ЮБИЛЕЙНА НАУЧНА СЕСИЯ – 30 години ФМИ
ПУ "Паисий Хилендарски", Пловдив, 3-4.11.2000

METHODOLOGY, RESOURCES AND TOOLS FOR COMPUTERIZATION OF BULGARIAN LANGUAGE (1988-2000)

George Atanassov Totkov

The main directions in the field of computerization of natural languages developed in the Department of Computer Science (University of Plovdiv) are presented and discussed. Our global idea [19, 20, 21] is to automatise the processes of extracting, building and updating of linguistic resources and software tools. There are proposed effective methodology, methods, techniques and software tools realizing this idea in last years (1988-2000). Methods for research of text corpora and automatic extraction of unknown grammatical characteristics and structures are offered and applied. The developed techniques and tools are applicable to other natural languages too. There are obtained concrete results and resources for Bulgarian as follows:

- lexical database – morphological dictionary of 67 500 entries (words in their basic form) and series of machine dictionaries (for proper names, rhymes, robust morphological analysis of unknown words, derivations, word frequency, homonyms, verb clusters, etc.)
- software tools for tokenisation, sentence splitting, part of sentence tagging; clause splitting and noun phrase extraction, acquisition of features of unknown language elements, anaphora resolution, etc.

The presented approaches allow decreasing the volume of the dictionaries, linguistics data bases and software tools necessary for the analysis and synthesis of natural language texts.

2000 Mathematics Subject Classification: *68T50 Natural language processing*

Formal Model of Machine Dictionaries

Feature structures are the most usable formalism in the field of computational lexicography. The lexical entry in this structure has some natural language features (morphological, syntactic and semantic). An automatic determination of these features has not been implemented.

In [25] a mathematical model of machine dictionary is proposed. The model is appropriate for automatic construction and minimizing of the number of lexical entries computer analysis and synthesis of natural language texts creation of a system of machine dictionaries (lexical database) to be suited for all inflective languages.

Automated Construction of Machine Dictionaries

A number of research methods of text corpora and automatic extraction of unknown grammatical characteristics and structures are offered and applied [19, 20]. In [4, 20] methods for automatic defining of linguistic characteristics of 'unknown' words have been elaborated through an auxiliary dictionary for robust morphological analysis. A number of experiments with the heuristic methods for automated creation of dictionaries have been carried out. In [11] the opportunities for compression of machine dictionaries by using the lexical information stored in them are studied. The respective software tools are designed and developed [4-9].

Computer Dictionaries for Bulgarian

A representative lexical database is built up - a morphological dictionary of 67 500 entries (words in their basic form), a dictionary of proper names, rhyming dictionary, model dictionary for approximate morphological analysis of unknown words, derivational dictionary, frequency dictionary, reverse dictionary, homonymic dictionary; dictionary of verbal clusters, etc. New dictionaries are created [9, 27] through processing already built dictionaries using the knowledge stored in them (Fig. 1). The most productive dictionaries are orthographic and morphological dictionaries.

