NLP MODULE FOR BULGARIAN TEXT PROCESSING

Stoyan Cherecharov,[1] Hristo Krushkov,[2] Mariana Krushkova[3]

Abstract: The wide use of web-based information systems and a lack of highly skilled developers are the primary motivation to search for methods and approaches to optimize the building of such systems. This paper describes a model for creating web-based information systems by using a core of reusable, independent, and installable base modules. Such a system is easily adapted to a client's needs and is extendable by adding specific modules that interact with the remainder of the system by following certain rules. The approach allows flexible and rapid development of applications for small to extremely large web-based systems, simply by adding modules with adequate functionality. The growing demand of Bulgarian customers for such systems is the reason for building a base module for automatic processing of Bulgarian text. This paper presents a module that performs automatic morphological analysis and synthesis, verifies syntactic agreement, automatically places stress, and processes complex verb forms, among other functions. The described functionality can be integrated with other modules using a suitable interface.

UDC Classification: *004.4`2*; **DOI:** http://dx.doi.org/10.12955/cbup.v5.1080

Keywords: natural language processing, computational linguistics, software modules, web-based systems

Introduction

In recent years, there has been an increased interest for wide use of web-based systems (Prokofyeva & Boltunova, 2017; Yang et al., 2016). The high customer requirements and the lack of highly skilled and experienced developers are primary motivations for seeking methods and approaches to optimize the building of such systems. Using similar functionality allows a system to be built with reusable, independent, and installable modules. The systems can be built rapidly without compromising their quality. For this purpose, developers use the principles of the Aspect-Oriented and Modular programming. The installation can be performed using the standard package depended upon by managers of technology.

For example, one can use the log mechanism to track exceptions. An analysis of different web-based systems reveals the exceptions are managed by code snippets. This aspect of the functionality is spread across the application with different parts involved in more than one role and elements strongly coupled. That is, the various components have more than one primary role, and hence, the functionality does not conform to principles of strong cohesion, loose coupling, and single responsibility. One way to remedy this nonconformity is to extract and encapsulate each functionality into a separate module with a single role. This approach would mean the modules having a single assignment would be independent of the system for recording exceptions (a log system).

Another example is the authorization system that controls access to system resources by different roles or users. Such functionality can be encapsulated in a separate module where the system resources are independent of the controlled access. The method of a dynamic and parallel slicing algorithm that is context-sensitive for distributed Aspect-Oriented Programming (AOP; Singh et al., 2017) can be used. However, there are certain code pitfalls in AOP and analyses that need to be avoided (Santos et al., 2016). Certain technology with agnostic programming techniques for modular programming (Toulson & Wilmshurst, 2017) can be used during the separation of the functionality into modules.

There have been increased problems with the use of the Bulgarian language, from the students in recent years. Errors are evident in the speeches of the politicians and journalists in the mass media. Some universities overcome this problem by organizing Bulgarian language courses during the first year of the education. The abilities of the Information Technology (IT) in this type of education are still not used extensively. According to Blagoeva et al. (2011), the support for Bulgarian language text analysis is fragmentary, while for English it is sound, and for Dutch, French, German, Italian, and Spanish moderate. The state of speech and text resources is the same for cited European languages, although languages with moderate support include Czech, Hungarian, Polish, and Swedish.

There are also some fragmentary publications related to the automatic processing of the Bulgarian language. Different problems emerge with different approaches. The formal models comprise two main types: rule-based and statistical. Rule-based models are traditional and are a result of research by

[1] Plovdiv University Paisii Hilendarski, Faculty of Mathematics and Informatics, cheresharov@ihahockey.com
[2] Plovdiv University Paisii Hilendarski, Faculty of Mathematics and Informatics, hdk@uni-plovdiv.bg
[3] Plovdiv University Paisii Hilendarski, Faculty of Mathematics and Informatics, mik@uni-plovdiv.bg

linguists. The creation of tagged text corpora allows for statistical methods to apply over large amounts of data. The development of IT enables statistical models to improve and extend the rule-based models with new rules derived empirically. Rule-based Part Of Speech (POS) tagging as a part of a machine translation system is presented in Jackov (2015).

Regression models are used for fine-grained sentiment analysis in Bulgarian movie reviews (Kapukaranov & Nakov, 2015). Statistical methods provide sound practical achievements, while rule-based ones give students appropriate linguistic knowledge, and are thus, better for educational purpose.

Applying the methods of Active Learning, in which the student is the center of the educational process, is an important step in the direction of increasing its quality. The gamification is one form of such education.

Different techniques from the game industry, such as earning points, badges and medals, levels, a list of the leaders, avatars, and virtual currency are used, to increase the motivation of the students. A scientific report, dedicated to this topic, reveals that these techniques are used mostly in the informatics and IT education (Dicheva et al., 2015). Educational computer games that support utilization of Bulgarian morphology are described in Krushkov et al. (2015).

