https://doi.org/10.53656/math2022-3-2-con

Educational Matters
Научно-методически статии

CONVERTING NUMERAL TEXT IN BULGARIAN INTO DIGIT NUMBER USING GATE

Dr. Nadezhda Borisova, Assist. Prof.,
Dr. Elena Karashtranova, Assoc. Prof.
South-West University "Neofit Rilski" – Blagoevgrad (Bulgaria)

Abstract. The Internet serves billions of users providing a variety of information resources whereby a lot of the information is presented in natural human language and needs an efficient approach to be processed.

Natural language processing (NLP) refers to the ability of computers to analyze and understand the structure of human language. By utilizing NLP this linguistic knowledge is transformed into algorithms for solving specific problems. GATE is widely used, open-source software infrastructure that provides a framework and components for solving NLP tasks. The available GATE tools can be adapted to other languages and text processing tasks.

This article will present an approach for converting numeric data, written as words in Bulgarian, into digit numbers. For this case, a relevant configuration file for Bulgarian has been integrated into the general tool set in the open source software for natural language processing GATE. The aim of this survey is to determine the exact numeric value of Bulgarian text numeric data, which can be used as a starting point for producing more complex annotations, such as monetary measurement units, etc.

Keywords: Natural language processing; Bulgarian grammar; GATE

1. Introduction

The vision of Tim Berners-Lee for the Semantic Web is to expand the existing World Wide Web through standards which aim to provide machine-readable data. Therefore, web technology is to retrieve the information via meanings with no difference in the ways and languages presented (Berners-Lee et al. 2001).

Natural language processing (NLP) is an area of research in computer science that explores interaction between computers and spoken or written human language. The effectiveness of the applied NLP techniques supports the automation of the process of adding semantics to the existing web data. Information Extraction (IE), as a form of analysis of natural language, is a process of obtaining structured information from raw texts. Named Entity Recognition (NER) is one of the basic sub-tasks in the process of IEand in (Cunningham

2005) it is presented as the most reliable technological method for IE. NER is mainly used for obtaining typical named entities such as names of places, organisations or people. This task is slightly dependent on the thematic field, and the effectiveness of the results can reach 95%. The task of entities extraction also includes the identification of monetary amounts and percentages, times, dates, numeric data, etc., which are not primarily presented as digit numbers (digits). Following the standardized spelling rules, numeric data can be presented as the mixing of numbers and words ("The distance is *2 thousand* meters*"*), as well as only in words ("*Nine thousand* undergraduates participated in the marathon").

Bulgarian literary language accepts writing numbers as words if they are at the beginning of a sentence or in cases when the word (phrase) does not represent an exact number. In addition, numbers from one to ten, when they do not signify a measurement, are written only in words (not as numbers) in the text. When data is presented as numbers containing five or more digits, an interval is placed before each group of three digits, except for the cases when the digits denote a corresponding number. ("The jackpot in the game will go over 8 400 000 leva!").

Numbers in text are crucial in numerous fields, for instance, in scientific articles, statistics, economics, etc. The present paper focuses on an approach for converting numeric data from text in Bulgarian. The ability for correct identification and work with numbers has significant impact for NLP tasks such as Information Extraction and Information Retrieval.

2. Related Work

Current state of language technology for Bulgarian language is discussed in (Koeva & Stefanova 2022). The conducted survey is based on resources collected and distributed by the „European Language Grid" (EU project 2019-2022) (ELG). The report claims that there are no developed natural language text processing methods for Bulgarian. Such methods concern the man – computer interaction, the simultaneous processing of text, image or video. With other methods, what is visible is the technological advancement, yet there are no applications which are suitable for broader use, such as the automatic resuming of document content, for example. Authors register "a significant difference" in the development of technological language methods for English as well as some other European languages, and the Bulgarian language. The overcoming of this difference is further complicated due to the specificity of the Bulgarian language being a linguistic "challenge" as an inflected language (with a Cyrillic alphabet).

G. Hristova (2021) presents an overview of the progress in the field of text analysis in Bulgarian in two directions – availability of language resources and

practical applications. The first focus of the review reveals that not all existing language resources are available and easy to implement. Despite admitting the progress of biomedical NLP for Bulgarian, the second focus summarizes that there is a scarce number of research articles related to applications of text analysis on practical problems.

