# Shallow language processing architecture for Bulgarian

**Hristo Tanev**
ITC-Irst,
Centro per la Ricerca Scientifica e Tecnologica
Povo,Trento, Italy 38050
tanev@itc.it

**Ruslan Mitkov**
School of Humanities,
Languages and Social Studies
Wolverhampton WV1 1SB UK
R.Mitkov@wlv.ac.uk

## Abstract

This paper describes LINGUA - an architecture for text processing in Bulgarian. First, the pre-processing modules for tokenisation, sentence splitting, paragraph segmentation, part-of-speech tagging, clause chunking and noun phrase extraction are outlined. Next, the paper proceeds to describe in more detail the anaphora resolution module. Evaluation results are reported for each processing task.

## 1 Introduction

The state of the art of today's full parsing and knowledge-based automatic analysis still falls short of providing a reliable processing framework for robust, real-world applications such as automatic abstracting or information extraction. The problem is especially acute for languages which do not benefit from a wide range of processing programs such as Bulgarian. There have been various projects which address different aspects of the automatic analysis in Bulgarian such as morphological analysis (Krushkov, 1997), (Simov et al., 1992), morphological disambiguation (Simov et al., 1992) and parsing (Avgustinova et al., 1989), but no previous work has pursued the development of a knowledge-poor, robust processing environment with a high level of component integrity. This paper reports the development and implementation of a robust architecture for language processing in Bulgarian referred to as LINGUA, which includes modules for POS tagging, sentence splitting, clause segmentation, parsing and anaphora resolution. Our text processing framework builds on the basis of considerably shallower linguistic analysis of the input, thus trading off depth of interpretation for breadth of coverage and workable, robust solution. LINGUA uses knowledge poor, heuristically based algorithms for language analysis, in this way getting round the lack of resources for Bulgarian.

## 2 LINGUA - an architecture for language processing in Bulgarian

LINGUA is a text processing framework for Bulgarian which automatically performs tokenisation, sentence splitting, part-of-speech tagging, parsing, clause segmentation, section-heading identification and resolution for third person personal pronouns (Figure 1). All modules of LINGUA are original and purpose-built, except for the module for morphological analysis which uses Krushkov's morphological analyser BULMORPH (Krushkov, 1997). The anaphora resolver is an adaptation for Bulgarian of Mitkovs knowledge-poor pronoun resolution approach (Mitkov, 1998).

LINGUA was used in a number of projects covering automatic text abridging, word semantic extraction (Totkov and Tanev, 1999) and term extraction. The following sections outline the basic language processing functions, provided by the language engine.

### 2.1 Text segmentation: tokenisation, sentence splitting and paragraph identification

The first stage of every text processing task is the segmentation of text in terms of tokens, sentences and paragraphs.

LINGUA performs text segmentation by operating within an input window of 30 tokens, applying rule-based algorithm for token synthesis, sentence splitting and paragraph identification.

#### 2.1.1 Tokenisation and token stapling

Tokens identified from the input text serve as input to the token stapler. The token stapler forms more complex tokens on the basis of a

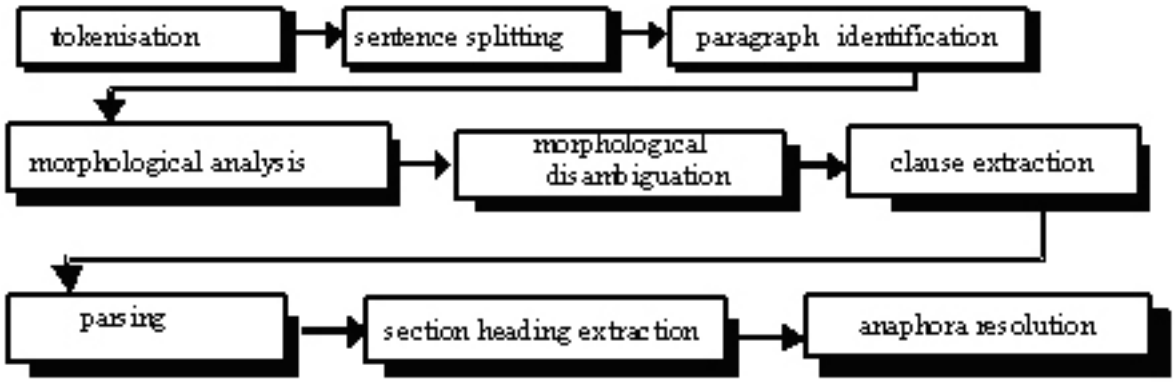


Figure 1: General architecture of LINGUA

token grammar. With a view to improving tokenisation, a list of abbreviations has been incorporated into LINGUA.