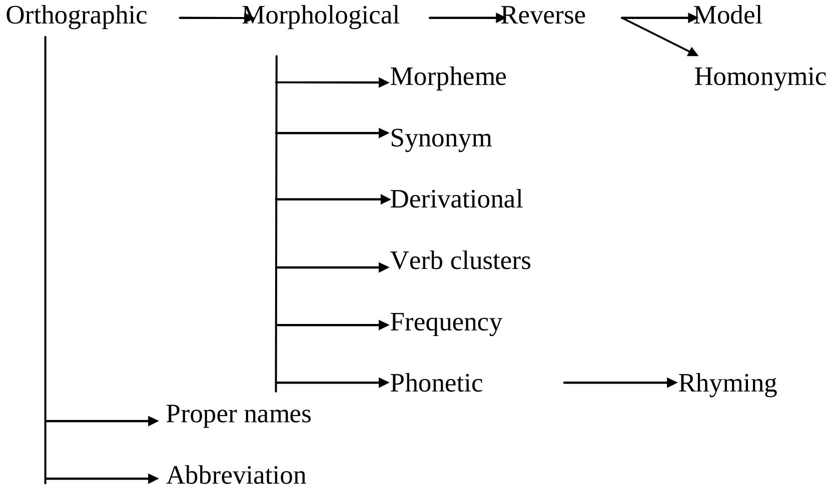

Figure 1. Creation of new dictionaries through transformations

Computerized Morphological Analysis and Synthesis

A morphological processor for Bulgarian was built on the basis of common properties to which submit inflective morphology [23, 26]. A part of speech is represented as set of classes. Every set can be divided into subsets depending on criteria pertaining to this particular part of speech. There are 231 different inflectional types in that classification. The inflectional type number determines the part of speech the analyzed word belongs to. From a mathematical point of view the Bulgarian words are divided into disjoint classes of equivalence. Every class has a unique machine number for identification and a list of rules for generation of the paradigm. Two words are in the same class if their paradigms are generated in the same way. The paradigm is described as a list of word-forms with specific grammatical features for each of them. Every word-form also has a number. Two word-forms with equal numbers have the same grammatical features.

It has been found [27] that there is a possibility for computer-aided building of a universal generator of optimum (on different criteria as rate, volume. accuracy of hypothesis) parsers.

The task for automated generating of such parsers can be only considered solved when binary trees have been for making analysis. It has been made up an algorithm for building such trees through a local, node optimization memory necessary for the semantic structure that helps identifying the grammatical of the word–forms in the Academic Dictionary is not more than 60 K (identifying of more than 15 word–forms through one bit!). Furthermore, for "not-known" word–forms or for word–forms that have been written in a wrong way, the particular procedure offers a hypothesis for identifying of the above–mentioned characteristics with precision of 99%.

Morphological Synthesis

For every word a proper pattern is built up. The pattern and the inflectional type number determine the paradigm of that word. The pattern shows which letters are constant in all word-forms in the paradigm of the word and which are changing. The changing letters are marked with '*' in the pattern. The pattern and the inflectional type number incorporate information for the whole paradigm of a particular word (incl. proper noun [15]). The inflectional type involves for every word-form a list of letters (including the empty one) for replacing the symbol '*' of the pattern and morphemes for appending the pattern. The pattern can be extracted from the paradigm automatically [2, 10]. The synthesis of the morphological paradigm is based on the following simple mechanism. Every word-form can be constructed from the pattern operating only with the rules of replacing (*/letter) and appending (+ morpheme), described after the word-form number. Once extracted, the rules for a member of some inflectional type are the same for all other members of this type.

The Bulgarian verb system is very rich of forms and meanings. The automatic processing of such forms is a very important prerequisite for the syntactic parsing. A formal model of complex verb forms as well as an algorithm for automatic analysis and synthesis of these forms is presented [13]. The software module contains an editor for input and updating the model of complex verb forms as well as functions for analysis and synthesis of such forms.

Unlike the number, definiteness, tense and other grammatical categories, the category aspect is permanent for all word-forms in the paradigm of the verb. This category was not included yet in the lexical database. It was not necessary in the case of morphological analysis. The missing of this category brings some problems in case of morphological synthesis as the overgeneration etc. The main goal of the investigation [14] is to define procedures for automatic guessing of the verb aspect and attaching this category to every verb entry of the lexical database.

Morphological Analysis

The goal of the automatic morphological analysis is to perform automatically a morphological classification of an arbitrary word-form. This includes identifying the base form of the word, its grammatical features and to which inflectional type (part of speech) it belongs. In case of homonyms (when the word-form belongs to more than one inflectional types and has different grammatical features) all possible types must be found.

A machine dictionary consists of word-pattern entries with associated inflectional type number. When an arbitrary word-form has to be classified the analyzer looks up a matching word-pattern in the dictionary. If such a pattern has been found, using the second part of the entry pair (inflectional type number) the rules are extracted from the generation table. On the basis of these rules a paradigm from this pattern is generated. If the analyzed word coincides with a word-form from the generated paradigm it obtains the grammatical features of that word-form. In such way the word is morphologically completely determined.