In previous works, the task of named entity recognition in Bulgarian is presented in (Georgiev et al. 2009). The proposed approach recognizes entities in news text and categorizes them as persons, organizations, locations and miscellaneous. The study uses Conditional Random Fields, a widely used modeling technique for NLP tasks, and is the first to implement statistical approach for NER in Bulgarian.

The research of numeracy in NLP for various languages indicates that it is still at an early stage. The latest advances and work on numeracy in NLP are summarized in (Thawani et al. 2021). The authors have synthesized the best practices for representing numbers in text and have divided the numeric tasks into seven subtasks. The first NLP-centric taxonomy of numeracy tasks and of number representations is provided as a result.

Wallace et al. (2019) and Naik et al. (2019) have explored embedding methods, number decoding and addition tasks. The results confirm that the common embedding techniques successfully capture magnitude (e.g. 3<4), but are unsatisfactory when the numeric mapping of string to numeric value is concerned ("three" → 3).

Alkhateeb et al. (2016) presents an approach for converting written Arabic numeral text into a digit number and vice versa. The proposed approach is implemented in an Android mobile-based application for children. In a previous work (Al-Taani et al. 2009) a numeral checker application for the Arabic language implemented as a Finite State Transducer is proposed. The application transforms a number into the corresponding word format with respect to the gender agreement feature.

In (Uskov et al. 2019), a method for extracting numeric data from texts in Russian has been proposed. The method employs semantic networks and semantic frames to determine the boundaries and to extract the numeric data from the text.

Regarding Bulgarian language processing by the General Architecture for Text Engineering (GATE), an earlier work (Borisova et al. 2013) has demonstrated the functionality of GATE to perform regular expressions over annotations for detecting noun-adjective agreement errors. The provided code samples have been used for the detection and retrieval of word groups meeting a specific set of criteria.

Borisova (2015) has presented an approach for ontology based Information extraction from unstructured text in Bulgarian. Consequently, a diction-

ary-based lemmatizer for Bulgarian has been developed and integrated for improving the results of the part-of-speech (POS) tagger (Iliev et al. 2015).

Bulgarian Plugin in GATE provides a processing resource which integrates the BulStem inflectional stemmer for Bulgarian (Nakov 2003). The basic aim of stemming is to retrieve the root of a given word.

At the time this paper was written, we were not aware of the existence of other processing resources in GATE specifically targeting the Bulgarian language.

In this paper, we will focus on annotating numbers represented as words in Bulgarian text with the open source software GATE. The presented approach covers specifics in Bulgarian and annotates numbers written as words or numbers (or a combination of both), and determines their numeric value.

In (Atanasova 2019), the teaching knowledge management module (TKM) that is part of the university knowledge management system is presented and described. The TKM module that contains tools for the evaluation of learning resources in distance e-learning has a significant feature – it can be extended. The approach for annotating numbers represented as words in Bulgarian texts with the open source software GATE could be used as a part of intelligent software tools for assessing the quality and the correctness of the curriculum, syllabus, and learning resources.

3. Proposed Approach

GATE is free open source project widely used in the field of language processing. The infrastructure of the framework gives the opportunity to use and implement components for different language processing tasks.

GATE supports various document formats, including plain text, HTML, XML, RTF, SGML, Email, PDF (some documents), etc. Additional formats (for example, Twitter-style JSON format) are provided by relevant plugins. The supported documents have to be encoded in one of the standards accepted by Java. The most popular character encoding nowadays is UTF-8, which is the most widely used way to represent Unicode text (GATE 2022).

The algorithm for converting numbers from text format should work with natural text in Bulgarian. For this purpose, in addition to the standard components for text analysis, a configuration file for Bulgarian has been integrated in the GATE pipeline. The rest of this section describes the steps performed to detect numeric data from Bulgarian text.

3.1. Tokenization

In the phase of the tokenization of a document, the boundaries of tokens are identified. As a result, two sets of annotations are generated, namely "Token" and "Space Token". Each token is classified as a word, punctuation, symbol or number. A token can be classified as Number if it contains only combinations

of consecutive digits. The intervals present in the text are identified by means of set of annotations "SpaceToken", and for each number (consisting of five or more digits) that has an interval placed before each group of three digits, the tokenization defines each group as a separate token. For instance, the results for the numbers „3 586 571"and „48.7%"are demonstrated with an example below (Figure 1).

