Bulgarian Nominal Chunks and Mapping Strategies for Deeper Syntactic Analyses[*]

Petya N. Osenova[†]
BulTreeBank Project
Linguistic Modelling Laboratory
Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A
1113 Sofia, Bulgaria
petya@bultreebank.org

Abstract

The paper focuses on the incremental integration of the chunking stage for robust deeper syntactic analyses. It first describes a base NP chunker for Bulgarian, which is hand-made, sensitive to lexical information and implemented as a cascaded regular expression grammar within the CLaRK system. The problems, related to the NP chunker are discussed from both points of view - linguistic and implemental. Then a mapping scheme between the output chunks and dependency relations is considered with respect to deeper parsing within HPSG.

1 Introduction

Building a detailed syntactic treebank faces various problems, some of which are connected with theoretical dependency (different viewing on the same linguistic phenomena due to different linguistic theories) and some of which are caused by the linguistic data (genuine ambiguities).

The syntactic annotation process within the project BulTreeBank relies on the development of two formal grammars: 1) a partial grammar for shallow parsing; 2) an HPSG-based general grammar [Simov, Popova and Osenova 2001] and [Simov et.al. 2002b]. The first one relies on several pre-processing components: sentence extraction, morphosyntactic tagging, named entity recognition, disambiguation and partial parsing. The latter evaluates the attachment possibilities over the partially parsed language input. Hence, the chunking module is expected to serve as a reliable template for deeper syntactic analyses.

The chunking stage was introduced in [Abney 1996] in order to ensure consistency at lower levels of processing. It includes the notion of constituent and the notion of head. Hence, it is supposed to guarantee a reliable base for further annotation - either monolevel, or multilevel.

[*] This work 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

The chunks are defined as 'the non-recursive core of an intra-clausal constituent to its head, but not including post-head dependents' [Abney 1996, p.1] and [Abney 1997, p.9]. When dealing with Bulgarian data, we keep the notion of non-recursivity, but at the same time cases with post-head dependents are not excluded. Abney's strategy relies on three heuristics: detecting 'islands of certainty', applying 'easy-first parsing' and 'preferring syntax to semantics'.

In Abney's version chunking suggests cascaded bottom-up strategy of syntactic annotation and relies on human rule-writing. However, the basic idea has developed and undergone some changes in the following directions: the levels under and above chunk level; the chunk types; adding some top-down filtering upon chunking level and establishing a connection with shallow parsing [Mueller 2002]; including an active-learning component when developing and NP chunkers [Brill and Ngai 1999], [Ngai and Yarowsky 2000] and [Daelemans et. al. 1999].

The nominal chunker for Bulgarian is hand-made and relies on a cascaded bottom-up strategy, because of the following reasons: some means for top-down filtering are still under parallel development (like the Valency frames of the most frequent verbs in Bulgarian), there is no enough annotated data yet for enabling the automatic extraction of the rules from a corpus. Therefore our chunker is consistent preferably with Abney's approach. Certain attempts have been made for pre-solving some problems on the sub-chunk level. The paper contribution, therefore, is viewed in three directions: 1. the specificity of the Bulgarian nominal groups, 2. encoding the rules within the CLaRK system and 3. mapping the chunk module to dependency relations in order to facilitate the HPSG analyses.

The paper concentrates on the stages of building a nominal chunker for Bulgarian, discusses some related problems and pre-processing steps, and views its integration within the treebank.

The structure of the paper is as follows: In the next section the nominals in Bulgarian are discussed. In section 3 the main chunk types are presented and the sub- and super-chunk levels are defined. Section 4 discusses the application of the chunk grammars within the CLaRK system. Section 5 focuses on the problems, which the NP chunker faces in the processing. In Section 6 some preliminary evaluation results are reported. Section 7 describes the process of mapping the NP chunker to dependency relations with respect to the HPSG theory. The last section outlines the conclusions and future work.

2 Bulgarian Nominals

In this section we discuss some specific features of Bulgarian nominals (nouns, pronouns, numerals and adjectives) that are relevant for both - the chunk level and deeper syntactic analyses.

2.1 A brief overview

Bulgarian language has no nominal declension system. Two remarks are in order here: 1. Only some of the pronouns are specified for case (nominative, accusative or dative), but it is not systematic and 2. The feminine and masculine nouns possess as a rule vocative forms.

Hence, all the elements in Bulgarian NPs must agree in number and gender only (except for some quantifiers that do not inflect, for ex. cardinal numerals). It is assumed that the elements within the NP usually have a rather fixed ordering, but in the real language multi-componential NPs show a lot of internal positional varieties. The quantifiers can have different scope realizations as it is shown in the examples below:

(1) moi dvama dobri prijateli
my two good friends
two good friends of mine

(2) dvama moi dobri prijateli
two my good friends
two good friends of mine

Note that in example (1) the possessive pronoun has a scope over the numeral, while in example (2) the numeral has a scope over the possessive pronoun.