Morphological disambiguation is performed through grammar rules and heuristics, which take into account the context of the word and choose only one morphological hypothesis. Object oriented technology is used in implementation of the algorithm. Experiment was carried out with texts from the area of computer science and archaeology.

Robust Morphological Analysis

The robust morphological analyzer [9, 20, 26] makes probabilistic morphological classification of ”unknown” word-forms, using an auxiliary dictionary of word-endings. The robust analysis algorithm allows classifying words as well as their inflectional forms, which are not presented in the dictionary. The algorithm is based on the links between the word-form endings and corresponding grammatical information. There are two ways for hypothetical morphological classification of unknown words:

- comparison between the ending of the word-form under consideration and of a pattern word-form (stored in a dictionary with related grammatical information)
- recursive step by step separation of all possible prefixes from the word-form and analysis of the right part of the separation.

In both cases an analyzed word-form obtains the grammatical features of the word stored in the dictionary with maximum number of matching letters belonging to the endings of the compared words. In case a) a model (inverted) dictionary is needed, in case b) procedures for automatic word separation are expected.

The presented approach allows to decrease the volume of the dictionary necessary for the analysis and to perform analysis for unknown (without citation form in the dictionary) or misspelled words.

Automatic Checking of the Syntactic Agreement

The main purpose of theoretical investigation is to formalize the grammatical rules related to the syntactic agreement as well as to find appropriate data structures and effective algorithms for checking the agreement. This is a very difficult task for the Bulgarian language because of much exclusion of the grammatical rules, what supposes wrong results. Furthermore on this stage of our investigation (cause of lack of semantic information) the checking of words agreement is possible only on the basis of the grammatical features of the words.

For every pair of neighbor words [5] in the sentence a checking of right or wrong syntactic agreement between them can be performed. A table of agreement is needed for this purpose. For every cell of the table a list of rules has been defined for right syntactic agreement between related to the cell parts of speech. Filling in the table we have to bear in mind all possible variants of right agreement of two words. For describing the grammatical rules only morpho-syntactic information about the words is available. That means we know what part of speech is every word as well as the grammatical features (gender, number, person) if the word is inflecting. A table is used for checking the right agreement of two words as follows:

- determining what part of speech is the former word we determine the row of the table in which the cell with rules is situated

- determining what part of speech is the latter word we determine the column of the table in which the cell with rules is situated.

Automatic Building of Morphological Processors

LING is an universal morphological processor for inflectional languages. It can be customized for any inflectional language (it was used for Bulgarian and Russian up to now, but it may also be used with Polish, Czech, German and so on). LING has tools for building, editing and viewing morphological dictionaries. The system has universal morphological processor unit, which can make analysis of any inflectional language, which vocabulary is entered in morphological dictionary. The system is used in the following way: First – morphological dictionary is built for some language. Second – texts in this language are analyzed with the universal morphological processor.

The system automatically classifies the word forming types of paradigms and generalizes in table the word-forming rules in the language. In this way the system automatically builds model of the morphology of the entered language. It can be used for linguistic researches, as learning tool or as a morphological processor, where it is necessary.

LING was used as a learning tool in the seminars in computer lexicography. As example, short Russian dictionaries were created in the classes in computational linguistics. They are available for further development.

Linguistic Processor Based on Many Sorted Algebraic Specifications

In our further investigation the problem of building up a natural–linguistic interface to different user–systems with a synthesis of solutions has been discussed. It has been found that it is possible the building up of a common approach to the problem treated (so-called methodology and software system for Object Modeling and Intellectual Resolving – OMIR [1, 16, 28]). It is based on so called automated algebraic specifications, in the base of which the notions of many sorted algebra, conceptual modeling and a synthesis of solutions of functional–computing models has been considered. Differently to many other languages and systems proposed for specification, OMIR already has been used in some real applications. The object-oriented tool system OMIR is designed for constructing and solving conceptual models of subject domains (SD). The system OMIR offers a further stage in the development of methods for building of user object-oriented systems and has the following specific features:

- hierarchical organization (about inheritance) for algebraic specifications and concepts (classes) in SD
- derivation of programs from conceptual SD models
- automatic synthesis of solutions upon a functional-semantic set representing problems in the SD, etc.

