Bulgarian-English Question Answering: Adaptation of Language Resources

Petya Osenova* Alexander Simov† Kiril Simov‡ Hristo Tanev§
Milen Kouylekov¶

Abstract

This paper describes the Bulgarian part of a Bulgarian–English question answering system. The Bulgarian modules are implemented as a question analysis procedure within a Bulgarian question answering system — **BulQA**. The paper presents the available language resources and corresponding technology which is used for the analysis of the questions in Bulgarian and their translation into English format necessary for answer extraction. As implementation platform CLaRK System is used.

1 Introduction

This paper describes the first steps in the development of a question answering system for Bulgarian — **BulQA**. The system is planned to have three main modules: *Question analysis module*, *Interface module*, *Answer extraction module*. The *Question analysis module* deals with the syntactic and semantic interpretation of the question. The result of this module is independent on task and domain representation of the syntactic and semantic information in the question. The *Interface module* maps the interpretation received from the first module to the input necessary for the third module. The *Answer extraction module* is responsible for the actual finding of the answer in the corresponding corpus. This architecture allows reusing some of these modules in other tasks such as - Bulgarian as source language in a multilingual question answering or Bulgarian sa target language. In general, only *the Interface module* has to be re-implemented in order to tune the connection between Bulgarian modules and the modules for the other languages.

Here we describe the current question analysis module and the *Interface module* in a Bulgarian to English question answering system. In this system the *Answer searching module* is based on the Diogene system implemented at the ITC-Irst, Trento, Italy.

The structure of the paper is as follows: first we describe the language resources and tools developed within the BulTreeBank Project, then in section 3 we discuss their adaptation for the analysis of Bulgarian questions, section 4 describes briefly the DIOGENE system for document retrieval and answer extraction, in section 5 we discuss the interface between the system BulQA for the analysis of Bulgarian questions and DIOGENE System, section 6 gives a general overview of the CLaRK System in which the modules of BulQA are implemented, the last section reports on the results of the question answering track and concludes the paper.

2 The BulTreeBank Language Resources and Tools

In this section we describe the available language resources and tools which we have adapted in order to implement *the Question analysis module* for Bulgarian. Generally, a language technology is supposed to include the following modules: tokenization and named entities recognition,

*Laboratory for Linguistic Modelling, Bulgarian Academy of Sciences, Bulgaria, petya@bultreebank.org
†Laboratory for Linguistic Modelling, Bulgarian Academy of Sciences, Bulgaria, alex@bultreebank.org
‡Laboratory for Linguistic Modelling, Bulgarian Academy of Sciences, Bulgaria, kivs@bultreebank.org
§TC-irst, Trento, Italy, tanev@itc.it
¶TC-irst, Trento, Italy, kouylekov@itc.it

morphological analyzer and disambiguator, syntactic and semantic analyzer. Most of them we have already implemented during the creation of a syntactic treebank for Bulgarian — [Simov, Popova and Osenova 2002].

2.1 BulTreeBank Language Technology

Here we list the tools that we had at our disposal before we started to implement our system:

Tokenization

There is a hierarchy of tokenizers within the CLaRK system, which tokenizes the texts in an appropriate way. Additionally, one can decide what the category of the token is and to assign it.

The Morphosyntactic analyzer

It assigns all possible analyses to the word tokens. The lexicon is too large to be loaded as one grammar in CLaRK and this is why we have divided it into several grammars which are applied in a group. The separation of the lexicon is on the basis of the frequencies of the word forms within the corpus. In this way the application has been speeded up. As it was mentioned above, together with the morphosyntactic analyzer we use the gazetteers. They are also implemented within the CLaRK system. In the places where competing analyses arise between a common word and a name or an abbreviation, we try to use the token classification strategy and the prompts of the context.

MorphoSyntactic Disambiguation

We have already implemented a preliminary version of a rule-based morpho-syntactic disambiguator, encoded as a set of constraints within the CLaRK system. This rule-based disambiguator exploits context information like *agreement between an adjective and a noun in a noun phrase*, specific positions like *a noun after a preposition*, but it also deals with some fixed phrases. The disambiguator does not try to solve unsure cases, but leaves them for further processing. Its coverage is about 80 %. For automatic disambiguation we have developed a neural-network-based disambiguator (see [Simov and Osenova, 2001]). It achieves accuracy of 95.25% for part-of-speech and 93.17% for complete morpho-syntactic disambiguation.

