

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/266489273>

A Web-based Morphological Tagger for Bulgarian

Article

CITATIONS

5

READS

92

5 authors, including:



[Aleksandar Savkov](#)

University of Sussex

8 PUBLICATIONS 46 CITATIONS

[SEE PROFILE](#)



[Laska Laskova](#)

4 PUBLICATIONS 10 CITATIONS

[SEE PROFILE](#)



[Petya Osenova](#)

Sofia University "St. Kliment Ohridski"

143 PUBLICATIONS 1,162 CITATIONS

[SEE PROFILE](#)



[Kiril Simov](#)

Institute of Information and Communication Technologies at Bulgarian Academy of ...

179 PUBLICATIONS 1,293 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Cognitive Technologies in Big Data Analysis [View project](#)



Open Biodiversity Knowledge Management System [View project](#)

A Web-based Morphological Tagger for Bulgarian

Aleksandar Savkov, Laska Laskova, Petya Osenova,
Kiril Simov, and Stanislava Kancheva

Linguistic Modelling Lab,
Institute for Information and Communication Technologies,
Bulgarian Academy of Sciences

{savkov, laska, petya, kivs, stanislava}@bultreebank.org

21. October 2011

Abstract

In this paper, we present a web-based morphosyntactic module for Bulgarian, which includes a statistical tagger and a lemmatizer. Both tools are implemented as a pipeline, which comprises an SVM-based tagger, a lexicon look-up component, a set of morphosyntactic context rules and a lemmatizer. The input and output of each component is defined according to the WebLicht format. Thus, ensuring a better workflow and compatibility with other NLP architectures for Bulgarian and other well-processed languages.

1 Introduction

Recently, the NLP community focused on two perspectives: integrating the existing resources and tools for various languages, and making them available on the web for the public community. The time of compiling various resources and tools is far from over, although many such tools and resources already exist. However, most of them are not accessible or hard to integrate into user-friendly 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* to the area of humanities.

To be in line with these most recent requirements, our group has started to integrate our resources in pipelines equipped with the necessary web services. Our goal is not just to publish the available resources and tools, but also to improve them according to the experience we gained during their exploitation within several projects. We have started with a language infrastructure for Bulgarian, as developed within the BulTreeBank project (Simov et al., 2004). This infrastructure included the following language resources and tools: a text

archive of more than 100 mil. running words, a morphologically annotated corpus of 1 mil. tokens, a syntactic Treebank of 214 000 tokens, various lexicons, a morphological analyser, partial grammars, named entity analysers. Some of our resources have been expanded and improved in time so we decided to re-implement or re-train some of our tools. These processes rely on new approaches, new parameters and new linguistic knowledge.

We aim at providing at least the following web-based services accessing our resources and tools:

- Language Resources Services:
 - Concordance over plain text;
 - Concordance over annotated text.
- Language Technologies Services:
 - Morphological analysis of documents provided by the users;
 - Lemmatization of documents provided by the users;
 - Syntactic analysis of documents provided by the users.

In this paper we describe our linguistic pipeline for the first two of the language technology services. The pipeline incorporates a statistical SVM-based tagger, a large morphological lexicon, a rule-based component for correction of the morphological annotation and a lemmatizer. The pipeline is implemented on the base of several language modules. For each of these modules we have implemented web services. This way, they might be easily integrated also in other pipelines. In fact, we are developing alternative modules for some of the tasks in order to provide better flexibility opportunities to the user (such as, providing a choice between several statistical taggers). The presented pipeline is made available on the web as a free service. Its components are compatible with the WebLicht data format (Hinrichs et al., 2010), which allows it to be integrated in similar pipelines.

The paper structure is as follows: Section 2 discusses the state-of-the-art taggers for Bulgarian; Section 3 describes the architecture and the implementation details of the SVM-based tagger; Section 4 focuses on the error analysis at the various stages of processing; and Section 5 concludes the paper.

2 State-of-the-Art Morphological Analysis for Bulgarian

In this section we present the most prominent systems for morphological tagging of Bulgarian that we are aware of. The first attempts were just applications based on morphological lexicons. For example, morphological analysis has been performed by the morphological dictionary for Bulgarian (Popov et al., 1998). However, with this approach the morphosyntactic ambiguities remained unresolved. After having compiled also some annotated gold standard texts with

resolved ambiguities, the researchers have directed their efforts towards the creation of automatic taggers to handle the disambiguation task.