Some experiments [28] with conceptual descriptions of the lexis, morphology, syntax and semantics of natural language, (incl. the basic algebraic structures, the properties of the operations of the particular signatures and the methods of planning the solutions on linguistic functional models, etc.) have been made.

Automated Acquisition of Grammars

The automated acquisition of grammars is very important in the practical construction of program language components such as systems in the artificial intelligence where there exist enhanced requirements for the dialogue. It was proposed [3] an approach for automated construction of unification grammars on the basis of sample sentences with assigned grouping and sub-sentence characteristics of the words. The methods of selecting examples leading to grammars adequate to the language are indicated. The approach is applicable for automated revealing of arbitrary inter-sentence relations. We accept it as linguistic facts that:

- the possibility a fixed position to be occupied by some lexeme is entirely defined only by some of the lexeme categories which we shall call type qualifying (TQ);
- in the Bulgarian (as well as in many other languages, for instance in Russian, in English, in French, etc.) TQ almost always is limited to the category *part of speech*. In a small number of exceptions TQ subsume also specific for the various cases lexical meanings;
- if some relation between sentence units is not compulsory then a sentence must exist where the quoted relationship is violated.

Automated Extraction of Word Semantics

The semantic analysis of a text is a difficult stage in the computational linguistic. The main problem of semantic analysis remains how to define word semantics. It is proposed and tested a method for automatic extraction of word semantics through text analysis. The method presupposes description of government models of different verbs.

“Understanding” presupposes the representation of the meaning of the text through a mathematical model, which can be processed by the computer algorithm. We created and tested method for automatic extraction of semantics through text analysis. The main method is described in [20]. Every word has its own model of government, which describes the syntax, and the semantics of the context of the word. As a starting point for the analysis of the word semantics we take the verb as a main component – predicate of the sentence. The method presupposes description of government models of different verbs: In every government model several components participate (syntactic-semantic clusters) – positions which the verb opens around himself. Structure of every position is described in the table as follows.

Syntactic role	Preposition, conj.	Part of speech	Semantic metaconcept	Semantic role	Alternatives: “/” continue: “+” end of model: “**.**”

Every verb has one or more models of government. If the models of government of a given verb are known, when word (phrase) with unknown semantics is found in the sentence, in which this verb is present, the semantics of the word may be defined as the word (phrase) is unified with semantics of position, which it takes in the model. In this way the word is connected with definite metaconcept. Semantic metaconcept is a class of concepts with common characteristics. Examples of semantic metaconcepts are: âðåìå (time), ìÿñòî (place), ëèöå (person), ÷óâñòâî (feeling), íà÷èí (way).. Some semantic metaconcepts are specific for definite sphere of the science. When in the text is identified a described verb, all the phrases and words, which take part in the verb phrase are connected with metaconcepts according to their position in the verb phrase. For example: “Ìàðèÿ å â ñòàÿòà” (“Mary is in the room”).

According to the models of government of “ñúì” (be), “Mary” is identified as a “person”, and “the room” as a “place”. In this example metaconcepts, which characterize the words are “person” and “place”.

The *software implementation* of the algorithm [28] is a program, which enters models of government of verbs from the user. The program analyses text, given on its input as a text file. The program automatically performs: sentence splitting, tokenisation, segmentation of complex sentences into simple ones, morphological analysis with computer dictionary of Bulgarian language and procedures of approximate morphological analysis, morphological disambiguation through grammatical heuristics, NP extraction with simple NP grammar.

