre. After analysing the errors in the results we noticed that some of them were repairable by employing the **Repairing Algorithm** described in Section 4.2.3. below achieving result of 94.65 % accuracy.

4.2.3. Rule-based Module

The rule-base module exploits two sources of linguistic knowledge: the morphological lexicon and the gazetteers, described above in Section 3.1., and the set of 70 disambiguation rules, implemented in Simov and Osenova (2001) and Savkov et al. (2011).

The rules are hand-crafted and then arranged as an algorithm in a specific order in a way similar to Hinrichs and Trushkina (2004). We employ two versions of the algorithm in the Processing Pipeline: one repairs the results of the SVMTool (see Section 4.2.2.) and the other narrows down the decisions on the guided learning system (see Section 4.2.1.).

These rules work on an input in which the tokens are annotated with all possible tags provided by the morphological lexicon. First the algorithm looks up the morphological dictionary and retrieves all possible tags for each token in the text. Then the rules can narrow down the possible tags for a given word by selecting one of the possible tags. In the rest of the cases all possible tags remain in the annotation. They were designed to achieve higher precision even at the cost of low recall. We have tried to keep their margin of error around 0% for the experiments that we have done and in theory that should not change on other data.

A simple example of such a rule disambiguates the tagging of the word *стола* (STOLA), which may be interpreted both as *the chair* `Ncmsh` and as the count form of *chairs* `Ncmt`. In case the previous word is a number or numeral or is a plural adjective form the rule chooses the first tag form. It is important to note that the order of the rules is crucial for the result. The algorithm was implemented using cascaded regular grammars in the CLaRK system (Simov et al., 2001).

Repairing Algorithm This algorithm use the rule-based module to repair some errors of the statistical tagger — in this case the SVM Tagger. It is designed to find the places where the linguistic evidence suggests that the statistical model is wrong and try to pick the best possible choice in cases where the linguistic evidence suggests more than one answer. There are four case scenarios that describe how the final decision is made:

- if the rule-based module yields a single tag, it is assumed to be the final decision;
- if the rule-based module yields multiple tags and one of them is also suggested by the SVMTool, that tag is the final decision;
- if the rule-based module yields multiple tags and none of them is also suggested by the SVMTool, then all possibilities are kept and the decision is deferred to a later stage;
- if the rule-based module does not output any tags, e.i. cannot recognise the word, the SVMTool suggestion is kept as final.

By using this version of the algorithm, we have managed to repair almost 30% of the errors made by the SVM Tagger shifting the accuracy of the joint analysis to 94,65%, after taking the most probable tag in the cases when the rule-based module can not take a decision.

Alternatively, all cases where a final decision was not possible even after applying the context rules the decision can be made by applying the other statistical model (see Section 4.2.1.).

Preparing Algorithm This version of the rule-based module was created to pre-process the data before feeding it to the guided learning system (see Section 4.2.1.), which performs at its best when provided with a small set of tags to choose from (Georgiev et al., 2012). In this case the algorithm tries to narrow down the list of the possible tags as much as possible on its own and then lets the statistically trained system make the final choice.

Words	Tags
Той	**Ppe-os3m**
обаче	*Cc*; **Dd**
няма	Afsi; Vnitf-o3s; Vnitf-r3s; Vpitf-o2s; Vpitf-o3s; **Vpitf-r3s**
възможност	**Ncfsi**
да	*Ta*;**Tx**
следи	*Ncfpi*; *Vpitf-o2s*; *Vpitf-o3s*; **Vpitf-r3s**; *Vpitz-2s*
...	...

Table 1: Sample fragment showing the possible tags suggested by the lexicon. The tags that are further filtered are in italic; the correct tag is in bold (Georgiev et al., 2012)

The sample output of this algorithm presented in Table 1 shows an example where the rules were able to identify the correct tag (see следи, sixth line) and an example where the rules have failed to do so (see няма, third line). In the first case GTagger will receive just one tag as input. In the second case all six tags.

The two algorithms presented in this section are packed in two of the modules of BTB-LPP. And although they are capable of performing their analysis independently, their independent results are of less importance, because the linguistic knowledge that they are based on is finite and they yield no results when presented with completely unknown or irregular data. Thus it is recommended that they are used with their respective statistical tools.

4.3. Lemmatization

The lemmatization module comprises a set of transformation rules that we have developed, based on the morphological lexicon (see Figure 2). They were implemented via finite state automata in the CLaRK system instead of word forms directly being looked up in the lexicon. We motivate our decision with its faster operation speed. Furthermore, the rules were based on the morphological dictionary, presented above. We also believe that these rules can be used on unknown words in order to produce some guessing about their word lemmas.