Partial Grammars

We have constructed grammars for:

1. **Sentence splitting.** At the moment it is fully automated and reliable only for the basic and clear cases. For solving complex and ambiguous cases this grammar is combined with supporting modules for abbreviation detection.
2. **Named-entity recognition.** Identifying numerical expressions, names, abbreviations, special symbols (see [Ivanova and Dojkoff 2002], [Osenova and Kolkovska 2002]). They are designed to work in cooperation with the morphosyntactic analyzer. If necessary, the grammars can overwrite the analysis of the morphosyntactic analyzer.
3. **Chunking.** Two basic modules have been developed: an NP chunker ([Osenova 2002], [Osenova and Kolkovska 2002]) and a VP chunker [Slavcheva 2002]. Generally speaking, the chunking process conforms to the following requirements: it deals with non-recursive constituents; relies on a clear-indicator strategy; delays the attachment decisions; ignores the semantic information; aims at accuracy, not coverage. Additionally, there are chunk grammars for APs, AdvPs, PPs and some non-problematic clauses.

2.2 BulTreeBank Language Data

From the language resources the most important for the task are the lexicons:

The Morphological Dictionary

The dictionary is an electronic version of [Popov, Simov and Vidinska 1998] extended with new words from the corpus. It covers the grammatical information of about 100 000 lexemes (1 600 000 word forms) and serves as a basis for the morphological analyzer.

The Gazetteers

Two basic lists with items, missing in the morphological dictionary, have been compiled with respect to their frequency:

1. Gazetteers of names. These consist of 15 000 items and include Bulgarian as well as foreign person names, international and national locations, organizations. The most

frequent names are additionally classified according to three criteria: (1) grammatical (gender and number); (2) semantic - with respect to an extended SIMPLE core ontology (names for different types of locations, organizations, artifacts, persons' social roles etc.) and (3) ontological - some person names were connected with specific individuals in the world and thus some encyclopedic information was provided in addition to the semantic classification. All this information was ready to be used for practical applications like Information Extraction or Retrieval, Data Mining, Question Answering etc. Special attention is paid to the names of mountains and artifacts (books, films, broadcasts), because their internal agreement does not always coincide with the external one, which is needed for the sentence analysis.

2. Gazetteers of the most frequent abbreviations. They consist of 1500 acronyms and graphical abbreviations. The acronyms' extensions were mapped against the names (mostly organizations) and therefore, assigned the same semantic and grammatical label. In cases of idiosyncratic grammatical behaviour, the relevant patterns have been added as well.
3. Gazetteers of the most frequent introductory expressions and parentheticals. This is considered to be a step towards a basic list of collocations. They were classified according to their morphological type or behavior: verbal, adverbial, linking (for conjunctions), nominal (vocatives), idiomatic etc. We use them as an extended supplementary lexicon during the phase of the syntactic annotation.

The Valence Dictionary

It consists of 1000 most frequent verbs and their valence frames and it is based on a paper dictionary (see [Balabanova and Ivanova, 2002]). Each frame defines the number and the kind of the arguments and imposes morphosyntactic and semantic restrictions over them. The semantic restrictions over the arguments are extracted and matched against the SIMPLE core ontology. The frames of the most frequent verbs are compared to the corpus data and repaired if necessary (new frames are added, some of the existing frames are deleted or fine-grained). We envisage to enlarge the coverage of the dictionary with the help of some derivational means, such as the verb prefixes.

The Semantic Dictionary

Semantic information plays a crucial role in the process of named entity recogniion. Thus, in order to support the selectional restrictions imposed by the valence dictionary and to facilitate its usage, we decided to compile a semantic dictionary along the guidelines of SIMPLE project. It is worth mentioning that we follow an extended variant of the SIMPLE core ontology. At the moment we are classifying the most frequent nouns with respect to the ontological hierarchy without specifying the synonymic relations between them. Up to now we have classified about 3 000 nouns. Recall that the named entities also have been classified with respect to the same ontology.