Another peculiarity is the fact that the definite article is morphologically expressed, i.e. it is a part of the word. It serves as a phrasal affix within the nominal group. The only lexeme that historically bears the same definite information (with certain semantic differences, of course) is the demonstrative pronoun *tozi* ('this').

There exist possessive and reflexive-possessive clitics, which are shortened forms of these pronouns. As a rule they are placed after the first definite element in the NP:

(3) dobrata mi prijatelka
good-the-fem,sg my-clitic-1p,sg friend-fem,sg
my good friend

(4) tazi im osobenost
this-fem,sg their-clitic-3p,pl peculiarity-fem,sg
their peculiarity

2.2 NP Typologies

NPs can be classified with respect to different criteria such as the number of its internal elements (heaviness)[1], specific features (being names, numerical expressions etc), recursivity (recursive vs. non-recursive NPs), coordination (coordinated vs. non-coordinated NPs). Note that most of the mentioned criteria interleave.

2.2.1 Heaviness

We consider the notion of 'heaviness' quantitative. With respect to this characteristics Bulgarian NPs are divided into the following groups: *one-componential*, *two-componential* and *multi-componential.*

1. *One-componential NPs* can include words tagged as: a noun (common or proper), a vocative noun, a pronoun (personal - excluding the clitic forms; collective, demonstrative, interrogative, indefinite, negative and relative pronouns (for person or quantity only)).

[1] This criterion was introduced in [Aarts 1992]

2. *Two-componential NPs* can include the following types, in which the head noun is either pre-modified, or post-modified:

   (a) a noun, pre-modified by an adjective, a numeral, a pronoun, an adverb or a participle (*hubava zhena* 'pretty woman')

   (b) two nouns (*butilka vino* 'bottle wine')

   (c) a noun and a post-positioned possessive or reflexive clitic (*knigata mi* 'book-the my-clitic')

   (d) a vocative noun, pre-modified or post-modified by an adjective, a numeral, a pronoun (*Bozhe moj* 'God my')

   (e) the substantivized indefinite, interrogative or negative pronoun, post-modified by an adjective (*neshto hubavo* 'something nice')

   (f) the substantivized indefinite or negative pronoun, pre-modified by an adjective or a pronoun (*hubavo neshto* 'nice thing').

3. *Multi-componential NPs* are of two main kinds concerning the head noun:

   (a) premodified only - they include all the varieties of the modifiers' orderings: *visoka hubava zhena* 'tall pretty woman', *edna visoka hubava zhena* 'one tall pretty woman' etc.

   (b) both - pre- and post-modified. These patterns can include not only the internal for NPs recursivity: *hubavata dobra zhena* 'pretty-the nice woman'(i.e. a sequence of two or more adjectives), but external recursivity as well: *hubavata uchitelka po peene* 'pretty-the teacher in singing'(i.e. a sequence of two or more nouns).

2.2.2 Specificity

With respect to the stages of data processing the NPs are divided into three main kinds:

1. common NPs - their head is a common noun and they are processed by the base NP chunker.

2. named-entities NPs - their head is a proper name or a numerical expression of different complexity without any external modifiers. They are processed by the Named-entity module.

3. combined NPs - their head is a proper name with external modifiers included. They are processed by the base NP chunker. Note that for the chunker it is irrelevant whether the head is a simple or a complex one.

3 Chunking and Nominal Chunks

The chunker relies on a morpho-syntactically tagged corpus, which has been pre-processed on the level of the named entities recognition (names, numerals, special symbols, abbreviations)[2]. The fact, that we can rely not only on POS tags, but on disambiguated grammatical

[2] For more details on the pre-processing of special NPs see the following references [Osenova and Kolkovska 2002], [Osenova and Simov 2002] and [Ivanova and Dojkoff 2002].

information, is crucial for a free-order language like Bulgarian. The combination of the appropriate morpho-syntactic features (agreement in gender and number or the explication of the definiteness effect) has the advantage that it helps to rule out unwelcome and ungrammatical parses.

3.1 The Subchunk Level

Similarly to Abney's treatment of measure phrases below the chunk level [Abney 1997, p. 9], we rely on some phrases to be recognized before the real chunking. These are numerical expressions of different complexity, proper names and abbreviations. Thus, instead of having just compound NPs [Abney 1997, p. 11], we have NPs of different types - common, name, abbreviation, complex. All of them, irrespectively of their type, are assigned phrasal linguistic information that can be explored at further stages of processing.