There are several morphosyntactic taggers trained for Bulgarian. Our group has contributed to the training of some of them, such as Simov and Osenova (2001) and Georgiev et al. (2009), while there were also other attempts, such as Doychinova and Mihov (2004). In Simov and Osenova (2001) a gold standard of 2500 sentences has been used for training a neural network system. These sentences were selected with the aim to demonstrate the most frequent ambiguities per sentence. A rule-based component was also added before the automatic analysis. The system uses a hybrid architecture integrating a rule-based methods and methods based on neural network. The neural network was trained to solve the hardest cases when the input contains a lot of ambiguities. Thus, we expected it to perform better on a simple input. That is why in the hybrid system we first applied the rule-based component, which solved some of the ambiguity problems with almost perfect accuracy. Then the neural network was applied to solve the rest of the problematic cases. The accuracy for the part-of-speech feature only is 95.25%. When all the morphosyntactic characteristics were included in the evaluation, the accuracy dropped to 93.17%. The underlying tagset was larger than 600 tags. In order to cope with the sparseness of the data, we applied two approaches: (1) we selected a corpus with a large number of ambiguities (see above), and (2) we encoded the input of the neural network in the form of a vector of morphosyntactic features in order to learn the co-occurrences among them.

Another approach described in Georgiev et al. (2009) used a simplified tagset based on contributing local features. Another smaller tagset was designed by reducing the features which do not contribute to the disambiguation task. The experiments made with the reduced tagset (about 108 tags) yielded an improvement in the accuracy of 94.43% over all features. A rule-based method trained on a larger corpus was implemented by Doychinova and Mihov (2004) using FSA methods. It achieved a precision of 98.4%.

All mentioned taggers rely on the application of a rich morphological dictionary and linguistic rules. However, they differ in their automation methods and training data.

Additionally, several statistical POS-taggers were trained on the BulTreeBank morphologically annotated corpus. Atanas Chanev trained the TnT tagger, SVM and Example-based taggers. The parameters files are available at our web site¹. The Tree Tagger was trained by Julien Nioche within the European project LIRICS. The trained model is available at the TreeTagger web page² and it is also included in the distribution of GATE system.

Unfortunately only a few of these trained models are freely available and none of them are equipped with a web services. We consider our own tagger from 2001 to be outdated since the tagset of the training data has changed.

In this paper we present the tools that were trained on the improved versions

¹<http://www.bultreebank.org/taggers/taggers.html>

²<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger>

of the tagset and the training corpus. The new POS-tagging system takes a similar hybrid approach that includes a machine learning component based on Support Vector Machines and a rule-based component. The system is also equipped with a lemmatizer based on a morphological dictionary and a set of rules. The next section presents the architecture and the implementation of each component.

3 Implementation of the Language Pipeline

In this section we present the implementation of the pipeline performing the morphosyntactic tagging and the lemmatization of Bulgarian texts. Based on our own experience and the work of others, we directed our efforts towards a hybrid architecture combining a rule-based component and a machine learning component. There are two approaches to combining the two components. The first applies the statistical tools first and then corrects the results with the rule-based component and the second does the opposite. Our old approach using neural networks uses the latter configuration since it was possible to train the statistical tool to consider feature input and serve as a corrector. The approach presented in this paper uses a statistical tool that does not consider partially tagged input, thus imposing the statistical tool to precede the context rules. We also implemented a lexicon- and rule-based lemmatizer that works with the output of the POS-tagger.

3.1 Statistical POS-tagging

The machine learning component for POS-tagging wraps around the SVMTool (Giménez & Márquez, 2004). The SVMTool is a generator of sequential taggers that uses the SVMLight (Joachims & Schölkopf, 1999) implementation of Support Vector Machines (Vapnik, 1999). It uses the one-vs-all strategy to adapt the problem of POS-tagging to the binary setting of SVMs. The binary representation of the problem is realized by splitting the problem of tagging a word with one of the tags from a tagset into multitude of problems consisting of deciding whether a word should be tagged with a specific tag or not. To achieve that, the SVMTool trains an SVM for each tag in the tagset and then uses them to determine the tag for the word in focus.

