Learning a token classification from a large corpus (A case study in abbreviations)[*]

Petya N. Osenova[†] and Kiril Iv. Simov
BulTreeBank Project
Linguistic Modelling Laboratory, Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A, 1113 Sofia, Bulgaria
petyaosenova@hotmail.com, kivs@bgcict.acad.bg

May 15, 2002

Abstract

The paper presents a general learning procedure of interrelated detection, classification and linguistic interpretation of the meaningful tokens within a large, gradually annotated corpus. Our approach combines the existing ideas on global and local ranking. The proposed mechanism is applied on abbreviations being part of the general task.

1 Introduction

The work, reported in this paper is on the poor-knowledge based treatment of productive language elements like abbreviations, proper names and new words in a large corpus. Our goal is to propose an algorithm, which learns the categories of tokens using their appearance and distribution within the whole text archive. In order to achieve a higher precision, we first classified all the tokens into several classes, based on their orthographical occurrences. Then, for each token appearance we chose the class with the highest frequency evaluation relying also on additional sources of information as the morphological dictionary, positions in the text and others.

The above task is being performed in conjunction with an ongoing work within the BulTreeBank project [Simov, Popova and Osenova 2001] and [Simov et al. 2002]. Its main goal is the creation of a high quality syntactic treebank of Bulgarian which is HPSG oriented. In the course of reaching this main goal we have to produce additional language resources and tools which lack for Bulgarian. These expected results of the project can be summarized as follows:

- A set of Bulgarian sentences marked-up with detailed syntactic information. These sentences are mainly extracted from authentic Bulgarian texts. They are chosen with regards to two criteria. First, they cover the variety of syntactic structures in Bulgarian. Second, they show the statistical distribution of these phenomena in real texts.

[*]The work reported here is done within the BulTreeBank project. The project is funded by the Volkswagen Stiftung, Federal Republic of Germany under the Programme *"Cooperation with Natural and Engineering Scientists in Central and Eastern Europe"* contract I/76 887.

[†]Also at the Bulgarian Language Division, Faculty of Slavonic languages, St. Kl. Ohridsky University, Sofia, Bulgaria

- Inside the TreeBank a core set of sentences are designated. The purpose of this core set of sentences is to serve as a test-suite for software application processing texts in Bulgarian on the level of syntactic descriptions.
- Reliable partial grammar for automatic parsing of phrases in Bulgarian. This grammar is extensively tested and used during the creation of the TreeBank. It will be used as a module separate from the TreeBank in tasks which require only partial parsing of natural language texts such as Information Retrieval, Information Extraction, Data Mining from Texts and etc.
- A linguistically interpreted text archive representative for the contemporary Bulgarian language. All linguistically important elements on lexical level are marked-up with relevant information. This includes not only the actual lexemes, but also numbers, special signs and others.
- Software modules for compiling, manipulating and exploring the data in the TreeBank. This software supports both: the creation of the TreeBank, and its use for different purposes such as automatic extraction of grammars for Bulgarian.

This paper reports one of the steps towards the creation of a linguistically interpreted text archive. The compilation of a text archive is connected with the appropriate recognition of all meaningful elements, i.e. not only common word tokens, but also named entities (person names, organization names, dates, currency expressions), abbreviations, punctuation. The efficient handling of these pre-processing steps proved out to be of great importance for the consistency of next parsing stages and, at the same time, for data mining. Thus it facilitates earlier direct use of the text archive.

In our approach we propose assigning different ranks to the tokens with respect to their membership in one or more of predetermined category classes. In case of semantically ambiguous tokens we envisage local-based disambiguation of the ranked categories.

Although we are aware of many comprehensive and robust methods for assigning the right category to the unknown tokens in the text, we aim at a new method for a rapid bootstrapping with respect to the tokens, that occur in the documents within a large, gradually annotated corpus. The problems with the other approaches are of two kinds:

- some of them require an already marked-up resource in order to apply a learning procedure;
- some are only document or local-based approaches. Thus they are domain sensitive and miss generalizations (which are crucial when relying on word graphical features).