At this stage we include the recognition of some NPs of NN type in Bulgarian. We cannot rely on the purely syntactic and positional longest match rule in this case, because subject and object are likely to stand next to each other. This problem is partly repairable within a valence-frame-based top-down filtering, which would say that if the verb is intransitive, the two nouns (or NPs) most probably constitute one NP; if it is transitive, and the nouns are three, most probably two of them constitute one NP. But aiming at higher consistency at this very level, we decided to make lists of:

1. the most frequent quantitative nouns, which rather take a noun complement, and
2. the most frequent nouns of foreign origin, which are often used as first elements in compound nouns [Avramova, Osenova in press].

Thus in both cases the strategy for the named-entity recognition is relied upon, namely these nouns identify the beginning of a possible NN and only then the pre-modifiers are detected. The advantage of this approach is that it is consistent with our HPSG-based analysis of such NPs[3]. In the first case, viewing the second element as a complement of the first one, it is better to capture the two nouns first and then the pre-modifying elements. For example:

(5) [edna [chasha voda]]
one-fem,sg glass-fem,sg water-fem,sg
one glass of water

instead of:

(6) [[edna chasha][voda]]
one-fem,sg glass-fem,sg water-fem,sg
one glass of water

The same strategy is appropriate for the second case, because the two nouns are considered a compound. But being recursive, all these cases are pre-solved to chunking, at least the most frequent types. The examples are similar to the above ones with one difference only - the leftmost premodifiers usually do not agree with the first noun, but with the second. For this reason the analysis in (8) is not appropriate.

[3] For more details see [Osenova to appear]

(7)	[edna	[biznes	sreshta]]
	one-fem,sg	business-m,sg	meeting-fem,sg
	a business meeting		

instead of:

(8)	*[*[edna	biznes]	[sreshta]]
	one-fem,sg	business-m,sg	meeting-fem,sg
	a business meeting		

3.2 The Nominal Chunk Level

The Bulgarian nominal chunker deals with the non-recursive and common nominal chunks[4]. It captures only some simple cases of coordinated modifiers within an NP, such as *hubava i dobra zhena* 'pretty and nice woman'. The chunker relies on the preference of accuracy to coverage. For this reason some problematic NP groups are excluded from the rule grammar set, thus generating the appearance of partially recognized NPs.

The Bulgarian NP chunk either extends from the beginning of the noun phrase to the head noun or from the head noun to the end of certain non-recursive dependents. Here two such cases are presented for clarity.

1. The possessive clitics, which always take the second post-head position in a 'noun-clitic' construction.

(9)	shapkata	mi
	shapka-the	my-clitic
	my hat	

2. There are some specific structures with the collective, negative and indefinite neuter pronouns, being heads, whose adjectival modifier is in the slot on the right.

(10)	neshto	specialno
	something-n,sg	special-n,sg
	something special	

A strategy is applied, which relies on the presence of clear indicators, pointing to the unambiguous beginning of an NP. For example, the adverb and the adjectivally used non-definite past and passive participles are not clear indicators of an NP beginning. The reason is that due to the relatively free word order adverbs and participles might be a part of a verbal chunk.

A non-one-componential NP can unambiguously begin with the following words: an adjective, a numeral, a pronoun or a definite participle in adjectival use (because in this case its adjectivity is guaranteed). The accusative and dative clitics are excluded, because they are considered to belong to the level of the verbal complex.

One-componential NPs are considered unambiguous when tagged as nouns; long forms of the personal pronouns; collective, demonstrative, interrogative, indefinite, negative and relative pronouns - for person or quantity only.

[4] It processes some complex NPs as well, but these NPs are discussed in more detail in another paper [Osenova and Kolkovska 2002]

3.3 The Superchunk Level

At some point it is necessary to consider the problem of complex, or recursive (maximal) NPs. It becomes crucial for languages with a relatively free word order. Delaying the attachment decisions for later steps of processing is sometimes misleading (recall the classical example with a PP, which modifies the noun within another PP. Similar problems are caused by coordinated NPs).

Unfortunately, Bulgarian cannot rely on the helpful restrictive mechanisms of the Topological fields (as German and Dutch do). For the moment only named-entity NPs, which are identified via the Named-entity grammar module are captured as recursive NPs.

4 The Implementation of the Nominal Chunk Grammar within the CLaRK system

There are two main approaches to the NP shallow annotation:

1. viewing parsing as a word tagging problem [Daelemans et. al. 1999]
2. using regular expressions on words and POS tags as well as the surrounding context [Abney 1996], and [Brill and Ngai 1999].

We relied on the second approach, because the first one presupposes the existence of a syntactically annotated corpus, which lacks for Bulgarian. Thus, a hand-crafted symbolic regular expression grammar was designed for the base NPs. It was constructed on manually disambiguated fiction texts and then tested on manually disambiguated newspaper texts from the BulTreeBank corpus.