3 Adaptation to Question Answering Task

Although the above listed language processing tools were extensively tested during the compilation of our treebank, they needed some additional tuning to the task of question analysis. The main difference is that most of them were implemented in such a way that in unsure cases the ambiguity remained unresolved or the analysis was not produced. This tools' application was required in case when an annotator had to inspect the result of the processing.

With respect to the Question Answering task some ambiguities were resolved in the following way: (1) in ambiguities between 2 and 3 person or 1 and 3 person, always the 3 person was selected; (2) in ambiguities between present and past verb tense, the past tense was selected, etc. The first ambiguity was resolved because the questions given in CLEF are never in 1 or 2 person. Resolving between the different tenses in the question with respect to validation of the found answers is not currently supported by the Answer extraction module. Some other ambiguities we resolved on the frequency basis only — for each ambiguity class the most frequent option was selected.

The major addition with respect to the available tools was the construction of a lemmatizer for Bulgarian. We defined the lemma to be functionally determined by the wordform and its morphosyntactic characteristics. The cases of ambiguous lemmas are not resolved and all

possible lemmas are assigned to the corresponding wordform. Lemma is used later to access the semantic information from the semantic dictionary and the English equivalents in the Bulgarian–English dictionary.

Here is an example of the analysis of the question "През коя година Томас Ман получи Нобелова награда?" (in English: *Which year did Thomas Mann receive the Nobel Prize?*):

```
<analysis group="BTB">
    <PP>
        <Prep><w ana="R" bf="през">През</w></Prep>
        <NPA>
            <Pron><w ana="Pie-os-f" bf="коя">коя</w></Pron>
            <N><w ana="Ncfsi" bf="година">година</w></N>
        </NPA>
    </PP>
    <NPA sort="NE-Pers">
        <N><name ana="Npmsi" sort="PersNE">Томас</name></N>
        <H><name ana="Hmsi" sort="PersNE">Ман</name></H>
    </NPA>
    <V><w ana="Vpptf-o3s" bf="получа">получи</w></V>
    <NPA>
        <A><w ana="Afsi" bf="нобелов">Нобелова</w></A>
        <N><w ana="Ncfsi" bf="награда">награда</w></N>
    </NPA>
    <pt>?</pt>
</analysis>
```

Here each common word is annotated within the following XML element *<w ana="MSD" bf="LemmaList">wordform</w>*, where the value of attribute *ana* is the correct morpho-syntactic tag for the wordform in the given context of its usage. The value of the attribute *bf* is a list of the lemmas assigned to the wordform. Names are annotated within the following XML element *<name ana="MSD" sort="Sort">Name</name>*, where the value of the attribute *ana* is the same as above. The value of the attribute *sort* determines whether this is a name of a person, a location, an organization or some other entity.

The next level of analysis is the result of the chunk grammars. In the example there are three *NPA* elements (NPA stands for a noun phrase of head-adjunct type) and one *PP* element. Also, one of the noun phrases is annotated as a name with a sort attribute with value: *NE-Pers*.

The result of this analysis has to be translated into the format which the answer extraction module is used as input.

4 DIOGENE System in Brief

In this section we briefly describe DIOGENE System which was used for document retrieval and answer extraction. DIOGENE [Negri et al 2002] relies on the knowledge in multilingual ontology MultiWordNet [Pianta et al 2002], manually created rules for named entity recognition and question type identification, a set of handcrafted answer extraction templates and statistical information collected from the Web and off-line multilingual corpora.

In cross-language mode DIOGENE works as follows:

1. The question is processed and all the possible translations of the keywords from the source language in English are found (for the Bulgarian-English task this is performed in the BulQA system).
2. Finding correct combination of translations: We chose this combination of keyword translations (k_1, k_2,...,k_n) which has the highest frequency of co-occurrence in an English corpus (we used AQUAINT and TIPSTER collections). The main assumption is: the more often a keyword translation combination appears with the translations close to each other (in one and the same paragraph), the more plausible this combination is.
3. From keywords and their synonyms DIOGENE forms a Boolean query which is passed to Managing Gigabytes (MG) search engine. Some keywords can be deleted from the

query if it generates no hit or just a few hits. In this way several feedback loops can be performed. The output of this processing stage is a list of paragraphs where question keywords and their synonyms appear together.

