Automatic Identification and Linguistic Description of the Abbreviations in the BulTreeBank Electronic Text Archive*

Petya N. Osenova† and Kiril Iv. Simov
BulTreeBank Project
Linguistic Modelling Laboratory
Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A
1113 Sofia, Bulgaria
osenova@slav.uni-sofia.bg, kivs@bgcict.acad.bg

March 15, 2002

1 Introduction

This paper presents an ongoing work within the BulTreeBank project on

1. creation of linguistically annotated corpus of Bulgarian texts, and
2. an HPSG-oriented syntactic treebank.

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 with 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.

The work, reported in this paper is on the treatment of the productive language elements like the abbreviations, the proper names and the unknown words, based on a knowledge-poor approach. Our goal is to propose an algorithm, which learns the categories of certain tokens on the base of 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 we chose the class with the highest frequency evaluation. Thus, for a given token we can have several classes as possible candidates for the correct graphical class. For instance, 'NATO' was found erroneously written also as 'Nato' and 'nato', and according to the orthographical criterion, all these tokens of the same word could be classified equally as an abbreviation, as a proper name or as an ordinary unknown word. In our approach we propose assigning of different ranks to the tokens with respect to their membership in one or more of these classes. Thus, at the stage of automatic annotation each of the above occurrences will be marked-up as an abbreviation only. Of course, in the case of ambiguous tokens this approach is not perfect. For this reason, in future we envisage to develop methods for repairing such cases.

Although we are aware of many really 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:

*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

- some of them require an already marked-up resource in order to apply a learning procedure;
- some are document or local-based approaches and thus are domain sensitive.

Obviously the first kind is not applicable at this stage for our purposes. The second kind of approaches, although very powerful, depend 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 serious 'noise' for the mentioned strategies. Our approach aims at a fast recognition of the tokens with substantial impact over the whole text archive, with the presumption that some repairing is possible at later stages of the corpus development.

The structure of the paper is as follows: in the next section we give some information about our tagger and our corpus. Section 3 concentrates on the recognition task of the abbreviations. In section 4 we present our criteria for identification of the abbreviations in the text, as well as the linguistic description of the extracted abbreviations. The last section presents our conclusions and outlook.

2 The tagger and the corpus

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. But even large lexicons cannot cover all systematic and productive word derivations (deverbal nouns, diminutives, feminines), 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 and Panevova and Sgall] There are different attempts to repair this inconsistency in NLP by building guessers [Mikheev 1997], writing grammars or using a learning component [Barg and Walther 1999], [Cucerzan and Yarowsky 1999]; relying on pure statistical methods, rule-based ones or hybrid systems; with or without gazetteers [Mikheev and Moens and Grover 1999], [Buchholz and van den Bosch 2000]; handling the problems separately [Buchholz and van den Bosch 2000] or in combination and interactively [Mikheev 1999].

In our morphologically tagged 70-mln.-size corpus the different Cyrillic tokens are 660 000, out of which 340 000 remained untagged, because they were not recognized by the dictionary. The 340 000 tokens are all about 8% of 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 minimize the possible tag 'unknown' in order to minimize human labor in cross-validation techniques and to facilitate the work in the next stages. First, we consider useful to estimate the impact of the different types of unrecognized by the lexicon words over all the words in the corpus. This step will help us to keep better statistics and predictability over the enlarging corpus. But this task needs a preliminary classification, which is discussed in the next section.

3 The abbreviation recognition task

We have information about the type of orthography of every token occurrence. The basic types can be classified as follows:

- With capital letters only: NATO, SAST
- With first capital letter only: Petya
- Mixed: MvNR, tovaEstranno
- Lower-case only: boza

The first type contains potential abbreviations, in the second are potential proper names, the third represents usually a special group of abbreviations or typographical errors of two or more words written jointly, the last one contains the common words.

Needless to say, some tokens are written in several ways: With capital letters only and With first capital letter only: NATO, Nato, Petya, PETYA. We concentrate first on processing of abbreviation and then we extended our work to proper names and unknown common words. Thus the following text is mainly about abbreviations. The abbreviation identification task is not as much discussed as, for example, the named-entity recognition problem. [Mikheev 1999, p. 21] and [Grefenstette and Tapanainen 1994]. There are several possibilities for approaching this problem: without lexicon, just with corpus filter, lexicon without abbreviations or lexicon with some abbreviations.

As it was mentioned above, we applied a corpus-based rather than document-based approach for this task. In our opinion this task includes not only identificaton and classification of the abbreviations in the texts, but marking them with linguistic information as well at further stages.

Our initial procedure included the following steps:

- First we used some heuristics, based on graphical criteria. They were represented as regular grammars over tokens and token types in order to recognize the potential abbreviations.
- Then we used statistical methods to make a distinction between the most frequent actual abbreviations and non- or rare abbreviations. The linguistic description of the abbreviations was done by using the CLaRK system [Simov et al. 2001] facilities for creation of language resources.