Obviously the first kind is not applicable at this stage for our purposes. The second kind of approaches, although very powerful, depends on the specific properties of the document and therefore relies on different decisions for the same item in different documents. In addition, the possible typographical errors in the documents become a serious 'noise' for the mentioned strategies.

The structure of the paper is as follows: in the next section we present our motivation and discuss the variety of the token categories. Section 3 concentrates on our learning method of assigning all the appropriate token categories. Next section describes on the linguistic interpretation of the abbreviations. The last section presents the conclusions and outlook.

2 Motivation

As it was mentioned in the Introduction, our goal is to create a linguistically interpreted text archive of Bulgarian. The target size for the text archive is 100 million words, which is considered to be a standard size for a national corpus and we have processed up to now about 70 million words. In order to use the text archive in an efficient and linguistically motivated way, we envisage annotation of the texts with the relevant morphosyntactic information for wordforms and appropriate linguistic information for other kinds of expressions in the text, such as numerical expressions, special signs ($, % and others), foreign words and others. Sometimes these peculiar expressions bear lexemic information similarly to the wordforms, but sometimes they constitute phrases and need a special treatment. Additionally, we developed several partial grammars for processing the rest of the tokens in the texts. These grammars are as follows:

- sentence boundary determination;
- numerical expressions and special symbols;
- personal, organization and other names;
- foreign words.

These grammars are used after the application of the morphological analyser in order to mark-up the rest of the text tokens and also to mark-up the sentence boundaries.

Like many other NLP systems, our tagger relies on an extended dictionary - 100 000 lexical items. For clarity, we would like to state that by tagger we mean a tool, which supplies the recognized words with all possible morphological analyses. Then we rely on our connectionist-based disambiguating tool [Simov and Osenova 2001]. But even large lexicons cannot cover all systematic and productive word derivations (deverbal nouns, diminutives etc.), let alone non-closed classes like named entities and abbreviations. Hence, this information gap requires some proper handling at lower or higher stages in text processing procedures [Hajicova, Panevova and Sgall 2000]. This approach gives a very good results when processing clean texts, but it has some shortcomings in real text processing. The reasons for such shortcoming are well known: some common words are used in the text as proper names or abbreviations, the productivity of some language elements like proper names, abbreviations and others. Thus in our morphologically tagged 70-mln.-size corpus the different Cyrillic tokens are 660 000, out of which 340 000 remained untagged (not found in the morphological dictionary), because they were not recognized by the dictionary. The 340 000 tokens are about 8% from the size of text archive and together with non-Cyrillic tokens present a problem for the full linguistic interpretation of the text archive on this very basic level of processing. Therefore, it is crucial to avoid the possible tag ‘unknown’ and ‘wrong’ tags in order to minimize human labor in cross-validation techniques and to facilitate the work in the next stages. Very often the membership of common words in a lexicon is misleading because the actual use of the word can be as a proper name or an abbreviation. Thus our task is not just to deal with the unknown tokens in the texts, but also to determine the type of their particular usages. It is a well known fact that each common word can be used as a proper name in some context.

We consider each token in the text classified according to the following classifications:

- **Symbol-based classification.** The tokens are defined by their immanent graphical characteristics into several groups: Latin lowercased, Latin first capital, Latin all capitals, Latin mixed, Cyrillic lowercased, Cyrillic first capital, Cyrillic all capitals, Cyrillic mixed, symbols, spaces, punctuation and others. This classification is used for determining the potential meaningful elements in the text.

- **General token classification.** The tokens fall into several categories: common word, proper name, abbreviation, symbols, punctuation, error. This classification supports the decision which source of information for the token to be preferred when the grammatical and semantic properties of the token are determined. Here two factors are important: a global classification of the typical usages of the token and the changes of the typical usages in a local context.

- **Grammatical and semantic classification.** The tokens are presented in several lexicons, in which their grammatical and semantic features are listed. The information in the lexicons is attached to the tokens in the text and it is processed further for deeper analyses.