4. Named entity recognition and answer extraction templates are applied to extract candidate answers. In the cross-language mode DIOGENE applies answer extraction templates just for the definition questions. Candidate answers of the factoid questions are captured using named entity recognition and proximity to the question keywords.
5. Finally, the candidate-answers of the factoid questions are evaluated using Web based answer validation technique described in [Magnini et al 2002]. On the other hand, we evaluate the answers of the definition questions using different syntactic and semantic clues, among them is the presence of hyponym of the "person" or "organization" concept, presence of definite lexical templates, etc.

The format which was necessary to be supplied to DIOGENE System was as follows:

- **Head of the question**. The head of each question depends on the interrogative word in the question and helps to determine the kind of the answer. Some examples of question heads are: *what*, *who*, *what-who* etc.
- **Type of the question**. It determines the semantic category of the possible answers.
- **Head word of the question**. It is the word in the question which provides the type of the question. It can be a non-functional word or the interrogative word.
- **Sense of the head word**. This is the sense derived from WordNet for the head word. If such a sense cannot be determined, then the value is *NIL*.
- **Part of speech of the head word**. This is the POS tag of the head word with respect to the Pentreebank tagset.
- **Position of the head word**. A digit which determines where in the question the head word is.
- **List of key words**. A list of the non-functional words in the question. Each keyword is also annotated with its part of speech.

5 Interface module

Here we describe the implemented interface module which translates the result of the question analysis module into the template necessary for DIOGENE System, which extracts the answers of the questions. The process includes the following steps:

- Determining the head of the question.

  The determination of the question head is done by searching for the chunk which contains the interrogative pronoun. There are cases in which the question is expressed with the help of imperative forms of verbs: *назовете* (to name), *кажете* (to point out; to say), *избройте* (to list; to enumerate). After the chunk had been selected we classify the interrogative pronoun within a hierarchy of question's heads. In this hierarchy some other elements of the chunks — mainly prepositions — play an important role as well.
- Determining the head word of the question and its semantic type.

  The chunk determined in the previous step also is used for determining the head word of the question. There are five cases. First, the chunk is an NP chunk in which the interrogative pronoun is a modifier. In this case the head noun is the head word of the question. In the second case the chunk is a PP chunk in which there is an NP chunk similar to the NP chunk from the previous case. Thus, again the head noun is a head word for the question. Third, the interrogative pronoun is a subject of a copula verb. In this case the head word of the question is the head noun of the complement NP chunk of the copula. Here the important moment is the distinction between the cases when the interrogative pronoun is a subject of the copula and when it is its complement. The rule covers only the subject case. The fourth case is similar, but it covers the questions whit imperative verbs. Then again the head of the question is the head noun of the complement NP chunk. The last case covers all the remaining questions. Then the head word of the question is the interrogative pronoun itself.

The semantic type of the head word is determined by the annotation of the words with semantic classes from the semantic dictionary. When there are more than one semantic classes we add all of them. The type of the interrogative pronoun is used later for disambiguation. If no semantic class is available in the dictionary, then the class 'other' is assigned.

- Determining the type of the question.

  The type of the question is determined straightforwardly by the semantic type of the head word.

- Determining the keywords of the question and their part of speech.

  The keywords are determined by the non-functional words in the question. Their part of speech is determined by a mapping from the Bulgarian tagset into the tagset used by DIOGENE System. Sometimes it is possible to construct multi-token keywords like names (Thomas Mann), terms or collocations (Nobel prize). To some extend this is done after the translation into English.

- Translation of the question head word and the keywords into English.