4.1 The Cascaded Regular Expression Grammar Engine of the CLaRK system

The CLaRK system ideologically followed and developed further Abney's ideas on partial grammars, which were implemented in the Cass system [Abney 1997]. Let us compare the two software engines briefly: the input of both tools is tagged. CLaRK operates on XML-based documents, while Cass relies on the strategy 'one word per line'. CLaRK has its own tagger for Bulgarian texts, while Cass uses the mapping program tagfixes. The CLaRK output is conformant with XML, while Cass supports several graphical layouts. Both tools rely on cascaded FSA and include the longest match strategy (in CLaRK other strategies are included as well - shortest match, back-tracking etc.). CLaRK additionally suggests possibilities for using the prompts of the right and the left context, which is very useful when determining NP phrases with unclear/ambiguous starting indicators. For example, an NP, following a preposition, with an adverb as a vague first element, is recognized by CLaRK grammar, but not by Cass:

```
[R s] (with)
  [Adv mnogo ](very)
  [np1
    [Amsi gorest ](hot)
    [Ncmsi klimat ]](climate)
  [dot dot]
```

In the CLaRK system the definition of regular grammars[5] is implemented along the lines of [Abney 1996]. Thus a *cascaded regular grammar* is a sequence of regular grammars defined in such a way that the first grammar works over the input word and produces an output word, the second grammar works over the output word of the first grammar, produces a new output word and so on. Rules of the following kind are used:

```
C -> R
```

where `R` is a regular expression and `C` is a category of the words recognized by `R`.

An additional requirement suggested by [Abney 1996] is the so-called *longest match*, which is a way to choose one of the possible analyses for a grammar. The longest match strategy requires that the recognized sub-words from left to right have the longest length possible. Thus the segmentation of the input word starts from the left and tries to find the first longest sub-words that can be recognized by the grammar and so on to the end of the word.

When applied to XML documents, the regular grammars have the following specificities:

1. the text is segmented into meaningful non-overlapping tokens and they are treated as 'letters' of the grammars
2. when a text is considered an input word for a grammar, it is represented as a sequence of tokens.

The means for describing tokens, however, are enlarged with the so called *token descriptions* which correspond to the letter descriptions on regular expressions. In the token descriptions we use strings (sequences of characters), wildcard symbols `#` for zero or more symbols, `@` for zero or one symbol, and *token categories*. Each token description matches exactly one token in the input word.

The token descriptions are divided into two types - those that are interpreted directly as tokens and others that are interpreted as token types first and then as tokens belonging to these token types.

The first kind of token descriptions is represented as a *string* enclosed in double quotes. The string is interpreted as one token with respect to the current tokenizer. If the string does not contain a wildcard symbol then it represents exactly one token. If the string contains the wildcard symbol `#` then it denotes an infinite set of tokens depending on the symbols that are replaced by `#`.

Within the regular expressions the angle brackets are used in order to denote the boundaries of the element values. Inside the angle brackets we could write a regular expression of arbitrary complexity in round brackets. As letters in these regular expressions we use again token descriptions for the values of elements and the values of attributes. For tag descriptions we use strings which are neither enclosed in double quotes nor preceded by a dollar sign. We can use wildcard symbols in the tag name. Thus

`<p>` is matched with a tag `p`;

[5] The information here is strongly based on [Simov et. al. 2002a]

`<@>` is matched with all tags with length one.

`<#>` is matched with all tags.

It was decided that the category for each rule in the CLaRK System is a custom mark-up that substitutes the recognized word. Since in most cases we would also like to save the recognized word, we use the variable `\w` for the recognized word. For example the nominal groups are tagged as follows:

```
"mother"|"lazy boy" -><np>\w</np>
```

The mark-up defining the category can be as complicated as necessary. The variable `\w` can be repeated as many times as necessary (it can also be omitted). For instance, for "move" the rule could be:

```
<w aa="V;N">\w</w> -> "move"
```

Another extension of the regular grammars within the CLaRK System concerns the description of the left and the right context of a given subword recognized by the regular expression of a rule. Thus the rules in the grammar will have the format:

```
C -> LC : R : RC
```

where `C` is the category, `LC` is a regular expression describing the left context of the words recognizable by the rule, `R` is a regular expression describing the set of words recognizable by the rule, `RC` is a regular expression describing the right context of the words recognizable by the rule. The regular expressions `LC` and `RC` can be empty and then there are no constraints over the left and the right context.