The flexibility of the SVMTool allows it to be trained for an arbitrary language as long as it is provided with annotated data. We extracted 650 000 manually morphologically annotated tokens from the BulTreeBank for the purpose of the training. The accuracy we achieved with the optimal training configuration ranged from 89% to 91% depending on the genre of the text. On newspaper texts the results are closer to 91%, while the performance on fiction is closer to 89%. These results are further improved by the rule-based component described next in Section 3.2.

3.2 Lexicon Look-up and Morphosyntactic Rules

The task of the next component of the morphosyntactic tagger is to correct some of the erroneous analyses made in the process of the statistical POS-tagging. The typical errors made by the SVM-based analysis are described in Section 4 below. Often happens that the suggestion of the SVMTool is a good guess, but has a few wrong morphological features that still make the tag fairly inaccurate. The error corrections are made based on two sources of linguistic knowledge – the morphological lexicon and the set of 70 context based rules. The context rules are designed in such a way that they aim at achieving higher precision even at the cost of low recall. The lexicon look-up is implemented as cascaded regular grammars within the CLaRK system (Simov et al., 2001). The lexicon is an extended version of the dictionary described in Popov et al. (1998) and covers more than 110 000 lemmas. Additionally, a set of gazetteers were incorporated within the regular grammars. After the lexicon look-up application, 97.01% of the tokens (in the test corpus) received morphological analysis from the lexicon and the gazetteers.

In the examples below a WebLicht encoding of the data is presented with some additional attributes for the information from the lexicon. Each token is represented by the element `ns2:token`; all possible morphosyntactic analyses resulting from the lexicon queries are contained in the `@aa` attribute; the analysis produced by the SVMTool is encoded as the `@svm` attribute; and the final analysis is encoded in the `@ana` attribute. Tokens that are not listed in the morphological dictionary have no `@aa` attributes. In these cases the the POS-tagger analysis is considered the best option. If the attribute `@aa` is attached to a token, we assume that the value of the final analysis has to be among the values returned from the lexicon. Thus we have give precedence to the analysis based on the context rules. Any word form that is listed in the lexicon and receives one distinct analysis from the context rules is considered to be correctly analysed. The results of the statistical POS-tagger serve two purposes:

1. in case the context rules produce more than one analysis and one of them coincides with the analysis suggested by the SVMTool, the latter is considered to be true;
2. in case the context rules produce more than one analysis and the SVMTool suggestion is not among them, we merge all tags into one giving precedence to the analyses provided by the lexicon. In case the intersection does not reach a certain threshold (i.e. none of the morphological features are the same), the ambiguous information is left for later processing, including the SVMTool analysis.

About 30% of the errors are corrected merely by consulting the morphological dictionary and the gazetteers. The context rules take into account the output of the dictionary queries and the intrasentential morphosyntactic context. They

are implemented as value restriction constraints in the CLaRK system. Each constraint has a target section and a source section. The target section defines the token for which the rule will take a decision, and its context. For example, the phrase *големи котешки очи* (big cat eyes) is represented as:

```
<ns2:token aa="A-pi" ana="A-pi">големи</ns2:token>
<ns2:token aa="A-pi;Amsi" svm="Ncfpi">котешки</ns2:token>
<ns2:token aa="Ncnpi" ana="Ncnpi" svm="Ncnpi">очи</ns2:token>
```

In this case, the target of the rule is the element:

```
<ns2:token aa="A-pi;Amsi" svm="Ncfpi">котешки</ns2:token>
```

and the rest is the context that triggers the application of the agreement rule. In this case, the morphologically ambiguous form *котешки* (cat-adj.m.sg. or cat-adj.pl.), is assigned the masculine singular adjective tag "Amsi" because of the context agreement rule. The result is:

```
<ns2:token aa="A-pi" ana="A-pi">големи</ns2:token>
<ns2:token aa="A-pi;Amsi" ana="A-pi" svm="Ncfpi">котешки</ns2:token>
<ns2:token aa="Ncnpi" ana="Ncnpi" svm="Ncnpi">очи</ns2:token>
```

With respect to their accuracy, we consider the rules to be "sure" not in the sense that they achieve the highest possible recall, but in the sense that they achieve near perfect precision when tested on the BulTreeBank corpus (over 1 000 000 tokens). Some rules perform slightly worse, but we include them in the set if the precision is higher than 95%. However, they are the minority of the rules in the set – about 20%. Here is one example of such a rule. The word form *че* (that) is ambiguous and despite the fact that it usually functions as a subordinate conjunction (Cs), it could be tagged also as an emphatic particle (Te) or even as a coordination conjunction (Cc). The context rule ignores the latter two possibilities, but still achieves high enough precision of about 99.58% on our test corpus.