The main goal of our work is to learn the global classification of the tokens with respect to the general token classification. We plan to use this global classification in two ways: (1) as an initial classification of the tokens in the texts before consulting the dictionary, and (2) to process linguistically the tokens from the different classes and especially those, which have more impact over the whole set of texts. The second point is important for our language because there are no available lexicons of proper names and abbreviations for Bulgarian.

In the next section we present our approach towards learning general token categories.

3 Learning general token categories

There are man factors that can be taken into account in the global classification of the tokens, but in our work we chose their orthographical features. Thus we relied on the following token classes:

- Common words: typical - lowercased and first capital letter in sentence-initial position; non-typical - all caps;
- Proper names: typical - first capital letter; non-typical - all caps; wrong - lowercased
- Abbreviations: typical - all caps, mixed, lowercased (with period, hyphen or a single letter)

Note that some tokens can belong to more than one group according to their graphical properties. Another problem with a large set of texts is spelling errors which could cause misclassification. Thus our classification is not boolean but gradual-ranking of tokens with respect to each of the above categories. Our initial procedure included the following steps:

- First we used some graphical criteria for assigning potential categories to the unknown tokens.
- Then we used statistical methods to make a distinction within each category between the most frequent tokens of this category and tokens not in the category or rare tokens.

Graphical criterion - it takes into account the graphical specificity of the tokens. For each category a list of tokens potentially belonging to it was constructed. Some of the well known problems with the graphical criterion are:

- there are common words written in capital letters (especially headlines in newspapers) as a marker of emphasis or stylistic peculiarity.
- there are abbreviations which are written in a wrong way with small letters only. Similarly for proper names.

Conclusion: The graphical criterion is not sufficient for an adequate classification.

Statistical criterion

For each category, in order to get the maximal number of right predictions for candidate tokens, every candidate-token was ranked as follows:

TokPar

The number of appearances of the correctly written for the category form divided by the number of all appearances of the token.

DocPar

The number of the documents in which the correctly written for the category form was found if this number is less that 50, otherwise this maximal value is 50.[1]

Rank = TokPar * DocPar

If we had taken into account only the first parameter, we would have lost the real scope of distribution. The first parameter does not make difference between an item with one or two occurrences and an item with a hundred or more occurrences. Hence we decided to use also the impact of the token over the whole corpus. This is done by DocPar parameter. Additionally, we use a normalization via imposing an upper bound over the number of documents in which the token appears. This upper bound is 50. Thus the tokens that are rare or do not belong to the category receive a very small rank.

This method of token categorization tolerates the tokens with greater impact over the whole corpus and does not say much about the tokens appearing in small number of documents or in one document only, leaving their processing to local-based approaches.

The result of such a ranking is very useful for the other task mentioned above – the linguistic treatment of the unknown tokens. In the rest of the paper we describe the treatment of the abbreviations in our text archive.

4 Extraction of abbreviations

We consider the treatment of the abbreviations an important step in the pre-processing stage of the electronic archive. Abbreviations are semi-lexical linguistic items. They belong to the sub-paragraph level. They have a special typography and are language-specific. Together with proper names, numerical expressions and foreign tokens, they form a separate linguistic group with its own subgrammar and have to be recognized and processed before handling deep syntactic analyses. It is a necessary step due to two main reasons:

- the human work at further stages will be considerably minimized;
- it is a well-known fact that shallow parsed texts can serve as material for information extraction or retrieval.

One additional property of abbreviations is that they are a very productive class of linguistic items due to their relatedness to new organizations, products and other names. This is why we need a good procedure for identifying abbreviations in the text archive and for their automatic treatment. In the following subsections we apply the approach, described above, to abbreviation detection task in the text archive.

[1] This parameter is similar to idf parameter of $tf.idf$ techniques from Information Retrieval. In future work we will make also evaluation with respect to this parameter.