4.2 The NP chunk grammar - description and application

The base NP chunker operates over the content of the **ta** element, which presents the true morphosyntactic tag of the token. Hence, this module relies on the tagger accuracy and fails in cases when the token remains unrecognized. Here is a small example from our morphologically processed corpus, which demonstrates the above considerations:

```
<w><ph>okonchatelno</ph><aa>Ansi;D</aa><ta>D</ta></w>
```

The word *okonchatelno* ‘definitely’ has the following encodings: the tag **ph** consists of the phonological word form, the tag **aa** suggests all possible analyses for the given word and the tag **ta** supplies the correct analysis of the word in the given context.

The NP chunk grammar is divided into two subsequently following submodules: **after-preposition-NPs** and **NPs**. The former is run first, because it identifies all standard two- and multi-componential NPs plus those with the problematic beginners. The latter module captures all the rest NPs, whose occurrence does not depend on the context. Let us discuss the two submodules in more detail here:

1. The context-bound submodule captures all non-one-componential NPs plus the NPs with the problematic beginners, such as adverbs and non-definite participles. Its structure is as follows:

```
LC=<"R">
R=regular expression rules
RC=empty
RM=<np type="common">\w</np>
```

The left context area (LC) is occupied by a preposition (”R”). The regular expressions area (R) consists of rules, which possibly start with adverbs or problematic participles. In fact, one basic rule is split into several rules with respect to the agreement of gender and number features. The right context area (RC) is empty. The category ’np’ of certain type is assigned as return mark-up (RM). In the example below the regular expression rule shows agreement in neuter gender. The rule encodes patterns which might start with an adverb/adverbs (‘D’). There are no restrictions on the grammatical characteristics and types of the participles (‘V’). Simple internal coordination between modifiers is included as well. Otherwise the sequences encode the possible orderings between elements, their obligatoriness or optionality:

```
<"D">*,
<("Ps#snl#"|"Psx#snl#"|"Pd#sn"|"Pi#sn"|"Prp#sn"|
  "Pf#sn"|"Pf#sn#"|"Pn#sn"|"Pc#sn"|"Pc#sn#"|
  "M@ns#"|"Ans@"|"V#car@sn#"|"V#cv#sn#"|
  "V#cao@sn#")>?,
<("Psx#t"|"Ps#t")>?,
<"Ppxa---t">?,<",">?,<"C">?,
<"D">?,<"V#c#s#">?,
<"Ans@">*,<"M#ns#">?,
<"N@nsi">
```

2. The context-insensitive submodule consists of more rules, which reflect not only the agreement-based patterns, but the variety of NP types (one-, two- and multi-componential), and NP internal word orderings as well. Its structure is as follows:

```
LC=empty
R=regular expression rules
RC=empty
RM=<np type="common">\w</np>
```

The left context area (LC) is empty. The regular expressions area (R) consists of rules, which start with non-problematic beginners only. The right context area (RC) is empty. The category ’np’ of certain type is assigned as return mark-up (RM). The example rule below aims at sequences of one-componential NPs being nouns and certain kinds of pronouns.

```
<("N#"|"Pp-o#"|"Pp-a#l"|"Pp-d#l"|"Ps#l#"|"Pdn#"|"Pdr#"|"Pda#"|"Pfe#"|
"Pfa#"|"Pfp#"|"Pfq#"|"Pfy#"|"Pie#"|"Pia#"|"Piq#"|"Piy#"|"Pre#"|"Pra#"|
"Prp#"|"Pce#"|"Pca#"|"Pcq#"|"Pne#"|"Pna#"|"Pnp#"|"Pnq#")>
```

5 Discussion of Some Problematic Cases

The nominal chunker faces some difficulties in encoding certain linguistic phenomena such as:

1. The substantivization of modifiers and elliptical NPs
   The definite adjectives/participles are not treated as head nouns on the chunk level, because:

   (a) the chunker relies on the information from the tagger, i.e. it treats as heads only tokens with morpho-syntactic tags of nouns and certain pronouns. All other generalizations lead to errors.

   (b) they might be just a beginning of a complex NP, not head at all as in (12):

   (11) visokijat do poshtata vleze
   tall-the-m,sg near post-office-the-fem,sg came in
   the tall man near the post-office came in

   (12) visokijat do tavana chovek vleze
   tall-the-m,sg to ceiling-the-n,sg man-m,sg came in
   the man, tall to the sky, came in

   At the moment this context dependency is underspecified, but it is a good reason for finding an adequate way of dealing with complex NPs as well.

2. Additionally we aim at exploring an idea for the identification of more complex and recursive NPs including the following possible modifying elements: PPs, relative and interrogative clauses, 'da'-clauses, 'che'-clauses.

   The problems here concern the generalized recognition of the clause boundaries, especially the end. Thus special strategies have to be applied in cascaded manner for the recognition of the simplest and unambiguous syntactic models first.