These steps helped us to create a list of the most frequently used within the corpus abbreviations and then to re-use them for better recognition.

It is assumed that abbreviations are domain-dependent elements, i.e. some types are typical for some texts. In this respect we rely on the size and variety of our corpus.

4 Automatic Identification and linguistic interpretation of abbreviations

We consider the treatment of the abbreviations as an important step in the pre-processing stage of the electronic archive. Abbreviations are considered to be 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 organisations, products and other names. This is why we need a good procedure of identification of abbreviations in the text archive and possibly their automatic treatment. In the following subsections we present a statistically based approach to abbreviation identification in the text archive.

4.1 Classification 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 is our type of classification:

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. 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 range from non-lexicalized to partly lexicalized and completely lexicalized items.

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

- the so called group (motel=motel) and morphological (kola=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 need a multiple approach.

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. Three types of criteria were 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 internal hyphen
   - strings without vowels
3. mixed

Problems:

- there are 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.

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

Statistical criterion

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

TokPar

The number of appearances of the correctly written abbreviated form divided by the number of all appearances of the abbreviated form

DocPar

The number of the documents in which the correctly written abbreviated form was found if this number is less that 50, otherwise this maximal value is 50.

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 introduce a normalizing parameter, which value is 50. The new parameter helped us to distinguish between more frequent and more specific abbreviations and to solve the problem with the frequent initial position of non-abbreviations, because in such cases the rank value is much lower.

Evaluation

Having used the described criteria and especially the ranking of the abbreviations we extracted from the text archive about 37 000 candidate abbreviations. In order to evaluate the ranking procedure we decided to apply the following approach. First, after applying only the graphical criterion to possible candidate abbreviations the recall is very high - practically 100%, because all real abbreviations have to obey some of the imposed graphical criteria. On the other hand, if we consider all 37 000 candidates, the precision will be very low. Thus we decided to take as evaluation set the abbreviations with the highest rank. First for the abbreviations with ranks higher or equal to 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 for 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 documents) or they belong to other category of linguistic elements like proper names or ordinary 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 different types of groups, reported in Section 3. Then, comparing the different ranks of the candidate-abbreviation token, the classifier can assign the right category reliably enough.

The new idea in the proposed approach is that we do not cross-validate the tokens in the contexts, but rather on the orthography-based extraction combinations. 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 as 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).

Competition-based classifier

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

We used the same ranking mechanism as it was described above. The difference is that now other graphical characteristics of the tokens are counted as important. 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.

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

$$\mathbf{TokPar = 1000/1053}$$

for the abbreviation case,

$$\mathbf{TokPar = 50/1053}$$

for the proper name case,

$$\mathbf{TokPar = 1000/1053}$$

for the common word case.

Thus we have the same token, ranked in the four lists, presented in Section 3. Then we compared the token ranks in the lists and determined 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.

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.

Most of the proposed approaches on the recognition of abbreviations stop at the level of tokenisation. More elaborate interpretation work is done again within the named-entities area. We try to repair this gap. Our aim is not only to produce a reliable type classifier for automatic annotation. We also reuse the frequency list and the classification of abbreviations 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 rest of the paper we discuss the linguistic treatment of the abbreviations.

5 The 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 the abbreviations within XML format. We proposed a DTD which comprises the definitions that capture all relevant linguistic information. The linguistic information was divided into two parts: obligatory - for all abbreviations and possible - for some of the abbreviations.

Encoding the linguistic information shared by all abbreviations

Here is a list of the elements obligatory for each abbreviation description: (1) head element presents the abbreviation itself; (2) every abbreviation has a generalized type: acronym or word; (3) every abbreviation has at least one extension; (4) every extension element consists of a phrase.

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.

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:

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>Szhed</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>Ssred</class>
    </expan>
    <expan>
      <phrase>Demokraticheska partiya</phrase><class>Szhed</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.

```
<abbr>
  <head>ZUNK</head><acronym/>
    <expan>
      <phrase>Zakon za urezhdane na neobsluzhvanite krediti</phrase><class>Smed</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>Ssred</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 two kinds of abbreviation:

```
<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 present an approach for classification of tokens in a corpus with respect to their orthography-based membership in the three main groups of ABBREVIATIONS, Proper names, and common words. This classification is applicable for: automatic annotation of the corresponding tokens and for facilitating their linguistic treatment. We demonstrate how this is done in the case of linguistic treatment of abbreviations. We also reported some preliminary results on the utility of dividing the tokens into different classes for repairing of type-classification errors.

We envisage to prolong our work in several directions:

- Dealing with different kinds of class ambiguous tokens.
- Applying the same approach to sentences boundary problem.

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

[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 and 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

[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).