  We have two Bulgarian–English dictionaries: one for the common vocabulary and one for the names. The dictionary of names contains the transliterations of most frequent names that we found in Bulgarian corpus and in the English corpus. This dictionary is necessary because a vast amount of foreign names do not follow the same transliteration principles for Bulgarian. For instance, Washington as a name of the president George Washington, the state Washington and the capital of the USA is written as *Вашингтон* (Vashington), which follows the literal traditional transliteration, i.e. letter by letter. However, in all other cases this name is written as *Уошингтън*, which follows the new principles of transliteration, i.e. closer to the original pronunciation of the word. For the names which are not in the dictionary we apply the transliteration for Bulgarian into Latin as it is defined by the Bulgarian Post Services. Note that the last solution is far form perfect and it has to be improved afterwards. The main problem is that this transliteration does not take into account the sound representation of the names in the original language. For instance, the name Thomas (*Томас* in Bulgarian) will be transliterated as Tomas without 'h'. For that reson, this problem will require much more work in future. Some names of famous people and places are kept as one whole expression in the dictionary. For example, 'Thomas Mann' is a multi-token name in the dictionary. This helps us during the translation phase, because of the following: if we take the two names separately, we can receive, wrongly, also Thomas Man as a potential translation, where 'Man' is transliterated with one 'n'. For the words which have more than one translation we give all possibilities.

  Another very useful resource is the collocation dictionary for English. For example, the chunk *Нобелова награда* (Nobel prize)is a collocation in English, but we also translate it into Bulgarian as 'Nobel award'. If we have a collocation dictionary we could use it in order to recognize such multi-token expressions. In future work we also will try to use Internet to judge between the different possibilities. For the above examples, 'Nobel prize' is much more frequent than 'Nobel award'.

  Here we give the result of the analysis for the above question:

```
<analysis group="BTB">
  <QHead qhead="what" qtype="time">
    <PP>
      <Prep><w ana="R">През</w></Prep>
      <NPA>
        <Pron><w ana="Pie-os-f">коя</w></Pron>
        <N><w ana="Ncfsi" sort="time" eng="year">година</w></N>
      </NPA>
    </PP>
  </QHead>
  <NPA sort="NE-Pers">
    <N><name ana="Npmsi" sort="PersNE" eng="Thomas">Томас</name></N>
    <H><name ana="Hmsi" sort="PersNE" eng="Mann;Man">Ман</name></H>
```

```
  </NPA>
  <V><w ana="Vpptf-o3s" eng="get,receive;obtain">получи</w></V>
  <NPA>
    <A><w ana="Afsi" eng="Nobel">Нобелова</w></A>
    <N><w ana="Ncfsi" sort="other" eng="prize;award">награда</w></N>
  </NPA>
  <pt>?</pt>
</analysis>
```

Here the new element is *QHead* which determines the chunk head of the question. It has two attributes: *qhead* which has as a value the question head — *what* in the example; and *qtype* which has as a value the type of the question — *time* here. Some of the words received additional attributes: *sort* for the semantic class of the word, and *eng* for the possible translations into English.

- Filling the template.

  This step means the conversion of the information that has already beed explicated into the form necessary for the DIOGENE System. Here also we try to produce multi-token keywords. In the example above, such a keyword is Thomas Mann — a name that we had in the dictionary.

All the steps during the analysis of the questions and their transformation into the DIOGENE format are implemented in the CLaRK system, which is shortly described in the next section.

6 CLaRK System

In this section we describe the basic technologies of the CLaRK System[1] ([Simov et. al. 2001]). CLaRK is an XML-based software system for corpora development. It incorporates several technologies: *XML technology*; *Unicode*; *Regular Grammars*; and *Constraints over XML Documents*.

XML Technology

The XML technology is at the heart of the CLaRK System. It is implemented as a set of utilities for data structuring, manipulation and management. We have chosen the XML technology because of its popularity, its ease of understanding and its already wide use in description of linguistic information. In addition to the XML language [XML 2000] processor itself, we have implemented an XPath language [XPath 1999] engine for navigation in documents and an XSLT engine [XSLT 1999] for transformation of XML documents. We started with basic facilities for creation, editing, storing and querying XML documents and developed further this inventory towards a powerful system for processing not only single XML documents but an integrated set of documents and constraints over them. The main goal of this development is to allow the user to add the desirable semantics to the XML documents. The XPath language is used extensively to direct the processing of the document pointing where to apply a certain tool. It is also used to check whether some conditions are present in a set of documents.