#### 2.1.2 Sentence splitting

LINGUA's sentence splitter operates to identify sentence boundaries on the basis of 9 main end-of-sentence rules and makes use of a list of abbreviations. Some of the rules consist of several finer sub-rules. The evaluation of the performance of the sentence splitter on a text of 190 sentences reports a precision of 92% and a recall of 99%. Abbreviated names such as J.S.Simpson are filtered by special constraints. The sentence splitting and tokenising rules were adapted for English. The resulting sentence splitter was then employed for identifying sentence boundaries in the Wolverhampton Corpus of Business English project.

#### 2.1.3 Paragraph identification

Paragraph identification is based on heuristics such as cue words, orthography and typographical markers. The precision of the paragraph splitter is about 94% and the recall is 98% (Table 3).

### 2.2 Morphological analysis and part-of-speech tagging

#### 2.2.1 Morphological analysis

Bulgarian morphology is complex, for example the paradigm of a verb has over 50 forms. Krushkov's morphological analyser BULMORPH (Krushkov, 1997) is integrated in the language engine with a view to processing Bulgarian texts at morphological level.

#### 2.2.2 Morphological disambiguation

The level of morphological ambiguity for Bulgarian is not so high as it is in other languages. As a guide, we measured the ratio: $Number\_of\_all\_tags/Number\_of\_all\_words$. The results show that this ratio is comparatively low and for a corpus of technical texts of 9000 words the ratio tags per word is 1,26, whereas for a 13000-word corpus from the genre of fiction this ratio is 1,32. For other languages such as Turkish this ratio is about 1,9 and for certain English corpora 2,0 [1].

We used 33 hand-crafted rules for disambiguation. Since large tagged corpora in Bulgarian are not widely available, the development of a corpus-based probabilistic tagger was an unrealistic goal for us. However, as some studies suggest (Voutilainen, 1995), the precision of rule-based taggers may exceed that of the probabilistic ones.

### 2.3 Parsing

Seeking a robust flexible solution for parsing we implemented two alternative approaches in LINGUA: a fast-working NP extractor and more general parser, which works more slowly, but delivers better results both in accuracy and coverage. As no syntactically annotated Bulgarian corpora were available to us, using statistical data to implement probabilistic algorithm was not an option.

The NP extraction algorithm is capable of analysing nested NPs, NPs which contain left

[1] Kemal Oflazer, personal communication

modifiers, prepositional phrases and coordinating phrases. The NP extractor is based on a simple unification grammar for NPs and APs. The recall of NP extraction, measured against 352 NPs from software manuals, was 77% and the precision - 63.5%.

A second, better coverage parser was implemented which employs a feature grammar based on recent formal models for Bulgarian, (Penchev, 1993), (Barkalova, 1997). All basic types of phrases such as NP, AP, PP, VP and AdvP are described in this grammar. The parser is supported by a grammar compiler, working on grammar description language for representation of non context unification grammars. For example one of the rules for synthesis of NP phrases has the form:
$NP(def:Y\ full\_art:F\ ext:+\ rex:-\ nam:-)$
$\rightarrow AP(gender{:}X\ def{:}Y\ full\_art{:}F\ number{:}\ L\ )$
$NP(ext{:}-\ def{:}-\ number{:}L\ gender{:}X\ rex{:}-)$

The features and values in the rules are not fixed sets and can be changed dynamically. The flexibility of this description allows the grammar to be extended easily. The parser uses a chart bottom-up strategy, which allows for partial parsing in case full syntactic tree cannot be built over the sentence.

There are currently about 1900 syntactic rules in the grammar which are encoded through 70 syntactic formulae.

Small corpus of 600 phrases was syntactically annotated by hand. We used this corpus to measure the precision of the parsing algorithm (Table 3).

We found that the precision of NP extraction performed by the chart parser is higher than the precision of the standalone NP extraction - 74.8% vs. 63.5% while the recall improves by only 0.9% - 77.9% vs. 77% .

The syntactic ambiguity is resolved using syntactic verb frames and heuristics, similar to the ones described in (Allen, 1995).