Anaphora Resolution

Anaphora resolution is performed after pre-processing of every sentence with the previously mentioned tools. Automatic anaphora resolution in ARES is implementation of Professor R. Mitkov’s multilingual approach to pronoun resolution, which is already implemented and tested for English, Polish and Arabic. This method is adapted for Bulgarian language in the widespread genre of software technical texts and manuals. The approach used is based on the following sequence of operations: first, text is pre-processed through linguistic processing tools. After that third person personal pronouns are identified. In the same sentence and two sentences back noun phrases, which agree in gender and number with the pronoun are sought for. These phrases are called *candidates for antecedents*. Every candidate for antecedent is evaluated through number of indicators, which assign it score. The score from different indicators are summed up. The candidate with highest score is considered antecedent of the anaphora. Indicators consider the salience of every candidate for antecedent, similarity of its context and the context of anaphora, if the candidate is term /a list of software terms was extracted semi-automatically/. The most appropriate candidate is chosen and marked as antecedent of the pronoun. The precision of anaphora resolution experiments [17] is unexpectedly high: 85% in experiment with 107 anaphors in software manuals. Result, which is near the precision of the anaphora resolution in English.

Resources and Tools

Morphological dictionar*y* contains 67500 entries (or more than 1000000 word-forms) divided into 231 inflectional types (proper nouns incl.), morpho-syntactic information for each entry, and a morphological processor (MS DOS and WINDOWS platform) for morphological analysis and synthesis. The developed *Bulgarian morphological processor* (Bulmorph) performs morphological analysis (including robust analysis) and generation on MS-DOS platform (protected mode). The Bulmorph is being distributed by ELDA (European Language resources Distribution Agency). The ELRA Catalogue is available on addresses http://www.pu.acad.bg/dcs/lingua1.htm and http://rdesc.pu.acad.bg/dcs/lingua1.htm .

The ***Integrated Linguistic Environment*** (ILE) provides an important support of the researchers in the field of Computational Linguistics [7, 16, 18]. It speeds up the automatic grammar knowledge acquisition from natural language texts. ILE contains:

- morphological dictionary shell
- software tools for automatic observation of corpora of Bulgarian text; morphological analysis and generation; robust analysis of unknown words

- a number of machine dictionaries and thesaurus with quick access to them (rhyme, abbreviations, morphemes etc.)
- text editor in Bulgarian (spell-checking, detecting some wrong syntactic constructions, etc.)

Development of a ***Dynamic Link Library*** (LIBMORF.dll) containing grammatical functions is a very useful tool for different Windows applications. Grammatical functions can be used very convenient by Microsoft office (trough the Basic macros), Borland Delphi, Harlequin Lisp and all applications with dynamic link opportunities. The 32 bits dynamic library Libmorf.dll (Delphi 2.0 realization, working under Windows 95/NT) contains a morphological processor for Bulgarian and manipulates a morphological lexicon. A program (WB macro) for morphological analysis in Word 97 is realised which gives opportunities to the user to search words with definite grammatical features in a WinWord 7.0 document [12]. This macro uses a dynamic link library LIBMORF.dll.

The system for ***remote lexical database*** is situated on a WEB server [24]. The remote access to the system is via the convenient common gateway interface (CGI). It allows sending a request to the system (working under WINDOWS '95), and receiving the result through the network. The main advantage of this approach is the opportunity for multiple access of many users to the lexical resources and the possibility for substantial test of the system. The very important tool, which the network allows, is the immediate user feedback in case of problems and errors. In this manner a short time test of the system and the lexical database is feasible. On the other hand an extension of the system with missing user-requested applications is realizable The main functions of the system are:

- morphological analysis of user's input sentence in Bulgarian
- checking of the syntactic agreement of two Bulgarian words
- generation of the morphological paradigm of an arbitrary word-form
- lexical database queries, etc..

The user's platform requirements are IBM/PC compatible computer, WINDOWS and WWW browser with Cyrillic fonts. Access through other platforms (Macintosh, UNIX) is still impossible cause of different Cyrillic code tables of the platforms.