Tokenization

The CLaRK System supports a user-defined hierarchy of tokenizers. At the very basic level the user can define a tokenizer in terms of a set of token types. In this basic tokenizer each token type is defined by a set of UNICODE symbols. Above this basic level tokenizers the user can define other tokenizers for which the token types are defined as regular expressions over the tokens of some other tokenizer, the so called parent tokenizer. For each tokenizer an alphabetical order over the token types is defined. This order is used for operations like the comparison between two tokens, sorting and similar.

Regular Grammars

The regular grammars in CLaRK System [Simov, Kouylekov and Simov 2002] work over token and element values generated from the content of an XML document and they incorporate their results back in the document as XML mark-up. The tokens are determined by

[1]For the latest version of the system see `http://www.bultreebank.org/clark/index.html`.

the corresponding tokenizer. The element values are defined with the help of XPath expressions, which determine the important information for each element. In the grammars, the token and element values are described by token and element descriptions. These descriptions could contain wildcard symbols and variables. The variables are shared among the token descriptions within a regular expression and can be used for the treatment of phenomena like agreement. The grammars are applied in cascaded manner. The evaluation of the regular expressions, which define the rules, can be guided by the user. We allow the following strategies for evaluation: 'longest match', 'shortest match' and several backtracking strategies.

Constraints over XML Documents

The constraints that we have implemented in the CLaRK System are generally based on the XPath language (see [Simov, Simov and Kouylekov 2003]). We use XPath expressions to determine some data within one or several XML documents and thus we evaluate some predicates over the data. Generally, there are two modes of using a constraint. In the first mode the constraint is used for validity check, similar to the validity check, which is based on a DTD or an XML schema. In the second mode, the constraint is used to support the change of the document to satisfy the constraint. The constraints in the CLaRK System are defined in the following way: `(Selector, Condition, Event, Action)`, where the selector defines to which node(s) in the document the constraint is applicable; the condition defines the state of the document when the constraint is applied. The condition is stated as an XPath expression, which is evaluated with respect to each node, selected by the selector. If the result from the evaluation is improved, then the constraint is applied; the event defines when this constraint is checked for application. Such events can be: selection of a menu item, pressing of key shortcut, an editing command; the action defines the way of the actual constraint application.

Cascaded Processing

The central idea behind the CLaRK System is that every XML document can be seen as a "blackboard" on which different tools write some information, reorder it or delete it. The user can arrange the applications of the different tools to achieve the required processing. This possibility is called **cascaded processing**. For more on application construction abilities of CLaRK System see [Simov, Simov and Osenova 2004].

7 Results and outlook

Here we report on the result from the Bulgarian–English QA track. From all 200 questions the correct answers were extracted for 26 questions. 168 answers were wrong, 5 inexact and 1 unsupported. The distribution of the correct answers among the question categories is as follows: 5 definition questions: 2 for organizations and 3 for persons; 21 factoid questions: 5 for locations, 2 for manner, 1 for measure, 2 for objects, 1 for organizations, 2 for other categories, 4 for persons, and 4 for time. The main problem that caused the wrong answer extraction was the degree of the ambiguity in the translation from Bulgarian to English. Interestingly, the percentage of the ambiguities for nouns has bigger impact on the results that the ambiguity of verbs. Another problem is that our semantic dictionary does not have a mapping to the English WordNet synsets which is a crucial information for DIOGENE System for answer extraction.

Our plans for future work are in two directions. First, we plan to implement a complete question answering system for Bulgarian. With respect to the Bulgarian–English task we envisage: to extend the dictionaries, to map our semantic dictionary (at least the top part) to the WordNet synsets and to implement an efficient translation disambiguation module.

References

[Balabanova and Ivanova, 2002] Elisaveta Balabanova and Krassimira Ivanova. 2002. *Creating a machine-readable version of Bulgarian valence dictionary: (A case study of CLaRK system application).* In: *Proc. of The First Workshop on Treebanks and Linguistic Theories.* Sozopol, Bulgaria.