4.1 Preliminary typology of abbreviations

Abbreviations form a very heterogeneous group with its orthographic, grammatical and semantic peculiarities. For this reason we propose a typology model, which is relevant to their behaviour in written electronic texts. Here it is:

Abbreviations						
Lexical		With possibility for lexicalization			Graphical	
Initial		Non-lexicalized	Partly lexicalized	Completely lexicalized		
Only in capital letters	Mixed				with full stop	with hyphen
TZUM, VUZ	MVnR,BiBiSi	BRIB	AD-to	vec	ul.	h-l

Some explanations are in order. First of all, the type Lexical and the type With possibility for lexicalization can have common elements, because this is a dynamic lexical process. The term lexical is used to contrast the term 'graphical'. The definition with possibility for lexicalization indicates the extent to which the abbreviations resemble words. The scale ranges from non-lexicalized (pure acronyms) to partly lexicalized (acronyms with inflections, which make them be close to words) and completely lexicalized items (acronyms, written in lowercase and equal to words). There are, of course tokens, which can occur either in capital letters and therefore belong to pure acronyms, or in small letters - and in this case are common words.

We excluded some of the abbreviation types for a couple of reasons:

- the so called group (motel) and morphological (cola) abbreviations are part of the lexicon and are treated as words.
- some types of abbreviations are connected with name-entity recognition and foreign token recognition and they are usually treated as names, not as abbreviations.

4.2 Extraction criteria

At this stage of our work we do not aim at resolving all the abbreviation cases. We have dealt only with the most frequent ones in the archive as a preliminary step to their exhaustive treatment. In accordance with the general ideas, discussed in the previous sections, three types of criteria have been applied:

Graphical criterion - it takes into account the graphical specificity of the abbreviations and these specificities are used as heuristics.

1. written only in capital letters
2. written only in small ones with the relevant punctuation
   - one or two letters and a period
   - two or three letters with an internal hyphen
   - strings without vowels
3. mixed

Problems when relying only on the general word features:

- there are common words written in capital letters (especially headlines in newspapers) as a marker of emphasis or stylistic peculiarity.
- there are abbreviations which are written in a wrong way.
- there are some functional words that often stand in the sentence-initial position and wrongly could be considered to be abbreviations
- one and the same token could be homonymous and therefore used as an abbreviation and as a common word

In order to get the maximal number of right predictions for candidate-abbreviations, every abbreviated form was ranked according to the criteria in Section 3 as follows:

Ranking and Evaluation

Using the general word features, we extracted from the text archive about 37 000 candidate abbreviations. In order to evaluate the ranking procedure, we decided to do the following: First, according to the graphical criterion the possible candidate abbreviations have a very high recall - practically 100%, because all real abbreviations obey some of the imposed graphical criteria. On the other hand, if we consider all the 37 000 candidates, the precision is very low, because of noises, such as common words, written in all capitals or names. Thus we decided to take as evaluation set the abbreviations with the highest rank. For the abbreviations with ranks higher or equal to the threshold value 25 (557 candidates) only 7 candidates are non-abbreviations. Therefore for these 557 abbreviations the precision is very high - 98.7%.

The next step was the evaluation of the candidates with rank higher or equal to 2 (4814 candidates). We can consider the abbreviations with rank under 2 as very specific and therefore present in a small number of documents (usually one or two) or they belong to other category of linguistic elements like proper names or common words.

The results are as follows: from the highest rank down to rank 10 we achieved 80% correctness. Thus we decided to take into account the typology of the error sources within the abbreviation list and to apply a competitive looking-up strategy within the other types of groups, reported in Section 2. In this way we reuse the pre-compiled lists of tokens as knowledge resources. Then, comparing all the ranks of the candidate-abbreviation token, the classifier can assign the right category reliably enough.

We do not cross-validate the tokens in the local contexts first, but rather orthography-based extraction combinations and general categories are relied upon. Hence, by using error filters and classifier probability strategy, the reported result gets better.

Error filters

- the most popular capitalized newspaper rubrics, connected with advertisement and zodiac, are excluded.
- family names in Bulgarian are filtered by combining the capitalization factor with ending-guessing rules.
- the tokens that lie between the area of names and abbreviations are not considered errors when they are captured as abbreviations only (Biohim)

- morphological derivation guessing rules are implemented in order to recognize the most productive types (bulkontrola, monokosmos, mikrodvigateli).