There is a strict order in which the rules are applied, since some of them depend on the results of other rules. Therefore, for example, all rules that apply to elements that usually can be found in the NP internal structure, are preceded by rules that target tokens with possible noun morphosyntactic analysis.

In case there is no intersection between the lexicon set of analyses and the SVMTool outcome, and the rules do not suggest another solution, the features of the possible tags indicated by the lexicon are compared with the features of the statistically generated tag. First, the system compares the part-of-speech features of the tags and further comparison is made only if they coincide, otherwise the SVMTool decision about the word class of the token is considered completely wrong and the analysis ambiguity remains unresolved. The all other morphological features are merged as the result has to be common for both analysis sources. In the cases where that is not possible, the most numerous

lexicon analysis is preferred. In the following example the verb form has several analyses for tense and person, but always remains transitive. The suggestion of the SVMTool is that the form is intransitive, but the decision that the verb form is in present tense, third person, is correct. In order to repair this problem, the system copies the non-matching information from the @aa attribute to the result in the @ana attribute. In the example, the third position of the tag encodes perfectivity – *p* (perfective) or *i* (imperfective), and the seventh position encodes the tense – *o* (aorist) or *r* (present tense):

```
<ns2:token aa="Vpptf-o2s;Vpptf-o3s;Vpptf-r3s"
ana="Vpptf-r3s" svm="Vpitf-r3s">позабавлява</ns2:token>
```

Since the lexicon suggestions exclude the perfective verb analysis, the third feature has been rewritten. According to the information encoded in the inflectional dictionary, the value for the present tense for this token (seventh tag position) is possible, so it has not been changed. The percent of ambiguous tokens remaining in the output is about 2.55%. In our future work we will extend the set of rules in order to suggest the most probable solution for these cases.

3.3 Lemmatizer

The morphological analysis produced by the POS-tagger and the context rules provides the needed information for a very reliable rule-based lemmatization process. Thus we implemented a rule-based lemmatization module built from the morphological lexicon mentioned above. The lexicon lists word forms, morphological tags and lemmas, but since it is too big we generated FSA-based rewrite rules that achieve the same effect. The lemmatizer identifies the suitable replacement rules and applies them to the word form. The rules are based on the following intuition:

*if tag = **Tag** then {remove **OldSuffix**; concatenate **NewSuffix**}*

where **Tag** is the tag of the word form, **OldSuffix** is the string which has to be removed from the end of the word form and **NewSuffix** is the string which has to be concatenated to the beginning of the word form in order to produce the lemma. Here is an example of such a rule:

*if tag = **Vpitf-o1s** then {remove **-ox**; concatenate **-a**}*

The application of the rule to the past simple verb form for the verb *чемоx* (remove: **-ox**; concatenate: **-a**) gives the lemma *чема* (to read). Additionally we use the generated rules for unknown words like guesser word forms: **ox* and **tag**=*Vpitf-o1s*. In these cases the rules are ordered.

In order to facilitate the application of the rules, we attach them to the word forms in the lexicon. In this way, we gain two things: (1) we implement the lemmatization tool as a part of the regular grammar for lexicon look-up, discussed above and (2) the level of ambiguity is less than 2% for the correct tagged word forms. In case of ambiguities we produce all the lemmas. After

the morphosyntactic tagging, the rules that correspond to the selected tags, are applied.

3.4 Web Service Architecture and WebLicht Compatibility

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 major current problems is that many of the technologies are incompatible with each other. Although some have chosen to implement general data-encoding standards like the TEI, many linguistic tools and resources develop their own operational annotation formats. And very few choose to implement common interfaces. These facts impede the interoperability of language technologies. To make sure that the morphological analyzer for Bulgarian can be shared and reused properly, we decided to adopt some of the good ideas of the new-generation Linguistic Resources and Technologies project D-SPIN, part of the CLARIN Project. Its platform WebLicht is a web-based service environment that allows the users to integrate and use various language resources and tools (Hinrichs et al., 2010). The purpose of the platform is to make possible for the scientists to upload their resources and share their tools in one place with common operational and annotation formats, thus improving their collaboration. Although the morphological analyser is not a large scale project and does not intend to produce a quite so sophisticated web-based environment, many of the solutions and ideas that WebLicht provides, are applicable in its context.