The conclusion of the research indicates the positive effect the gamification has on the students, by increasing their motivation, activity, commitment, and success. The usage of only traditional methods in the education of the high school and university students is monotonous and demotivating, especially for the ‘Z-generation’ (Internet generation). This generation (born after 1994) accepts the virtual world as the normal environment and cannot imagine their education without information technology (IT). The number of the students, graduates, and prospective students is decreasing. In such an environment, the courses of teachers and schools that are attractive to students are those using innovative educational methods based on IT. Hence, the main aim of this study is to create a natural language processing (NLP) module for Bulgarian text processing that is an integral part of every web-based system.

Data and Methodology

Analyses showed that the base modules necessary for rapid web-system development are: authentication module, log module, authorization module using access control list, navigation module using the authorization, session module, and a content management module.

The system we have built was open to extensions. Different modules with specific functionality can be added following certain rules. This approach allows flexible, rapid application and development of small to extremely large web-based systems, by only adding modules with adequate functionality. It allows programming with event and control loops, which further augments the usage of the modular programming (Ricci, 2016).

The modules can be distributed and separated in the web space. The communication between these is based on RESTful Web application programming interface (API) over Hypertext Transfer Protocol (HTTP) or the secure hypertext version (HTTPS). The modules can be built using different programming languages and technologies.

To expand the scope of the systems, built with the modules, we started to add highly abstract modules with general purpose using different mathematical and linguistic formalisms. Such a module that helps to solve numerous problems is the Workflow Engine Module based on Petri Nets theory. It augments the rest of the modules and creates another abstraction layer. Different types of concurrent processes can be modeled with the Petri Nets Theory and implemented without writing programming code.

Another module is based on the formal model of the Bulgarian language grammar. It allows Natural Language Processing. It gives extra abilities for searching and proofreading of Bulgarian text in the web-based applications. It is extremely useful for building educational systems.

The Bulgarian language belongs to the group of inflected languages. The NLP module is based on the formal model described in (Krushkov, 1997). In this model, 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. A part of speech is a set of classes. Every set can be divided into subsets depending on criteria pertaining to this particular part of speech.

For example, the set of nouns includes the classes with machine numbers 1–75. There are four subsets depending on the gender as follows: with machine numbers 1-40: masculine, 41-53: feminine, 54-73:

neuter, 74–75: only plural. 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. For example, in the paradigm of the adjectives, word-form num. 1 has grammatical features (masc., sing.); word-form num. 2 has grammatical features (plural); etc. For all parts of speech word-form num. 1 is the base (citation) form. The model is appropriate for all inflected languages. It is extended for proper nouns (Krushkov, 2001).

A morphological processor is a tool performing automatic morphological analysis and synthesis. An approach for automatic morphological generation and analysis is investigated, based on the presented classification. For every word, a 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. For example, the pattern of the word 'верен' (faithful) is 'в*р*н'. All other words from the same inflectional type (84) 'тесен' (narrow), 'бесен' (mad), 'десен' (right), with the following patterns: 'т*с*н', 'б*с*н', 'д*с*н', have two changing letters (last two vowels) in the pattern. The rules for the word-form generation are of two types:

1) Replacing the '*' with a letter (including the empty one, ''); and
2) Appending the endings.

The paradigm of the word 'верен' according to the rules for type 84 is shown in Table 1.

Table 1: The paradigm of the word 'верен' according to the rules for type 84

Number/word form	Grammatical features	Rules for type 84
1. верен	masc., sing.	*/'е'; */'е'
2. верни	pl. / extended form	*/'е'; */''+'и'
3. вярна	fem., sing.	*/'я'; */''+'а'
4. вярно	neut., sing.	*/'я'; */''+'о'
5. верния	masc., sing., full def.art.	*/'е'; */''+'и'+'я'
6. верният	masc., sing., short def.art.	*/'е'; */''+'и'+'ят'
7. верните	pl., def.art.	*/'е'; */''+'и'+'те'
8. вярната	fem., def.art.	*/'я'; */''+'а'+'та'
9. вярното	neut., def.art.	*/'я'; */''+'о'+'то'

masc. = masculine; fem. = feminine; neut. = neuter; sing. = singular; def. art. = definite article

Source: Authors

For other adjective types (types numbered from 76 to 89) a column with respective rules is prepared. The rules for a member of some inflectional type are the same for all other members of this type.

The purpose of the automatic morphological analysis is to perform automatically 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. A machine dictionary consists of (word-pattern, inflectional type number) entries. 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. Based on 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. For every word, an inflectional type number (t) and a word-form number (g) can be extracted.