Competitive classifier

The most interesting filter, however, is the comparison between the ranks, which were assigned to the same token with respect to its predicted membership in several classification types.

We used the same ranking mechanism as it was described above. We relied on the graphical characteristics of the tokens, which are considered typical for a certain category. Recall that for the abbreviations this was the case when the token appeared in ALL CAPITAL LETTERS; for proper names this is the appearance with FIRST capital letter and for ordinary words this is the appearance with all small letters.

For example, if in the text archive we have 'NATO' 1000 times, 'Nato' - 50 times, 'nato' - 3 times, then the token parameters will be:

TokPar = 1000/1053

for the abbreviation case,

TokPar = 50/1053

for the proper name case,

TokPar = 1000/1053

for the common word case.

Thus we have the same token, ranked with respect to the four lists, presented in Section 2. Then we compare the token ranks in the lists and determine the most probable type. We have not succeeded to evaluate the whole set of tokens (660 000). But there is evidence, that the proposed repairing mechanism works well: for the capitalized tokens that are at the beginning of the abbreviation list, but are not abbreviations, the type class predictions are very promising. For example, common words as 'PRODAVAM'('sell') with rank 41.8 and 'SKICNIK'('drawing book') with rank 25.54 are deleted from the abbreviation list. The first one is ruled out on the base of the newspaper rubrics criteria, mentioned above; the second - on the base of its presence in the group of common words with rank 36.2. And for 'NATO' the result is as follows: this token was classified as a proper name and as a common word with ranks under 1. Therefore, it is clearly an abbreviation.

Nevertheless, there appears another problem. Up to now we relied only on global corpus ranking. But when applied to every document, some errors could come up due to document-specific reasons. That is why, another parameter, which is locally based, is needed in order to compensate these possible errors. Luckily, we can employ the Document-centered approach, developed and applied by [Mikheev 1999]. The idea behind this approach is the following: the disambiguation of a graphically classified token within a document is viewed by detecting the unambiguously used occurences of this token in the document. Additionally, we have the global ranking as well and we can estimate the relation between the two parameters, i.e. they can coincide, the global parameter can overwrite the local one and the local one can override the global one. This is left for future work.

Context criterion

There are more and more new abbreviations coming into the language. This fact allows for a lot of ambiguities. In order to minimize this ambiguity, we extracted some candidate-abbreviations together with their extensions (usually in brackets). Applying phoneme-graphemic criteria, we could introduce some rules as: extract an item which is followed by a *right bracket-text-left bracket* and which initial letters possibly coincide with the initial letters of the text string between the brackets. However, this method is far from unifying, because of the following reasons: (1) sometimes the first letters of an acronym do not coincide with the first letters of the extension, (2) there are a number of word order possibilities of presenting the acronym and its extension. The rule, proposed above, presents just one alternative.

There is work on finding abbreviations and their definitions via pattern-based rules (for details see – [Park and Byrd 2001], but in our approach this contextual method is just supplementary because of the restrictions, listed above. Nevertheless, being language and domain-specific, it supports the evidence for an adequate linguistic interpretation.

Most of the proposed approaches on the recognition of abbreviations stop at the level of tokenisation [Mikheev 1999]. More elaborate interpretation work is done again within the named-entities area [Mikheev, Moens and Grover 1999], [Buchholz and van den Bosch 2000], [Barg and Walther 1999]. But our aim is not only to produce a reliable type classifier for automatic token recognition. We also reuse the initially extracted lexicons and initially built frequency lists as well as the structural classification of the special tokens (proper names and abbreviations respectively) for constructing a model for an adequate linguistic interpretation. The linguistically interpreted tokens are a good source for linguistic research and a step towards a word sense disambiguation task. In the following sections we discuss the linguistic treatment of the abbreviations in particular.

5 Linguistic treatment of abbreviations – a model

In order to process the abbreviations linguistically in a consistent way, we decided to represent all the information about them within XML format. We proposed a DTD (see the structure of the DTD in the appendix) which comprises the definitions capturing all the relevant linguistic information. The linguistic information was divided into two parts: obligatory - for all abbreviations and possible - for some of the abbreviations. Thus it serves for two implementation purposes: 1. the information in it is further incorporated into regular-expression based grammars and then applied on real texts, 2. it supports the manual annotation of the texts.