Pre-processing tools for Bulgarian include part of speech tagger, tokeniser (to split the text into tokens (every word, number, shortening or sequence of Latin words), sentence splitter (to split the text into sentences) and paragraph splitter, procedure for clause extraction, noun phrase grammar, procedures for anaphora resolution, procedure for heading identification, etc. Although these tools are created for pre-processing of technical manuals, POS tagger, tokeniser, sentence splitter, paragraph splitter and clause chunker can be used without any adaptation with every text. Noun phrase grammar and heading identification may be adapted to be used with genre independent texts. Every computer text analysis should be preceded by these tasks, or at least by some of them. The tools analyze the text structure through a set of SGML tags, which are similar to the HTML tags and prepare the text for further processing. A number of end-of-sentence rules are implemented. In these rules punctuation marks, capital Latin and Cyrillic letters, numbers and shortenings are considered.

The text is divided into sentences through its juxtaposing with different patterns for sentence boundary. The program for semantic extraction uses 50 end of sentence patterns. For more precise end of sentence identification were created:

- a dictionary of shortenings in Bulgarian

- **paragraph splitter** – finds paragraph boundaries, using the indent of the first line in the paragraph, or the distance between the last line in the previous paragraph and the right margin.
- **section heading extractor** /created for technical texts / – this tool indicate heading of chapters and subheading. It is apt for technical texts. This tool is useful, when in technical text key-words have to be extracted.
- **clause chunker** – the clause chunker splits the complex sentences into simple ones. It doesn't use syntactical rules, but simple heuristics, which assume that boundaries of clauses are conjunctions and commas.
- **morphological disambiguator and POS tagger –** we use morphological processor. In the output of the processor, ambiguous words are sought and the ambiguity is solved.

The precision of the sentence extraction is 94% .In order to make the sentence splitting more precise we will use dictionary of shortenings [20]. The program uses many heuristics and grammar rules for disambiguation. The precision of the part of speech tagger is 91%.

Morphological disambiguation is performed through grammar rules and heuristics, which take into account the context of the word and choose only one morphological hypothesis. Example for such a heuristic (specific for Bulgarian):

If *X* is *an homonymwith two hypothesises adverb* or *adjective*
and *after X adjective or noun appears, which is not neutral gender or its number is plural*
then *X* is *adverb*

POS tagger is constantly developed and enhanced with new rules of solving morphological ambiguity. The NP extraction of simple NP (not including PP) is satisfactory. However parsing of complex NP with PP is ambiguous and the precision is not so high. As a whole, NP extraction is in stage of development, but it may be used for linguistic processing.

The module for noun phrase extraction uses the following grammar:

NP → *((AdjP) Qu) (AdjP) Np' (Prep NP)N*
NP' → *N | Np и (and) Np| (Np,)N Np и (and) Np,*

where *N* is noun, *Ad*j - adjective, *Qu* - quantifier: "many", "little", "some", numeral.

After part of speech tagging (morphological analysis + morphological disambiguation) every word and NP phrase is represented through attribute structure, describing its morphological characteristics. The attribute structures from one sentence are stored in list of structures. This list is juxtaposed with the models of government of the verb in the sentence. It is possible the list of attribute structures and a model to be unified, if the attribute structures contain attributes, which are equal (or partially equal) to the morphological attributes in corresponding positions in the model. If it is possible for a model of government and list of attributes to be unified – a semantic metaconcept(s) from the corresponding position in the model is given to every word or phrase in the attribute list. The correspondence word or phrase – metaconcept is saved in specialized dictionary and after that outputs in a text file.

The software ***system for extraction of word semantics*** is in experimental stage. The linguistic module of the system will be enhanced with more grammar rules, more heuristics and rules for part of speech tagging, rules for segmentation of the complex sentences. The cognitive module is in experimental stage. We plan to create dictionary of models of government (not only for verbs). Our purpose is to create semantic database of the Bulgarian words through analysis of large text corpora.

Another linguistic software is the ***system for extraction of metaconcepts***. This system uses as input connected text and a model of government of one or more verbs and extracts the semantics of some words in the texts. Up to now this system is in experimental stage.

The previously mentioned procedures are united in the ***ARES-program*** for preprocessing texts and anaphora resolution. The result from the pre-processing is output in text file with SGML markers, through these markers is represented the structure of the text. ARES marks noun phrases, parts of speech, sentence, clause and paragraph boundaries, it also marks anaphora-antecedent pairs in the text on the basis of the anaphora resolution method, implemented in the system.