For checking the agreement between two words, the following sequence was obtained: t_1, g_1, t_2, g_2, where t_1 and t_2 were the inflectional type numbers of the words; g_1, g_2 were the word-form numbers of the words. For example, the analysis of 'вярната жена' (the faithful woman) produces the sequence 84, 8, 41, 1, which means that:

- The former word is an adjective (t_1=84), feminine, singular, definite article (g_1=8);
- The latter word is a noun, feminine (t_2=41), singular (g_2=1).

The agreement of words is right if there exists a row in the table of agreement tb_{fw}, te_{fw}, g_{fw}, b_{sw}, te_{sw}, g_{sw}, where $tb_{fw} \leq t_1 \leq te_{fw}$ and $g_1 = g_{fw}$ and $tb_{sw} \leq t_2 \leq tb_{sw}$ and $g_2 = g_{sw}$ gives true.

In our example, the 5-th row of Table 2 agrees with the sequence 84, 8, 41, and 1:

$$76 \leq 84 \leq 89 \text{ and } 8 = 8 \text{ and } 41 \leq 41 \leq 53 \text{ and } 1 = 1$$

If such a row does not exist hypotheses for the right agreement are building up:

1. if $tb_{fw} \leq t_1 \leq te_{fw}$ and $g_1 = g_{fw}$ and $tb_{sw} \leq t_2 \leq tb_{sw}$ and $g_2 \neq g_{sw}$, then g_2 is wrong. It has to obtain the value of g_{sw}; and
2. if $tb_{fw} \leq t_1 \leq te_{fw}$ and $g_1 \neq g_{fw}$ and $tb_{sw} \leq t_2 \leq tb_{sw}$ and $g_2 = g_{sw}$, then g_1 is wrong. It has to obtain the value of g_{fw}.

For example, the analysis of 'верен жена' (present Google translation of 'faithful woman') produces the sequence 84, 1, 41, and 1, which means that the former word is an adjective (t_1=84), masculine, singular (g_1=1), and the latter is a noun, feminine (t_2 = 41), singular (g_2 = 1). These two words are not in agreement.

There is no row in the table of agreement giving true for the sequence 84, 1, 41, and 1. However, there are two rows which can produce hypotheses:

- the 4-th row ($76 \leq 84 \leq 89$ and $1 \neq 3$ and $41 \leq 41 \leq 53$ and 1=1) gives the sequence of right agreement 84 3 41 1 (вярна жена – ‘faithful woman’)
- the 5-th row $76 \leq 84 \leq 89$ and $1 \neq 8$ and $41 \leq 41 \leq 53$ and 1=1 gives the sequence of right agreement 84 8 41 1 (вярната жена – ‘the faithful woman’)

Table 2: A part of the table related to the agreement between adjective and noun

First word	Second word	tb_{fw}	te_{fw}	g_{fw}	tb_{sw}	te_{sw}	g_{sw}
masc., sing.	masc., sing.	76	89	1	1	40	1
masc., sing. short def. art.	masc., sing.	76	89	5	1	40	1
masc., sing. full def. art.	masc., sing.	76	89	6	1	40	1
fem., sing.	fem., sing.	76	89	3	41	53	1
fem., sing. def. art.	fem., sing.	76	89	8	41	53	1
neut., sing.	neut., sing.	76	89	4	54	73	1
neut., sing. def. art.	neut., sing.	76	89	9	54	73	1
plural	plural	76	89	2	1	75	4
plural def. art.	plural	76	89	7	1	75	4

masc. = masculine; fem. = feminine; neut. = neuter; sing. = singular; def. art. = definite article. Source: Authors

The practical research leads to the conclusion that, it is better to build another table. This is a table of the syntactic disagreement. In the table, there are some variants of incorrect agreement of adjacent words. This is due to the inability for each two adjacent words to determine if the agreement is correct or incorrect without syntactic analysis. The second table is built for the adjacent words, which certainly cannot reach an agreement (e.g., preposition and verb).

A phonetic classification is performed to determine the stress position of all words of the paradigm. To every dictionary word, a vector is attached. The length of the vector coincides with the length of the paradigm. The element v[i] of the vector shows the relative position (in vowels) of the stress of the *i*-th word of the paradigm according to the stress position of the base form. That is why the first element of every vector is 0: v[1] = 0. If the stress remains constant for all the words in the paradigm, the vector consists of zeros c = (0, 0, 0, …, 0). For example, the vector assigned to the word 'враг' (enemy) is f = (0, l, l, 2, 2). The paradigm of this word is 'вра̀г' (enemy), 'врагà' (the enemy – short def. art.), 'врагъ̀т' (the enemy – full def. art.), 'врагов̀е' (enemies), 'врагов̀ете' (the enemies).