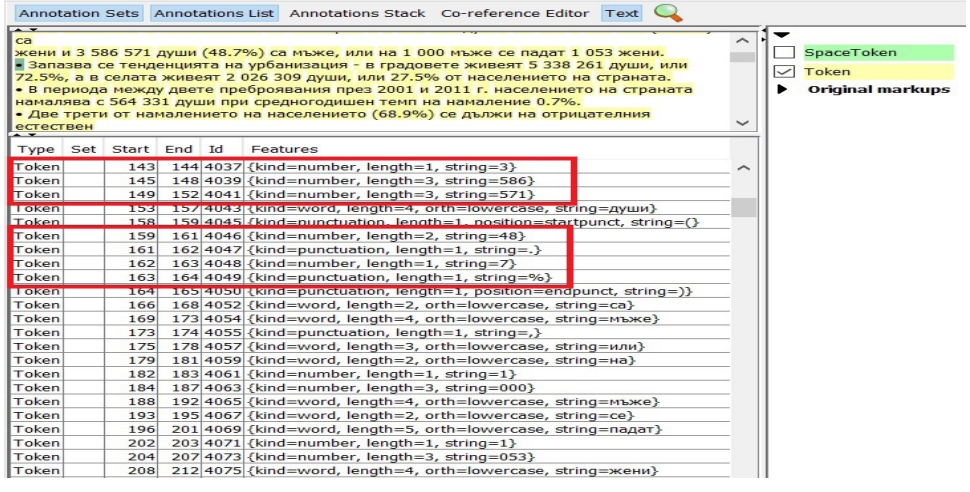

Figure 1. Tokenization of numbers in a document

We can obtain similar result when the numeric value is expressed in words in the text. The example in Figure 2 shows that each numeric value is written in words, with a punctuation mark – a full stop – used as an interval between them. After the tokenization of the document, 332 tokens in 103 sentences have been annotated and the value of the feature *kind* for all of them is equal to „word" or „punctuation". In fact, 229 tokens with numeric data were treated as separate "words"!

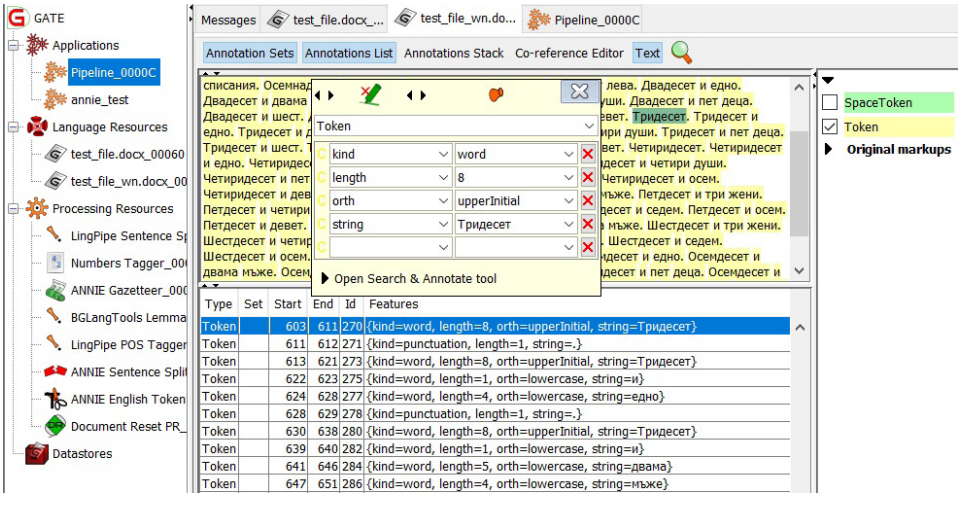

Figure 2. Tokenization of numbers written as words in a document

The work of the tokenizer is only limited to the identification of tokens in a text. The application of adaptable grammatical rules with the consecutive text processing gives the opportunity for achieving flexibility and effectiveness. (GATE 2022).

3.2 Sentence detection

The GATE sentence splitter is a cascade of finite-state transducers and this module is required for the Part-of-Speech (POS) tagger. The splitter segments the text into sentences, based on a set of rules and each sentence is annotated with the type "Sentence".