The WebLicht platform addresses the two main problems of research collaboration: different data annotation formats, and technical issues and support of tools/technologies. A common data annotation format allows the resources and tools registered on the platform to be chained together forming flexible linguistic chained processes. Although it is not required, the use of the Text Corpus Format (TCF), which is a stand-off XML annotation format developed within the D-SPIN project, is recommended. The structure of TCF documents is based on information blocks whose elements are connected by references allowing the annotation of different kinds of information with possible overlapping scope in one file. TCF annotation also makes adding and removing information from the document painless and error-prone, which is a key advantage when putting together custom tools such as the one we described in this paper. The WebLicht architecture requires all the tools to be implemented as RESTful web services that are also recommended to work with the TCF. Web services offer a simple and painless solution to the problem of installing and configuring tools by allowing the authors to host and support them while they are being used in more complex tools. In this way the different steps of a linguistic analysis may be carried out by different tools in different places producing one final result. Adopting the TCF document annotation and implementing all analysis steps as web services not only allows us to share our specific tools through the WebLicht platform, but also to develop, extend and improve our complex tools with less effort.

4 Error Analysis

This section presents some qualitative error analysis of the hybrid POS-tagger system. Errors occur in the process of statistically identifying the parts of speech, as well as in identifying some of their morphological characteristics.

Wrong part-of-speech tag

1. The part-of-speech tag assigned by the SVMTool is wrong for a word form with known distinct analysis. The following more frequent sub-cases are observed:
 - a) the word is considered a participle instead of a finite verb: **престана** *да говори* (she/he **stopped** speaking);
 - b) the word is considered a noun instead of a participle – *посърнали* (haggard-3rd peron, pl);
 - c) the word is considered a noun instead of an adjective – *сълмили* (sleepy-3rd peron, pl);
 - d) the word is considered an adjective instead of a noun – **капчица** *любов* (**droplet** love, droplet of love). This error is typical for NP phrases of type - NN (or NP NP). It can be explained by the fact that adjectives usually precede nouns when in a NP.

An interesting case for this automatic tagger is the analysis of the family names. If there is a sequence of a given name and a family name of a person in the sentence, the SVM Tagger would annotate correctly the family name. However, if there is only a family name in the sentence, the SVM Tagger would annotate it like an adjective or a participle (regardless of the capital letter).

2. The second case is when the assigned part-of-speech tag is wrong for an ambiguous word. The following more frequent sub-cases are observed:
 - a) the word is identified as an adjective instead of an adverb – *лицето ѝ беше извънредно слабо* (her face was **extremely** thin);
 - b) the word is identified as an adverb instead of an adjective – *лицето ѝ беше извънредно слабо* (her face was extremely **thin**).

It is worth noting that the forms from the examples above, *извънредно* (extreme or extremely) and *слабо* (thin or thinly), are part-of-speech homonyms in Bulgarian. They can be realized as an adjective or an adverb depending on the context. Other examples for such homonymy are: *преди* – a preposition (previous to) or an adverb (before); *си* – a pronoun (personal or possessive reflexive) or a verb (be-2nd p.sg.).

Wrong morphosyntactic features

The SVMTool often makes errors in determining only a few of the morphological features of the POS-tag. The errors usually occur when determining the classes

of the verbs and the pronouns. For example, the analyses of the verb forms have the following problems:

- whether the verb is transitive or intransitive;
- whether the verb is personal or impersonal;
- whether the aspect of the verb is perfective or imperfective;
- whether the tense is present (1stp.sg.) or past simple (2ndp.sg. or 3rdp.sg.)

These errors can be directly corrected by the morphological dictionary when the word form is not ambiguous. However, the problems with the verb aspect remain. The analysis of pronouns is also difficult, because very often one form expresses more than one meaning. For example, the pronoun *my* is homonymous denoting both a personal pronoun (I told *him*) or a possessive pronoun (*his* book). Another typical error is the wrong gender of a noun. However, this one can be corrected by the information from the morphological dictionary.