A formal model of complex verb forms, as well as an algorithm for automatic analysis and synthesis of these forms, is also investigated.

Conclusion

The NLP module performs automatic morphological analysis and synthesis, verification of syntactic agreement, automatically placing the stress, automatic processing of complex verb forms. These functionalities allow searching in Bulgarian texts for all word forms of a given word. It is possible to search for words, which belong to a specific part of the speech or have specific grammatical features, as well as to replace the word forms of a given word with the word forms of another word if they belong to the same part of the speech. If the replacement words are nouns, where it is appropriate in the adjacent words, the gender of the adjectives is changed. The main dictionary comprises 82 thousand base forms, which can produce over than 1 500 thousand word forms. Dictionary lookup allows one to retrieve words with specific grammatical and morpho-syntactic features. It is possible to extract words from the same morphological class, with the same type of stress mobility and more. The module can work together with the Content Management Module, Authorization Module and the Workflow Engine Module creating and managing complex workflow processes for proofreading, text analyzes, editing and more. Creating an article in the Content Management system could trigger proofreading and text analyzes in the NLP module. Depending on the results of the analyzes work items can be instantiated in the Workflow Engine Module for different roles and actors. The work items represent tasks to be completed by the roles and actors in the workflow process. The Petri Nets theory is used to create a model of the process. So, the NLP module adds a unique functionality to the system. The module will be used for building a web-based system for teaching the Bulgarian language. The main functionalities will help teachers to construct computer-aided e-lessons according to their wishes selecting appropriate lexical and grammatical material. Much of the options of the system could be done automatically: extracting sentences and vocabulary of selected text material, analyzing the number of the new words, generating tests and exercises similar to those described for English (Malinova & Rahneva, 2016).

Acknowledgements

This work was partially supported by the FP17-FMI-008 project of the Scientific Fund of the Plovdiv University Paisii Hilendarski, Bulgaria.

References

Blagoeva, D., Koeva, S., & Murdarov, V. (2011). *The Bulgarian Language in the Digital Age.* White Paper Series: Springer.

Dicheva, D., C, D., A. G., & G., A. (2015). Gamification in Education: A Systematic Mapping Study. *Educational Technology & Society, 18*(3), 75 – 88.

Jackov, L. (2015). Feature-rich part-of-speech tagging using deep syntactic and semantic analysis. *International Conference Recent Advances in Natural Language Processing* (pp. 173-180). Hissar: Association for Computational Linguistics (ACL).

Kapukaranov, B., & Nakov, P. (2015). Fine-grained sentiment analysis for movie reviews in Bulgarian. *International Conference Recent Advances in Natural Language Processing* (pp. 266-274). Hissar: Association for Computational Linguistics (ACL).

Krushkov, H. (1997). *Modelling and Building Machine Dictionaries and Morphological Processors (in Bulgarian).* Plovdiv: Ph.D. thesis, University of Plovdiv, Faculty of Mathematics and Informatics.

Krushkov, H. (2001, February). Automatic Morphological Processing of Bulgarian Proper Nouns. *Traitement Automatique des Langues, 41*(3), 709-726.

Krushkov, H., Atanasova, M., & Krushkova, M. (2015). Teaching Bulgarian through Games (in Bulgarian). *Annual Journal of Education and Technologies, 6*, 322 – 329.

Malinova, A., & Rahneva, O. (2016). Automatic generation of english language test questions using mathematica. *Conference: CBU International Conference on Innovations in Science and Education (CBUIC)* (pp. 906-909). Prague: Central Bohemia Univ, Unicorn College.

Prokofyeva, N., & Boltunova, V. (2017). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. *Procedia Computer Science, 104,* 51-56.

Ricci, A. (April 2016). Programming with event loops and control loops – From actors to agents. *Computer Languages, Systems & Structures, 45*, 80–104.

Santos, A., Alves, P., Figueiredo, E., & Ferrari, F. (1 April 2016). Avoiding code pitfalls in Aspect-Oriented Programming. *Science of Computer Programming, Volume 119*, 31–50.

Singh, J., Khilar, P. M., & Mohapatra, D. P. (May 2017). Dynamic slicing of distributed Aspect-Oriented Programs: A context-sensitive approach. *Computer Standards & Interfaces, 52,* 71–84.

Toulson, R., & Wilmshurst, T. (2017). Further Programming Techniques. In R. Toulson, & T. Wilmshurst, *Fast and Effective Embedded Systems Design (Second Edition)* (pp. 111-134). Elsevier Ltd.

Yang, W., Lee, S.-H., Zhu Jin, Y., & Hwang, H.-T. (2016, October). Development of web-based collaborative framework for the simulation of embedded systems. *Journal of Computational Design and Engineering, 3*(4), 363–369.