5.1 Encoding the linguistic information shared by all abbreviations

Here is a list of the elements obligatory for each abbreviation description:

- head element presents the abbreviation itself
- every abbreviation has a generalized type: acronym or word
- every abbreviation has at least one extension
- every extension element consists of a phrase

5.2 Encoding the linguistic information shared by some types of abbreviations

Here is a list of information relevant only to some kinds of abbreviations:

- the non-lexicalized abbreviations were assigned grammatical information according to their syntactic head. Thus the element class was introduced.

- the partly lexicalized abbreviations were assigned not only grammatical information according to their syntactic head, but also the grammatical information according to their inflection. Thus the element flex was introduced.
- the abbreviations which were transliterated from other languages (mainly from English) usually have an additional head element, called headforeign (headf) which presents the original abbreviated form. Consequently they could have the element expanf which presents the original extension of the abbreviation.

In the DTD also a header is defined. In the header element one can find the list of grammatical characteristics. This list is used to support the consistency of the grammatical description of the abbreviations.

Having compiled lists of abbreviations with some structured linguistic information in them, we can explore it for further tasks. One application, for example, is identifying complex named-entities via matching the extensions of the acronyms in texts. Here we rely on the linguistic fact that the extended variants of the acronyms are named-entities.

6 Examples of linguistic treatment

In this section we list several examples from the abbreviations that we have already processed and which demonstrate the different types of descriptions. The examples are transliterated and the grammatical characteristics are presented in more readable way:

Type **ACRONYM** (It is a generalized term over the lexical abbreviations)

1. The usual case in which the abbreviation has an assigned type, extension and grammatical characteristics:

```
<abbr>
  <head>ACHP</head><acronym/>
    <expan>
      <phrase>Agenciya za chuzhdestranna pomost</phrase>
      <class>Noun:f:sg</class>
    </expan>
<abbr>
```

2. Ambiguity case in which one head has two possible extensions with the relevant grammatical information:

```
<abbr>
  <head>DP</head><acronym/>
    <expan>
      <phrase>Dyrzhavno predpriyatie </phrase>
      <class>Noun:n:sg</class>
    </expan>
    <expan>
      <phrase>Demokraticheska partiya </phrase>
      <class>Noun:f:sg</class>
```

```
    </expan>
</abbr>
```

3. Partly lexicalized case in which the abbreviation is assigned two grammatical characteristics, one according to its syntactic head and another, according to its inflectional variant. In this case both of them coincide.

```
<abbr>
  <head>ZUNK</head><acronym/>
    <expan>
     <phrase>Zakon za urezhdane na neobsluzhvanite krediti</phrase>
     <class>Noun:m:sg</class>
     <flex>ZUNK-a, ZUNK-yt, ZUNK-ove</flex>
    </expan>
</abbr>
```

4. Foreign case in which the abbreviations are additionally assigned their original form:

```
<abbr>
  <head>FBR</head><headf>FBI</headf><acronym/>
    <expan>
      <phrase>Federalno byuro za razsledvane</phrase>
      <class>Noun:n:sg</class>
    </expan>
<abbr>
```

Type **WORD** (It handles the graphical abbreviations):

1. The usual hyphen case:

```
<abbr>
   <head>g-ca</head>
   <word/>
   <expan><phrase>gospozhica</phrase></expan>
</abbr>
```

2. The usual full stop case:

```
<abbr>
   <head>gr.<head>
   <word/>
   <phrase><expan>grad</phrase></expan>
</abbr>
```

3. Ambiguity case in which one word has more abbreviated variants:

```
<abbr>
```

```
    <head>v.</head>
    <head>v-k</head>
    <word/>
    <expan>vestnik</expan>
</abbr>
```

4. Ambiguity case in which one abbreviation has more than one extension:

```
<abbr>
    <head>ez.</head>
    <word/>
    <expan>ezero</expan>
    <expan>ezik</expan>
</abbr>
```

7 Future work

In this paper we presented an approach for classification of tokens in a corpus with respect to their orthography-based membership in the three main groups of categories: abbreviations, proper names and common words. This classification is applicable for: automatic annotation of the corresponding tokens and for facilitating their linguistic treatment. We demonstrated our ideas on the abbreviations. We also reported some preliminary results on the utility of pre-solving the classification task on token level and then its incremental fine-graining via creating and reusing knowledge resources of different kinds. Our next step in this direction is dealing with category ambiguity.