In conclusion, some of the errors of the SVM tagger can be overcome by the application of the dictionary, while others can be corrected by a set of linguistic rules. However, a strategy was needed for a final selection of the correct tag among the competing suggestions (see Section 3.2). Although in many cases the described strategies worked, in some cases they led to a wrong selection.

The total accuracy obtained after applying the lexicon queries and the morphosyntactic context rules is 94.65%. The corrections affect predominantly the morphosyntactically unambiguous tokens – about 75% are directly assigned with the only possible POS-tag from the lexicon. The results for ambiguous tokens are considerably worse: about 30% of the rule-based decisions are correct, 20% of the tokens were assigned all their possible analyses or a subset of them. As expected, most of the errors are caused by the homonymy of the verb forms.

The results demonstrate that the hybrid system of a statistical tagger, a morphological dictionary and a set of context rules employing a more sophisticated tagset could be as accurate as the machine learning tools using a simplified tagset of Georgiev et al. (2009). A true comparison of the results of Doychinova and Mihov (2004) is not possible because their tool and related data are not available.

5 Conclusion

In this paper we presented a web-enhanced morphological tagger and a lemmatizer for Bulgarian. The work has built on our previous experience with various taggers and tagsets. The processing has been organized in a pipeline, which includes an SVM-based tagger, queries to a morphological dictionary of Bulgarian and a set of linguistic context rules. The decisions behind the pipeline have been motivated by a number of factors, such as whether the word is morphologically ambiguous, and whether the SVMTool suggested a tag listed as possible in the dictionary. The system that we present is compatible with the TCF format, which makes it usable within the CLARIN community, and thus part of integrated language architectures.

Our plans for future work follow two directions: (1) improvement of the strategy for selecting the correct tags while experimenting with different combinations of processing steps, and (2) adding a shallow parser on the top of the morphological analyser and the lemmatizer.

6 Acknowledgments

This work has been supported by two European projects: EuroMatrixPlus project (IST-231720) and CLARIN (IST-212230).

References

- Doychinova, V., & Mihov, S. (2004). High performance part-of-speech tagging of bulgarian. In *Proceedings of eleventh international conference on artificial intelligence: Methodology, systems, applications (aimsa-2004)* (p. 246-255).
- Georgiev, G., Nakov, P., Osenova, P., & Simov, K. (2009, September). Cross lingual adaptation as a baseline: Adapting maximum entropy models to bulgarian. In *Proceedings of the workshop on adaptation of language resources and technology to new domains* (p. 35-38). Borovetz, Bulgaria. (In conjunction with RANLP'09)
- Giménez, J., & Márquez, L. (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)*. Lisbon, Portugal.
- Hinrichs, E. W., Hinrichs, M., & Zastrow, T. (2010). Weblicht: Web-based lrt services for german. In *Proceedings of the acl 2010 system demonstrations* (p. 25-29). Uppsala, Sweden.
- Joachims, T., & Schölkopf, B. (1999). Making large-scale svm learning practical. In C. Burges & A. Smola (Eds.), *Advances in kernel methods - support vector learning*. Cambridge, MA, USA: MIT Press.
- Popov, D., Simov, K., & Vidinska, S. (1998). *A dictionary of writing, pronunciation and punctuation of bulgarian language (in bulgarian)*. Sofia, Bulgaria: Atlantis KL.
- Simov, K., & Osenova, P. (2001, 5-7 September 2001). A hybrid system for morphosyntactic disambiguation in bulgarian. In *Proceedings of the ranlp 2001 conference* (p. 288-290). Tzigrav Chark, Bulgaria.
- Simov, K., Osenova, P., Kolkovska, S., Balabanova, E., & Doikoff, D. (2004). A language resources infrastructure for bulgarian. In *Proceedings of lrec 2004* (p. 1685-1688). Lisbon, Portugal.
- Simov, K., Peev, Z., Kouylekov, M., Simov, A., Dimitrov, M., & Kiryakov, A. (2001). Clark - an xml-based system for corpora development. In *Proc. of the corpus linguistics 2001 conference*. Lancaster, UK.

Vapnik, V. N. (1999). *The nature of statistical learning theory* (2nd ed. ed.).
New York: Springer.