In theory, the lemmatization should be a deterministic process, but in some cases more than one lemma is assigned to a word form. This outcome can be expected hen the word form is ambiguous with respect to what the base form might be, and the disambiguation process requires some bigger context or other type of analysis. In these cases the lemmatizer will let the decision to be postponed for a later stage of analysis.

The lemmatization module can be executed on its own as a separate module, but it is also incorporated into the other modules relying on lemmatization. Thus, unless an experiment is deliberately aimed at separating this step from the rest of the analysis, it should be used as part of the modules

a. *if* **pos-tag = POS-Tag** *then*
{*remove* **OldSuffix**; *concatenate* **NewSuffix**}

b. *if* **pos-tag = Vpitf-o1s** *then*
{*remove* -ох;*concatenate* -а}

Figure 2: Examples of lemmatisation transformation rules in a. and replacement rule for *четох* (I read) in b.

with morphosyntactic rules.
Combining the lemmatization rules with the best result on morphological tagging results in more that 95 % accuracy.

4.4. Parsing

For the parsing part of BTB-LPP we trained the MaltParser tool (Nivre and Hall, 2005) on the dependency version of the BulTreeBank using an off-the-shelf configuration for Bulgarian (Marinov, 2009). The original model trained with this configuration also was based on the BulTreeBank, however the set of dependency relations was changed. The model uses a set of features generated from different positions of the tags in the BulTreeBank tagset. For example, the tag `Ncfsi` stands for `noun`, `common`, `feminine`, `singular`, `indefinite`, the latter three being recognized as training features. The training features based on the POS-tag of each word are included in the CoNLL version of the training dataset.

DepRel	Pr	Re	DepRel	Pr	Re
prepcomp	98.48	98.36	adjunct	67.12	65.95
clitic	95.02	99.29	comp	90.06	92.06
ROOT	94.97	88.32	subj	84.12	87.87
mod	91.54	90.61	punct	99.64	100
obj	82.15	81.32	conj	97.83	98.80
conjarg	82.85	86.29	xsubj	40.19	71.92
xcomp	88.08	78.70	indobj	63.97	63.08
xadjunct	55.90	58.25	marked	96.05	97.06
pragadjunct	47.78	63.92	xmod	75.06	74.31
xprepcomp	77.27	80.95	**Average**	**80.43**	**83.00**

Table 2: Precision and Recall results of the MaltParser for each dependency relation

Although the precision and recall measures reported in (Marinov, 2009) are better than our preliminary results, we prefer our set of relations because it is basis for the development of future processing modules. Especially the semantic one. Thus, our future work will be on the improvement of the dependency parsing.

5. Conclusion and Future Work

Here we describe a linguistic processing pipeline for Bulgarian that produces analysis on the morphological and syntactical levels. The analysis results are encouraging enough. Thus, we believe that the components of the pipeline are useful linguistic tools that will be of service both separately and as a whole. We also intend to use them for supervised processing of Bulgarian data that will extend the volume and variety of texts in the BulTreeBank.

Model	**BTB-LPP**	**(Marinov, 2009)**	
		+FEATS	Optimized
LAS:	**84.29**	86.09	84.81
UAS:	**88.30**	89.48	88.42
Label Accuracy:	**90.08**	-	-

Table 3: Parser performance results comparison. Comparing our results to the models described by (Marinov, 2009)

One immediate application of BTB-LPP is the implementation of a semantic analysis module which produces Minimal Recursion Semantics structures by applying transformation rules as described in (Simov and Osenova, 2011). This module uses the lemmas, the morphosyntactic tags and the dependency analyses in order to do its task.
There are a number of improvements that can be made to the current modules of the processing pipeline. For example, the current configuration of the MaltParser can be improved by the introduction of partially parsed input. Some partial syntactic trees can be derived using context rules. We believe that the parsing accuracy may increase significantly if the parser is trained on such data. Also, some partial grammars can be used for correcting the dependency analyses produced by MaltParser. With respect to the morphological tagging we plan to improve the result of GTagger with a better guesser as much as the main errors are related to acronyms and numbers in the text.
On the technical side of things, we are working on the web platform WebCLaRK that should feature BTB-LPP among other services. We are also working on upgrading our software to a state that allows parallelization of the processes, which will enable us to process larger slabs of data and will also significantly increase the processing speed of smaller queries, which is a necessity for online services.