3. Partial capturing of some non-recursive NP groups
   Note that it happens in cases when the base NP starts with an adverb or a past/passive participle and does not occur after a preposition. These underspecifications could be repaired to some extent by the following steps: 1. for problematic participles - the incorporation of the verbal chunker and 2. for the adverbs - listing the definite adverbs (*malkoto* 'little-the', *povecheto* 'most-the', *mnogoto* 'many-the' etc.), which are 100 % nominal modifiers.

4. Verbal elements within NP chunks
   They are connected with the presence of the participles being a hybrid part-of-speech category. When the participle is captured unproblematically, then all the verbal clitics such as the clitic 'se' and the accusative and dative personal clitics are captured as well. When the participle is not captured, then the clitics are not captured either.

From all the problems, listed above, it becomes obvious, that the coverage of the chunker needs to be improved. Aiming at 100% accuracy, it covers only non-problematic base NPs. The problematic ones are only partially repaired for the moment.

6 Evaluation

At present the evaluation of the hand-crafted base nominal-phrase grammar is under preparation. Some preliminary tests, however, have been done by expert human inspection over the output data. At the same time a golden standard is being constructed for testing the accuracy and coverage of the NP chunker. Below we describe in more detail the present evaluation procedure and results against a preliminary 'golden standard' over 3500 words.

6.1 Procedure

The initial procedure, that has been performed, contains the following steps:

1. two newspaper texts consisting of about 3500 words have been manually annotated with base NPs
2. the NP chunk grammar has been run on the same texts (the version without annotation)
3. the NP grammar output has been compared to the 'golden standard' NPs. The number of mismatches has been detected and then divided into two files: 1. the cases, which have been manually annotated, but unrecognized by the grammar, i.e. missed chunks and 2. the cases, which have been recognized by the grammar, but not manually annotated, i.e. incorrect chunks.

6.2 Metrics

The errors in the matching procedure have been divided into two groups: 1. grammar-driven errors and 2. errors, caused by wrong morpho-syntactic tag prediction or unrecognized words.

The grammar-driven errors include undetected or partially detected NPs beginning with an unclear indicator (despite the fact that the grammar does not aim at them at this stage); NPs, which are not covered by the rule set yet; problematic NPs as: elliptical ones *drugi dvama* 'other two' or substantivized modifiers *zhelanoto* 'wished-the'.

The tagging errors include wrongly assigned morphological information or undetected and hence - unprocessed names, abbreviations etc.

The metrics are as follows:

1. Coverage: within texts of about 3500 words, from 954 manually tagged base NPs the grammar missed 21 NPs on fully reliable pre-processed text (i.e. 1014 guesses and 933 correct guesses) and 117 NPs otherwise (i.e. 924 guesses and 837 correct guesses).
2. Recall: with tagging errors excluded - 97,8% and with tagging errors included - 87,7%.
3. Precision: with tagging errors excluded - 92% and with tagging errors included - 90,1%.

Hence, the tagging repair seems to influence more significantly Recall and slightly increases Precision. But note that the evaluation metrics might change when more 'golden standard' NPs are compiled.

The chunk grammar itself contributes to the creation and improvement of the golden standard and of a test-suite. We envisage to support this process as follows: 1. running the grammar module on the core set of sentences, extracted from Bulgarian grammars and 2. manual repairing of the incorrect NPs and checking the possible problematic cases such as ellipsis etc. When enough reliable data is compliled, it could be incorporated in a learning corpus-based algorithm and tested on arbitrary corpus texts [Cardie and Pierce 1998]. Otherwise the learning experiments would suffer from the sparseness problem [Veenstra, Mueller and Ule 2002].

However, the comparison between the results from human-written rules and machine show that both of them are effective on the most frequent types only - [Brill and Ngai 1999]. For this reason, the most frequent NPs were the target of our chunker.

7 Nominal chunks - the interface between shallow parsing and HPSG

Chunking is considered to be a theory independent step, which precedes the deep syntactic analyses. But we have to take into account its further integration into the HPSG-based implementation. Such an attempt was introduced in [Richter and Sailer 1996] for German with respect to the word order, for example.

Within BulTreeBank project the HPSG grammar itself is viewed as a definition platform for the annotation scheme of the treebank. Hence, the preparation of the annotation scheme requires a proper handling of the following specific tasks:

1. Identification of the relevant for Bulgarian part of the HPSG hierarchy as described in [Pollard and Sag 1994]; its modification with respect to the language specific phenomena.

2. Representation of the HPSG Universal Grammar principles and their parametrization for Bulgarian.

3. Mapping the information from the compiled language tools and resources into HPSG compatible format, i.e. 1. adapting the lexical and grammatical information from the morphological dictionary of Bulgarian to the sort hierarchy and to HPSG principles and 2. mapping chunking rules into dependency relations.