The parser reaches its best performance for NPs (74.8% precision and 77.9% recall) and lowest for VPs (33% precision, 26% recall) and Ss (20% precision and 5.9% recall) (Table 3). The overall (measured on all the 600 syntactic phrases) precision and recall, are 64.9% and 60.5% respectively. This is about 20% lower, compared with certain English parsers (Murat and Charniak, 1995), which is due to the insufficient grammar coverage, as well as the lack of reliable disambiguation algorithm. However the bracket crossing accuracy is 80%, which is comparable to some probabilistic approaches. It should be noted that in our experiments we restricted the maximal number of arcs up to 35000 per sentence to speed up the parsing.

## 3 Anaphora resolution in Bulgarian

### 3.1 Adaptation of Mitkovs knowledge-poor approach for Bulgarian

The anaphora resolution module is implemented as the last stage of the language processing architecture (Figure 1). This module resolves third-person personal pronouns and is an adaptation of Mitkov's robust, knowledge-poor multilingual approach (Mitkov, 1998) whose latest implementation by R. Evans is referred to as MARS [2] (Orasan et al., 2000). MARS does not make use of parsing, syntactic or semantic constraints; nor does it employ any form of non-linguistic knowledge. Instead, the approach relies on the efficiency of sentence splitting, part-of-speech tagging, noun phrase identification and the high performance of the antecedent indicators; knowledge is limited to a small noun phrase grammar, a list of (indicating) verbs and a set of antecedent indicators.

The core of the approach lies in activating the *antecedent indicators* after filtering candidates (from the current and two preceding sentences) on the basis of gender and number agreement and the candidate with the highest composite score is proposed as antecedent [3]. Before that, the text is pre-processed by a sentence splitter which determines the sentence boundaries, a part-of-speech tagger which identifies the parts of speech and a simple phrasal grammar which detects the noun phrases. In the case of complex sentences, heuristic 'clause identification' rules track the clause boundaries.

LINGUA performs the pre-processing, needed as an input to the anaphora resolution algorithm: sentence, paragraph and clause splitters, NP grammar, part-of-speech tagger,

[2] MARS stands for Mitkov's Anaphora Resolution System.

[3] For a detailed procedure how candidates are handled in the event of a tie, see (Mitkov, 1998).

| Text | | Pronouns | Weight set | | |
|---|---|---|---|---|---|
| | | | Standard | Optimised | Baseline most recent |
| Software manuals | Success rate | 221 | 75.0% | 78.8% | 58.0% |
| | Critical succ. rate | | 70.0% | 73.0% | 54.0% |
| | Non trivial succ. rate | | 70.0% | 78.8% | 58.0% |
| Tourist guides | Success rate | 116 | 68.1% | 69.8% | 65.0% |
| | Critical succ. rate | | 63.3% | 64.4% | 58.8% |
| | Non trivial succ. rate | | 67.2% | 69.0% | 65.0% |
| All texts | Success rate | 337 | 72.6% | 75.7% | 60.4% |
| | Critical succ. rate | | 67.7% | 70.0% | 55.7% |
| | Non trivial succ. rate | | 72.3% | 75.4% | 60.4% |

Table 1: Success rate of anaphora resolution

section heading identification heuristics. Since one of the indicators that Mitkov's approach uses is term preference, we manually developed[4] a small term bank containing 80 terms from the domains of programming languages, word processing, computer hardware and operating systems [5]. This bank additionally featured 240 phrases containing these terms.

The antecedent indicators employed in MARS are classified as *boosting* (such indicators when pointing to a candidate, reward it with a bonus since there is a good probability of it being the antecedent) or *impeding* (such indicators penalise a candidate since it does not appear to have high chances of being the antecedent). The majority of indicators are genre-independent and are related to coherence phenomena (such as *salience* and *distance*) or to structural matches, whereas others are genre-specific (e.g. *term preference*, *immediate reference*, *sequential instructions*). Most of the indicators have been adopted in LINGUA without modification from the original English version (see (Mitkov, 1998) for more details). However, we have added 3 new indicators for Bulgarian: *selectional restriction pattern*, *adjectival NPs* and *name preference*.

The boosting indicators are

*First Noun Phrases:* A score of +1 is assigned to the first NP in a sentence, since it is deemed to be a good candidate for the antecedent.

*Indicating verbs:* A score of +1 is assigned to those NPs immediately following the verb which is a member of a previously defined set such as discuss, present, summarise etc.

*Lexical reiteration:* A score of +2 is assigned those NPs repeated twice or more in the paragraph in which the pronoun appears, a score of +1 is assigned to those NP, repeated once in the paragraph.

*Section heading preference:* A score of +1 is assigned to those NPs that also appear in the heading of the section.