Perspectives

The ***further investigations*** will be in the following directions:

- development of methodology, data structures and algorithms for modeling language structures and processes
- studying various features of language structures and processes and creating a number of parsers (optimal in rate, memory used and precision)
- setting up of an annotated large Bulgarian text corpora
- designing corpus-based and error-driven methods for constructing natural language parsers
- designing architecture, implementation and evaluation of software system for acquiring natural language parsers by training over corpora or parsed text
- creating software tools on other platforms and operation systems (Macintosh, UNIX), etc.

REFERENCES

1. ***Doneva R.***, *Automatic Construction of Intellectual Object Oriented Environments for Conceptual Modeling*. Ph. D. Thesis, Sofia, 1994 (in Bulgarian).
2. ***Ivanov K., G. Totkov***, *The Linguistic Processor: System for Research the Word Paradigms in Inflective Natural Languages*. Proceedings of the 20th spring conference of the Union of Bulgarian Mathematicians, 254-259, 1991 (in Bulgarian).
3. ***Ivanova P., K. Ivanov, G. Totkov***. *Automated Acquisition of Grammars Representing Inter-sentence Relation*s. Proc. of the XXIV summer school, Sozopol, in B. Cheshankov, M. Todorov (Eds.), “Applications of Mathematics in Engineering”, Inst. of Applied Mathematics and Informatics, Technical University of Sofia, Heron Press, Sofia, 1998, 231-234.
4. ***Krushkov Hr.***, *Automatic Construction of an Auxiliary Dictionary for Robust Morphological Analysi*s. Proceedings of the 21th International Conference ITP'96: Interaction between Intelligent Entities, Plovdiv, 1996, pp. 85-88 (in Bulgarian).
5. ***Krushkov Hr.***, *Automatic Checking of the Syntactic Agreement*. Proceedings of the 1st National conference Inf’94, November 8-10, 1994, Sofia, pp.127-134.
6. ***Krushkov Hr.***, *Automatic Construction of Machine Dictionaries*. Proceedings of the 25th spring conference of the Union of Bulgarian Mathematicians, April 6-9, 1996, Kazanlak, pp.199-204 (in Bulgarian).

7. ***Krushkov Hr***., *Development of Integrated Linguistic Environment*. Proc. of the 20th International Conference ITP'95: Interaction between Intelligent Entities, 16-21 June1995, Plovdiv, p.131.
8. ***Krushkov Hr***. *Methodology and Computational Tools for Aautomatic Construction of Dictionaries (Rhyme Dictionary)*. Proceedings of the 23th spring conference of the Union of Bulgarian Mathematicians, April 1-4, 1994, Stara Zagora, pp. 381-387 (in Bulgarian).
9. ***Krushkov Hr.***, *Modeling and construction of machine dictionaries*. Ph. D. Thesis, Plovdiv, 1997 (in Bulgarian).
10. ***Krushkov Hr., Hr. Tanev, M. Krushkova,***. *Automatic Extraction of a Pattern and Synthesis Rules from the Word Paradigm*. Proceedings of the 7th National Conference "Contemporary tendencies in the Development of the Fundamental and Applied Sciences", 6-7 June 1996, Stara Zagora, Bulgaria, pp. 167-171 (in Bulgarian).
11. ***Krushkov Hr., M. Krushkova***, *Methodology and Computational Tools for Compression and Searching in Machine Dictionaries*. Proceedings of the 23th spring conference of the Union of Bulgarian Mathematicians, April 1-4, 1994, Stara Zagora, pp. 388-394 (in Bulgarian).
12. ***Krushkov Hr.***. *Development of Linguistic Software for WORD 7.0*. Proceedings of the 28th spring conference of the Union of Bulgarian Mathematicians, April 5-8, 1999, Montana, pp.199-203 (in Bulgarian).
13. ***Krushkov Hr., M. Krushkova***, *Automatic Analysis and Synthesis of Complex Verb Forms*, Proc. of the 1st International Conference Automatica and Informatics'2000, Sofia (in print).
14. ***Krushkov Hr., M. Krushkova***, *Automatic Guessing of the Verb Aspect*, Proc. of the 1st Intern. Conf. Automatica and Informatics'2000, Sofia (in print)
15. ***Krushkova M., G. Popova, Hr. Krushkov,*** *Automatic Classification of the Proper Nouns*. Proceedings of the 7th National Conference "Contemporary tendencies in the development of the fundamental and applied sciences. 6-7 June 1996, Stara Zagora, Bulgaria, pp. 162-166 (in Bulgarian).
16. ***Petrova P., G. Totkov, K. Ivanov***, *Syntactic Analyser*. Proceedings of the 20th spring conference of the Union of Bulgarian Mathematicians, 1991, pp. 341-345 (in Bulgarian). 1991.
17. ***Tanev Ch.***, *Computerised Text Processing and Anaphora Resolution for Bulgarian.*.Papers from Third Conference on FASSBL , Plovdiv September 1999, Trondheim Working Papers in Linguistics., pp.339-347.
18. ***Tanev Ch. Hr. Krushkov***, *Language Processing Tools for Bulgarian*. ACIDCA2000, Corpora and Natural Language Processing, March 2000, Monastir, Tunisia, pp.221-227
19. ***Totkov G***., *Formalisation of Bulgarian Language and the Development of a Linguistic Processor*. Universite de Plovdiv, Travaux scientifiques, Mathematique, vol.26, fasc.3, 1988, pp. 301-311 (in Bulgarian).