[Ivanova and Dojkoff 2002] Krassimira Ivanova and Dimitar Doikoff. 2002. *Cascaded Regular Grammars and Constraints over Morphologically Annotated Data for Ambiguity Resolution.* In: *Proc. of The Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

[Magnini et al 2002] Magnini, B., Negri, M., Prevete, R., Tanev, H. 2002. *Comparing Statistical and Content-Based Techniques for Answer Validation on the Web*. Proceedings of the VIII Convegno AI*IA, Siena, Italy.

[Negri et al 2002] Negri M., Tanev H., and Magnini B. 2003. *Bridging Languages for Question Answering: DIOGENE at CLEF-2003*. Proceedings of CLEF-2003, Trondheim, Norway.

[Osenova 2002] Petya Osenova. 2002. *Bulgarian Nominal Chunks and Mapping Strategies for Deeper Syntactic Analyses.* In: *Proc. of The Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

[Osenova and Kolkovska 2002] Petya Osenova and Sia Kolkovska. 2002. *Combining the named-entity recognition task and NP chunking strategy for robust pre-processing.* In: *Proc. of The Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

[Osenova and Simov 2002] Petya Osenova and Kiril Simov. 2002. *Learning a token classification from a large corpus. (A case study in abbreviations).* In: *Proc. of the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics*, Trento, Italy.

[Pianta et al 2002] Pianta, E., Bentivogli, L., Girardi, C. 2002. *MULTIWORDNET: Developing an Aligned Multilingual Database.* Proceedings of the 1st International Global WordNet Conference, Mysore, India.

[Popov, Simov and Vidinska 1998] Dimitar Popov, Kiril Simov and Svetlomira Vidinska. 1998. *A Dictionary of Writing, Pronunciation and Punctuation of Bulgarian Language*. Atlantis LK, Sofia, Bulgaria.

[Simov et. al. 2001] Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, Atanas Kiryakov. 2001. *CLaRK - an XML-based System for Corpora Development.* In: Proc. of the Corpus Linguistics 2001 Conference. pp 558–560.

[Simov and Osenova, 2001] Kiril Simov and Petya Osenova. 2001. *A Hybrid System for MorphoSyntactic Disambiguation in Bulgarian.* In: Proc. of the RANLP 2001, Tzigov chark, Bulgaria.

[Simov, Popova and Osenova 2002] Kiril Simov, Gergana Popova, Petya Osenova. 2002. *HPSG-based syntactic treebank of Bulgarian (BulTreeBank).* In: *"A Rainbow of Corpora: Corpus Linguistics and the Languages of the World",* edited by Andrew Wilson, Paul Rayson, and Tony McEnery; Lincom-Europa, Munich, pp. 135-142.

[Simov, Kouylekov and Simov 2002] Kiril Simov, Milen Kouylekov, Alexander Simov. 2002. *Cascaded Regular Grammars over XML Documents.* In: Proc. of the 2nd Workshop on NLP and XML (NLPXML-2002), Taipei, Taiwan.

[Simov, Simov and Kouylekov 2003] Kiril Simov, Alexander Simov, Milen Kouylekov. 2003. *Constraints for Corpora Development and Validation.* In: Proc. of the Corpus Linguistics 2003 Conference, pages: 698–705.

[Simov, Simov and Osenova 2004] Kiril Simov, Alexander Simov, Petya Osenova. 2004. *An XML Architecture for Shallow and Deep Processing.* In: Proc. of the ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP. Nancy, France. pages: 51–60.

[Slavcheva 2002] Milena Slavcheva. 2002. *Segmentation Layers in the Group of the Predicate: a Case Study of Bulgarian within the BulTreeBank Framework.* In: *Proc. of The Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

[XML 2000] XML. 2000. *Extensible Markup Language (XML) 1.0 (Second Edition).* W3C Recommendation. http://www.w3.org/TR/REC-xml

[XPath 1999] XPath. 1999. *XML Path Language (XPath) version 1.0.* W3C Recommendation. http://www.w3.org/TR/xpath

[XSLT 1999] XSLT. 1999. *XSL Transformations (XSLT). version 1.0.* W3C Recommendation. http://www.w3.org/TR/xslt