6. Acknowledgements

The research presented above has been supported by the EU FP7 project 231720 EuroMatrixPlus.

7. References

L. Burnard and S. Bauman, 2007. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative Consortium, p5 edition.

M. Dredze and J. Wallenberg. 2008. Icelandic data driven part of speech tagging. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics: short papers*, pages 33–36, Columbus, Ohio, USA. ACL '08.

G. Georgiev, V. Zhikov, P. Osenova, K. Simov, and P. Nakov. 2012. Feature-rich part-of-speech tagging for morphologically complex languages: Application to bulgarian. In *EACL 2012*.

J. Gimenez and L. Marquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

E. Hinrichs and J. Trushkina. 2004. Forging agreement: Morphological disambiguation of noun phrases. *Research on Language and Computation*, 2(4):621–648, December.

E. W. Hinrichs, M. Hinrichs, and T. Zastrow. 2010. Weblicht: Web-based lrt services for german. In *Proceedings of the ACL 2010 System Demonstrations*, pages 25–29, Uppsala, Sweden.

T. Joachims and B. Schlkopf. 1999. Making large-scale svm learning practical. In C. Burges and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.

S. Marinov. 2009. *Dependency-Based Syntactic Analysis of Bulgarian*. Ph.D. thesis, University of Gothenburg.

J. Nivre and J. Hall. 2005. Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 13–95.

P. Osenova and K. Simov. 2002. Learning a token classification from a large corpus. (a case study in abbreviations). In *Proceedings of the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics*, pages 16–28, Trento, Italy, August 5-16.

D. Popov, K. Simov, and S Vidinska. 1998. *A Dictionary of Writing, Pronunciation and Punctuation of Bulgarian Language*. Atlantis KL, Sofia, Bulgaria. in Bulgarian.

D. Popov, K. Simov, S. Vidinska, and P. Osenova. 2003. *Spelling Dictionary of Bulgarian*. Nauka i izkustvo, Sofia, Bulgaria.

A. Savkov, L. Laskova, P. Osenova, K. Simov, and S. Kancheva. 2011. A web-based morphological tagger for bulgarian. In Daniela Majchrkov and Radovan Garabk, editors, *Natural Language Processing, Multilinguality*, pages 126–137, Modra, Slovakia, November, 2011. Slovko 2011, Tribun, EU.

H. Schmid. 2005. A programming language for finite state transducers. In *Proceedings of the 5th International Workshop on Finite State Methods in Natural Language Processing*, Helsinki, Finland.

K. Shen, G. Satta, and A. Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 760–767, Prague, Czech Rebublic. ACL '07.

K. Simov and P. Osenova. 2001. A hybrid system for morphosyntactic disambiguation in Bulgarian. In *Proceedings of the EuroConference on Recent Advances in Natural Language Processing*, RANLP '01, pages 5–7, Tzigov chark, Bulgaria.

K. Simov and P. Osenova. 2011. Towards minimal recursion semantics over bulgarian dependency parsing. In *Proceedings of Recent Advances in Natural Language Processing*, page 471478, Hissar, Bulgaria, September.

K. Simov, Z. Peev, M. Kouylekov, A. Simov, M. Dimitrov, and A. Kiryakov. 2001. Clark - an xml-based system for corpora development. In *Proceedings of the Corpus Linguistics 2001 Conference*, Lancaster, UK.

K. Simov, P. Osenova, S. Kolkovska, E. Balabanova, and D. Doikoff. 2004a. A language resources infrastructure for bulgarian. In *Proceedings of LREC 2004*, pages 1685–1688, Lisbon, Portugal.

K. Simov, P. Osenova, and M. Slavcheva. 2004b. Btb-tr03: Bultreebank morphosyntactic tagset. Technical report, Linguistic Modelling Laboratory, Bulgarian Academy of Sciences, April.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 173–180, Edmond, Canada. NAACL '03.

Y. Tsuruoka and J. Tsujii. 2005. Bi-directional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the Conference in Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada. HLT-EMNLP '05.

V. N. Vapnik. 1999. *The nature of statistical learning theory*. Springer, New York, 2nd edition.

T. Váradi, S. Krauwer, P. Wittenburg, M. Wynne, and K. Koskenniemi. 2008. Clarin: Common language resources and technology infrastructure. In N. Calzolari (Conference Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odjik, S. Piperidis, and D. Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.