20. ***Totkov G.***, *Robust Methods for Automatized Analysis of Bulgarian Texts and the Development of a Linguistics Processor*. Proceedings of the 19th spring conference of the Union of Bulgarian Mathematicians, 1990, pp. 295-303 (in Bulgarian).
21. ***Totkov G.***, *The Development of a Linguistic Processor: problems, results, future*. Proceedings of the 20th spring conference of the Union of Bulgarian Mathematicians, 1991, pp. 43-50 (in Bulgarian).
22. ***Totkov G.***, ***E. Geourgieva, G. Daskalov***, *Automatised Analyzer of Derivation in Bulgarian and Construction of Computerised Ddictionary*. (1991, unpublished).
23. ***Totkov G., Hr. Krushkov, M. Krushkova***, *Formalisation of Bulgarian Language and the Development of a Linguistic Processor (morphology)*. Universite de Plovdiv, Travaux scientifiques, Mathematique, vol.26, fasc.3, 1988, pp. 301-311 (in Bulgarian).
24. ***Totkov G., Hr. Krushkov, Sv. Enkov***, *Remote Lexical Database Access*. Proceedings of International Summer School "Information Technologies in Social Sciences and Humanities, Bourgas, 1997.
25. ***Totkov G., Hr. Krushkov***, *Mathematical Modeling and Constructing of Machine Dictionaries*. Automatica & Informatics, issue 5/6 1997, pp. 59-62 (in Bulgarian).
26. ***Totkov G., Hr. Krushkov***, *Robust Morphological Analysis for Bulgarian Tests*. International Conference “Intelligent management systems”, Sept.’89, Varna, pp. 141-147 (in Russian).
27. ***Totkov G., R. Doneva, K. Ivanov***, OMIR-LING: *A Linguistic Processor Based on Many Sorted Algebraic Specifications*. Intern. Conf. on Mathematical Linguistics ICML'93, Taragona (Catalonia, Spain), Mar 30-31, 1993, pp. 13-14.
28. ***Totkov G., Cr. Tanev***, *Compuiterised Extraction of Word Semantic through Connected Text Analysis*. in A. Narin’iyani (eds.), Computational Linguistics and its Applications, Proc. of the International Workshop DIALOGUE’99, pp. 360-365.

The University of Plovdiv
24, Tzar Assen St.
4000 Plovdiv, Bulgaria
tel./fax: + 359 32 268 636
e-mail: totkov@pu.acad.bg