References

[Barg and Walther 1999] Barg, P. and Walther, M. *Processing Unknown Words in HPSG*. In Computational Linguistics submitted 1999. pp. 25

[Buchholz and van den Bosch 2000] Buchholz, S. and A. van den Bosch. *Integrating seed names and n-grams for a named entity list and classifier*. In Proceedings of the Second International Conference on Language Resources and Evaluation, pp. 1215-1221, Athens, Greece

[Cucerzan and Yarowsky 1999] Cucerzan, S. and Yarowsky, D. *Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence*. In Proceedings, Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora, pp. 90-99, 1999.

[Grefenstette and Tapanainen 1994] Grefenstette, G. and Tapanainen, P. *What is a word, What is a sentence? Problems of Tokenization.* 3rd Conference on Computational Lexicography and Text research. COMPLEX'94, Budapest.

[Hajicova, Panevova and Sgall 2000] Hajicova, E., Panevova, J., Sgall P., *A manual for tectogrammatical tagging of the Prague Dependency Treebank.* Carles University, Prague. 2000.

[Krumova and Choroleeva 1982] Krumova L., Choroleeva M., *Sykrashtavaneto i sykrashteniiata v balgarskiia ezik.* Bulgarian Academy of Sciences, 1982.

[Mikheev 1997] Mikheev A. *Automatic Rule Induction for Unknown Word Guessing* In Computational Linguistics vol 23(3), ACL 1997. pp. 405-423

[Mikheev 1999] Mikheev A. *Periods, Capitalized Words, etc.* In Computational Linguistics,submitted 1999, pp. 25

[Mikheev, Moens and Grover 1999] Mikheev A., Moens M., and Grover C. *Named Entity recognition without gazetteers* In EACL'99, Bergen, Norway ACL June 1999. pp. 1-8

[Park and Byrd 2001] Park Y. and Byrd R. *Hybrid text mining for finding abbreviations and their definitions.*In: Proc. of the EMNLP-2001. 2001.

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

[Simov and Osenova 2001] Simov K, Osenova P., *A Hybrid System for MorphoSyntactic Disambiguation in Bulgarian.* In: Proc. of the RANLP 2001 Conference, Tzigov chark, Bulgaria, 5-7 Sept., p. 288-290, 2001.

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

[Simov, Popova and Osenova 2001] Simov K, Popova G, Osenova P, *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 (to appear).

[Simov et al. 2002] K.Simov, P.Osenova, M.Slavcheva, S.Kolkovska, E.Balabanova, D.Doikoff, K.Ivanova, A.Simov, M.Kouylekov. 2002. *Building a Linguistically Interpreted Corpus of Bulgarian: the BulTreeBank.* In: Proceedings from the LREC conference, Canary Islands, Spain.

8 APPENDIX

DTD for representation of the abbreviations in XML format.

```
<! DOCTYPE abbreviations [
<! ELEMENT abbreviations (header, abbr+)>
<! ELEMENT header (#PCDATA)>  <!-- List of grammatical classes -->
<! ELEMENT abbr (head+,headf?,(word | acronym),expan+,expanf?)>
<! ELEMENT head (#PCDATA)>
<! ELEMENT headf (#PCDATA)>
<! ELEMENT word EMPTY>
<! ELEMENT acronym EMPTY>
<! ELEMENT expan (phrase,class?,flex?)>
<! ELEMENT expanf (phrase)>
<! ELEMENT phrase (#PCDATA)>
<! ELEMENT class (#PCDATA)>
<! ELEMENT flex (#PCDATA)>
]>
```