In this section we concentrate on the third task, which presupposes the previous two and ensures a platform for deeper syntactic processing.

This integration can be done within several mutually constraining directions:

1. The grammar rules present the main grammatical patterns within NPs, such as gender and number agreement or specific element order. This information can be used in producing the HPSG analyses. One advantage to be explored here is that chunking relies on the notion of head. Thus the head-adjunct structures are predicted easily during the chunking stage.

2. The compiled ‘golden standard’ set of sentences, which are assumed to represent the main NP patterns, can be used as a top-down filtering on the NP chunk patterns. Especially in cases, where the NP parses remained incomplete due to vague starting indicators. Thus the ‘golden standard’ patterns would add the relevant information to the incomplete parses.

3. The corpus itself can be used as a correcting mechanism with its data-driven frequency NP patterns. These patterns will help for modelling the typical grammatical relations within NPs.

Nominal Chunks and head-lexicalized relations

The morphosyntactic tags of the NP elements show the terminal categories of the constituent, but we aim at deriving explicit head-modifier information. This task is not trivial, because: 1. it is theory dependent and 2. it relies on the language specific decisions over certain linguistic phenomena.

For the base, non-recursive Bulgarian NPs we assume that the internal head-lexicalized elements are: *heads*, *adjuncts of different kinds*, *lexicalized prosodic groups like 'a definite adjective plus a possessive clitic'*.

Transforming Nominal Chunk Rules into HPSG-based dependency relations

In our work we rely on the support of two software tools: the CLaRK system for partial analyses and TRALE system [Götz and Meurers, 1997] for HPSG-based parsing. Thus the chunk output from the CLaRK system has to be in accordance with HPSG relational specifications. We decided to pre-encode the grammatical information into head-lexicalized features in the following way:

1. first, we have created the basic NP chunk grammar. As it was shown above, it consists of number of rules. Here two examples of shorter rules are given for simplicity:

```
Rule1 (target types: shapkata mi 'hat-the-my')

LC=empty
R="N#",("Psx#t"|"Ps#t")
RC=empty
RM=<np type="common">\w<np>
```

   The rule encodes the following: the left and right context areas are empty; `N#` stands for any noun; `Psx#t` stands for any reflexive-possessive pronoun clitic and `Ps#t` stands for any possessive pronoun clitic.

```
Rule2 (target types: vsichki moi shapki  'all my hats', niakakvi
svoi shapki 'some self hats', tezi moi shapki 'these my hats',
vsiaka moja shapka 'every my hat')

LC=empty
R=("Pc#"|"Pf#"|"Pd#"),("Ps#"|"Psx#")?,"N#"
RC=empty
RM=<np type="common">\w<np>
```

   The rule encodes the following: the left and right context areas are empty; `N#` stands for any noun; `Pc#` stands for any collective pronoun, and `Pf#` stands for any indefinite pronoun, `Pd#` stands for any demonstrative pronoun, `Ps#` stands for any possessive pronoun, `Psx#` stands for every reflexive-possessive pronoun.

2. As a second step, we pre-encode every chunk rule into subrules and apply the new rules within NP chunks. The following transformations are made: 1. every conjunct from the rule (no matter complex or not) occupies the position of the regular expression area from left to right direction. In this way when the last element of the chunk rule is in the regular expression area, all the others are available in the left context area. From the second new subrule on, the right context area becomes occupied as well. The category of every subrule, encoded as return mark-up next obeys the HPSG-based and modified for Bulgarian dependency relations as head, adjunct, complement, clitic etc. Here we give an example of such a decomposing strategy over just the first simple rule:

   ```
   Rule1 - new subrule1

   LC="N#"
   R="Psx#t"|"Ps#t"
   RC=empty
   RM=<clitic>\w<clitic>
   ```

   After the application of the first sub-rule, the possessive clitic receives the special clitic tag.

   ```
   Rule1 - new subrule2

   LC=empty
   R="N#"
   RC="Psx#t"|"Ps#t"
   RM=<head>\w<head>
   ```

   After the application of the second sub-rule, the noun receives the head tag.

   In this way we receive output structures in XML like the one below (note that the morphological tags are omitted for clarity):

   ```
   <np><head>shapkata</head> <clitic>mi</clitic></np>
   ```

   Similarly the result from the second rule is:

   ```
   <np><adjunct>tova</adjunct> <head>neshto</head></np>
   ```

Hence, by this strategy, the set of the subrules, constituting one rule within a grammar, gives dependencies and their linearity, valid for the recognized structure.

8 Conclusion and Outlook

This paper aimed at two things: describing the base nominal chunker for Bulgarian and presenting an idea on the transformation of its output into head-lexicalized relations. The two topics of discussion were related to the specific language data problems and the implementation.