*Collocation match:* A score of +2 is assigned to those NPs that have an identical collocation pattern to the pronoun.

*Immediate reference:* A score of +2 is assigned to those NPs appearing in constructions of the form " ...$V_1$ NP $<$ CB $>$ $V_2$ it " , where $<$ CB $>$ is a clause boundary.

*Sequential instructions:* A score of +2 is applied to NPs in the $NP_1$ position of constructions of the form: "To $V_1$ $NP_1$ ... To $V_2$ it ..."

*Term preference:* a score of +1 is applied to those NPs identified as representing domain terms.

*Selectional restriction pattern:* a score of

[4]This was done for experimental purposes. In future applications, we envisage the incorporation of automatic term extraction techniques.

[5]Note that MARS obtains terms automatically using TF.IDF.

| Text | Pronouns | Intrasentential: Intersentential anaphors | Average number of candidates per anaphor | Average distance from the antecedent in clauses | Average distance from the antecedent in sentences | Average distance from the antecedent in NP |
|---|---|---|---|---|---|---|
| Sofware manuals | 221 | 106 : 115 | 3.29 | 1.10 | 0.62 | 3.30 |
| Tourist guides | 116 | 17 : 99 | 3.35 | 1.74 | 0.98 | 5.13 |

Table 2: Complexity of the evaluation data

+2 is applied to noun phrases occurring in collocation with the verb preceding or following the anaphor. This preference is different from the collocation match preference in that it operates on a wider range of 'selectional restriction patterns' associated with a specific verb [6] and not on exact lexical matching. If the verb preceding or following the anaphor is identified to be in a legitimate collocation with a certain candidate for antecedent, that candidate is boosted accordingly. As an illustration, assume that 'Delete file' has been identified as a legitimate collocation being a frequent expression in a domain specific corpus and consider the example 'Make sure you save *the file* in the new directory. You can now delete *it*. ' Whereas the 'standard' *collocation match* will not be activated here, the *selectional restriction pattern* will identify '*delete file*' as an acceptable construction and will reward the candidate '*the file*'.

*Adjectival NP:* a score of +1 is applied to NPs which contain adjectives modifying the head. Empirical analysis shows that Bulgarian constructions of that type are more salient than NPs consisting simply of a noun. Recent experiments show that the success rate of the anaphora resolution is improved by 2.20%, using this indicator. It would be interesting to establish if this indicator is applicable for English.

*Name preference:* a score +2 is applied to names of entities (person, organisation, product names).

The impeding indicator is *Prepositional Noun Phrases*: NPs appearing in prepositional phrases are assigned a score of -1.

Two indicators, *Referential distance* and *Indefiniteness* may increase or decrease a candidate's score.

*Referential distance* gives scores of +2 and +1 for the NPs in the same and in the previous sentence respectively, and -1 for the NPs two sentences back. This indicator has strong influence on the anaphora resolution performance, especially in the genre of technical manuals. Experiments show that its switching off can decrease the success rate by 26% .

*Indefiniteness* assigns a score of -1 to indefinite NPs, 0 to the definite (not full article) and +1 to these which are definite, containing the definite 'full' article in Bulgarian.

## 4 Evaluation of the anaphora resolution module

The precision of anaphora resolution measured on corpus of software manuals containing 221 anaphors, is 75.0%. Given that the anaphora resolution system operates in a fully automatic mode, this result could be considered very satisfactory. It should be noted that some of the errors arise from inaccuracy of the preprocessing modules such as clause segmentation and NP extraction (see Table 3).

We also evaluated the anaphora resolution system in the genre of tourist texts. As expected, the success rate dropped to 68.1% which, however, can still be regarded as a very

[6] At the moment these patterns are extracted from a list of frequent expressions involving the verb and domain terms in a purpose-built term bank but in generally they are automatically collected from large domain-specific corpora.

| Language processing module | Precision % | Recall % | Evaluation data |
|---|---|---|---|
| sentence splitter | 92.00 | 99.00 | 190 sentences |
| paragraph splitter | 94.00 | 98.00 | 268 paragraphs |
| clause chunker | 93.50 | 93.10 | 232 clauses |
| POS tagger | 95.00 | 95.00 | 303 POS tags |
| NP extractor | 63.50 | 77.00 | 352 NPs |
| chart parsing | | | |
| NP | 74.84 | 77.89 | 294 NPs |
| AP | 65.15 | 67.19 | 64 APs |
| AdvP | 37.14 | 50.00 | 26 AdvPs |
| VP | 33.33 | 26.39 | 72 VPs |
| PP | 70.00 | 60.21 | 93 PPs |
| S | 20.00 | 5.88 | 51 Ss |
| Total | 64.93 | 60.50 | 600 phrases |
| Bracket crossing accuracy | 80.33 | - | 600 phrases |
| Anaphora resolution | 72.60 | - | 337 anaphors |

Table 3: Summary of LINGUA performance

good result, given the fact that neither manual pre-editing of the input text, nor any post-editing of the output of the pre-processing tools were undertaken. The main reason for the decline of performance is that some of the original indicators such as term preference, immediate reference and sequential instructions of the knowledge-poor approach, are genre specific.

The software manuals corpus featured 221 anaphoric third person pronouns, whereas the tourist text consisted of 116 such pronouns. For our evaluation we used the measures success rate, critical success rate and non-trivial success rate (Mitkov, 2001). Success rate is the ratio $SR = AC/A$, where AC is the number of correctly resolved and A is the number of all anaphors. Critical success rate is the success rate for the anaphors which have more than one candidates for antecedent after the gender and number agreement filter is applied. Non-trivial success rate is calculated for those anaphors which have more than one candidates for antecedent before the gender and number agreement is applied. We also compared our approach with the typical baseline model *Baseline most recent* which takes as antecedent the most recent NP matching the anaphor in gender and number. The results are shown in the Table 1.

These results show that the performance of LINGUA in anaphora resolution is comparable to that of MARS (Orasan et al., 2000). An optimised version [7] of the indicator weights scored a success rate of 69,8% on the tourist guide texts, thus yielding an improvement of 6,1%.

Table 2 illustrates the complexity of the evaluation data by providing simple quantifying measures such as average number of candidates per anaphor, average distance from the anaphor to the antecedent in terms of sentences, clauses, intervening NPs, number of intrasentential anaphors as opposed to intersentential ones etc.

## 5 Conclusion

This paper outlines the development of the first robust and shallow text processing framework in Bulgarian LINGUA which includes modules for tokenisation, sentence splitting, paragraph segmentation, part-of-speech tagging, clause chunking, noun phrases extraction and anaphora resolution (Figure 1). Apart from the module on pronoun resolution which was adapted from Mitkov's knowledge-poor approach for English and the incorporation of BULMORPH in the part-of-speech tagger, all modules were specially built for LINGUA. The evaluation shows promising results for each of the modules.

[7] The optimisation made use of genetic algorithms in a manner similar to that described in (Orasan et al., 2000).

## References


J. Allen. 1995. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc.

T. Avgustinova, K. Oliva, and E. Paskaleva. 1989. An HPSG-based parser for bulgarian. In *International Seminar on Machine Translation 'Computer and Translation 89'*, Moscow, Russia.

P. Barkalova. 1997. *Bulgarian syntax - known and unknown*. Plovdiv University Press, Plovdiv. in Bulgarian.

H. Krushkov. 1997. *Modelling and building of machine dictionaries and morphological processors*. Ph.D. thesis, University of Plovdiv. in Bulgarian.

R. Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *Proceedings of the 18.th International Conference on Computational Linguistics (COLING'98)/ACL'98 Conference*, pages 869–875, Montreal,Canada.

R. Mitkov. 2001. Towards a more consistent and comprehensive evaluation of anaphora resolution algorithms and systems. *Towards a more consistent and comprehensive evaluation of anaphora resolution algorithms and systems*, (15):253–276.

E. Murat and E. Charniak. 1995. A statistical syntactic disambiguation program and what it learns. *CS*, 29-95.

C. Orasan, R. Evans, and R. Mitkov. 2000. Enhancing preference-based anaphora resolution with genetic algorithms. In *Proceedings of NLP'2000*, Patras, Greece.

J. Penchev. 1993. *Bulgarian Syntax - Government and Binding*. Plovdiv University Press, Plovdiv. in Bulgarian.

K. Simov, E. Paskaleva, M. Damova, and M. Slavcheva. 1992. Morpho-assistant - a knowledge based system for bulgarian morphology. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy.

G. Totkov and Ch. Tanev. 1999. Computerized extraction of word semantics through connected text analysis. In *Proc. of the International Workshop DIALOGUE '99*, pages 360 – 365.

A. Voutilainen. 1995. A syntax-based part-of-speech tagger. In *Proceedings of the 7th conference of the European Chapter of EACL*, Dublin, Ireland.