The future tasks are viewed as follows:

1. applying the described evaluation procedure against an extended version of a 'golden standard'
2. relating the problem of the complex NPs with the problem of clausal boundaries detection in Bulgarian
3. more intensive experiments of the NP chunk grammar with other chunk grammars such as the VP module

9 Acknowledgements

I would like to thank Kiril Simov and two anonymous reviewers for their valuable comments on the earlier drafts of this paper. All remaining errors are my own.

References

[Aarts 1992] Aarts B. *Small clauses in English: the nonverbal types.* Berlin and New York: Mouton de Gruyter.

[Abney 1996] Abney S. *Chunk Stylebook*
On http://sfs.nphil.uni-tuebingen.de/ abney/Papers.html, draft.

[Abney 1997] Abney S. *The SCOL Manual. Version 0.1b.*
On http://citeseer.nj.nec.com/abney97scol.html

[Avramova, Osenova in press] Avramova T., Osenova P. *Otnovo po vaprosa za granitsata mejdu slojna duma i slovosachetanie.* In Balgarski ezik, (in press)(in Bulgarian).

[Brill and Ngai 1999] Brill E. and Ngai G. *Man vs. Machine: A Case study in Base Noun Phrase Learning* In Proceedings of ACL 1999. Association for Computational Linguistics.

[Cardie and Pierce 1998] Cardie, C. and Pierce, D. *Error-Driven Pruning of Treebank Grammars for Base Noun Phrase Identification.* In Proceedings of ACL-Coling, Morgan Kaufmann, San Francisco, pp. 218-224.

[Daelemans et. al. 1999] Daelemans W., Buchholz S. and Veenstra J. *Memory-based Shallow Parsing.* In Proceedings of the EACL'99 workshop on Computational Natural Language Learning (CoNLL-99), Bergen, Norway, June 1999.

[Ivanova and Dojkoff 2002] Ivanova K. and Dojkoff D. *Cascaded regular grammars and constraints over morphologically annotated data for ambiguity resolution.* In Proceedings from the Workshop on Linguistic Theories and Treebanks, 20-21 Sept., Sozopol, Bulgaria (in this volume).

[Götz and Meurers, 1997] Thilo Götz and W. Detmar Meurers. *The ConTroll system as large grammar development platform.* In *Proceedings of the ACL/EACL post-conference workshop on Computational Environments for Grammar Development and Linguistic Engineering.* Madrid, Spain.

[Mueller 2002] Mueller, F. H. *Shallow-Parsing Stylebook for German* Technical report, SfS, Universitat Tuebingen.
On http://www.sfs.nphil.uni-tuebingen.de/dereko/anno-doc.html

[Ngai and Yarowsky 2000] Ngai, G. and Yarowsky, D. *Rule Writing or Annotation: Cost-efficient Resource Usage for Base Noun Phrase Chunking.* In Proceedings of ACL-2000, Hong Kong, pp. 117-125.

[Osenova to appear] Osenova P. *Imennite grupi ot tipa NN v balgarskija ezik.* In Proceedings from the Slavic Readings Conference, 27-28 April 2002, to appear (in Bulgarian).

[Osenova and Kolkovska 2002] Osenova P. and Kolkovska S. *Combining the named-entity recognition task and NP chunking strategy for robust pre-processing* In Proceedings of the Workshop on Linguistic Theories and Treebanks, 20-21 Sept., Sozopol, Bulgaria (in this volume).

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

[Pollard and Sag 1994] Pollard C. and Sag I. *Head-driven Phrase Structure Grammar.* University of Chicago Press, Chicago, Illinois, USA.

[Richter and Sailer 1996] Richter F. and Sailer M. *Regions and Word Order* Handout (9 pages) for a talk given at the International Center Workshop on Computational Linguistics in Tuebingen on September 20th, 1996.

[Simov, Popova and Osenova 2001] Simov K. , Popova G. and 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, pp. 135-142.

[Simov et. al. 2002a] Simov K., Kouylekov M. and Simov A. *Cascaded Regular Grammars over XML Documents.* In Proc. of the 2nd Workshop on NLP and XML (NLPXML-2002), COLING2002, Taipei, Taiwan. September 1, 2002. (to appear)

[Simov et.al. 2002b] Simov K., Osenova P., Slavcheva M., Kolkovska S., Balabanova E., Dojkoff D., Ivanova K., Simov A., Kouylekov M. *Building a Linguistically Interpreted Corpus of Bulgarian: the BulTreeBank.* In Proceedings from the LREC conference 2002, Canary Islands, pp. 1729-1736.

[Veenstra, Mueller and Ule 2002] Veenstra J., Mueller F. and Ule T. *Topological field chunking for German* In Proceedings from the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics, Trento, Italy. August 5-16, 2002, pp. 91-98.