3.3. Part-of-Speech Tagger

The standard subtask (after identifying the tokens and sentences in the document) is the marking of separate tokens with their corresponding part of speech. In order to do that, LingPipe POS Tagger has been used. It presents two models for Bulgarian – Bulgarian-full.model and Bulgarian-simplified. model, executed on a transformed version of the BulTreeBank-DP (Simov et al. 2004). As a result of the implementation of the tagger, a feature *category* is added to each token in the annotation (for each corresponding part of speech). In this particular case, each numeric value written in words in the text has been annotated as *Mc-pi* (cardinal number; Plural; non-definite). For example, the set of features for token „Шест" (Six) is: {category= Mc-pi, kind=word, length=4, orth=upperInitial, string=Шест}

3.4 Number tagging

The Tagger_Numbers creole repository affords the following processing resources (PRs) for annotation of numbers within documents:

– The "Numbers Tagger" annotates numbers, which are written in numbers or numeric words;
– The "Roman Numerals Tagger" annotates Roman numerals.

Both PRs produce *Number* annotations with standard feature *value*, which stores the actual value of the number that has been annotated as a Double. This feature allows the application of the created annotations for more complex annotations. In this case, a "Numbers Tagger" as a subject of this article has been used independently.

In order to create an instance of the "Numbers Tagger" what is necessary is the configuration of the following parameters upon initialization: 1) the URL and the encoding of the configuration file and 2) the URL of the JAPE grammar used for post-processing. The configuration files currently provide support for English, French, German and Spanish. There is a file for each of the mentioned languages, whereby the following items have been assigned:

1) **words** (one word), which can be used as numerals;
2) words which are used as **multipliers** (for hundreds, thousands) and
3) **conjunctions**, which are used to combine sequences of words defining numerals.

In Bulgarian, the numeral is a part of speech signifying numbers and numeric features. In linguistic reference books, it is related to the notion of quantity, number, cardinals and ordinals (Boyadzhiev 2004).

Following their structure, numerals can be subdivided into simple, complex and compound. *Simple* numerals have a root morpheme and are simple words: three (три), five (пет), ten (десет), hundred (сто), thousand (хиляда), etc. *Complex* numerals are formed from the combination of two simple numerals in a compound word: twelve (дванадесет), seventeen (седемнадесет), two hundred (двеста), five hundred (петстотин), etc. *Compound* numerals are formed my means of combining two or more numerals which preserve their characteristics of separate words, yet they notify one number: twenty-three (двадесет и три), nine hundred and eighty-four (деветстотин осемдесет и четири), etc.

According to the rules for the formation of quantitative numerals in Bulgarian, a configuration file has been integrated. The structure of the file has its own specificity that defines a decimal symbol and a digit grouping symbol, words, multipliers and conjunctions as follows:

– for a decimal symbol a decimal point is used and for a digit grouping symbol – white space
– **words** that can be used as numbers (Table1) – simple numerals which are non-derivative forms and cannot be formed from other words. These quantitative numerals are basic, since they denote the numbers from the decimal ordinal system.

Table 1. Simple numerals (basic)

Simple numerals (basic)	
Digit	Word
1	Един (м.р.), една (ж.р.), едно (ср.р.)
2	Два (м.р.), две (ж.р. и ср.р.)
3	Три
4	Четири
5	Пет
6	Шест
7	Седем
8	Осем
9	Девет
10	Десет
20 – 90	Двадесет, ..., деветдесет
100	Сто
1 000	Хиляда
1 000 000	Милион

According to their gender, the only numerals that vary are *един* for masculine, *една* for feminine, *едно* for neuter gender, as well as *два* for masculine, *две* for feminine and neuter gender.

The other quantitative numerals are formed from the simple numerals by means of multiplication or addition of derivative words (Gramatika 1983).

– **Multipliers** for quantitative numerals, formed by multiplication (Table 2)

Multiplication is a method used with homogeneous numeral categories (tens (десетици), hundreds (стотици), thousands (хиляди), millions (милиони)). "**Four** multiplied by *a hundred* is **four** *hundred* 4*100=400".

To determine their numeric value, a multiplicand (which denotes a simple numeral) is multiplied by a multiplier (raised by 10^{th} power). The following possible multipliers have been assigned: 10 to 2^{nd} power for the "hundreds", 10 to 3^{rd} power for the "thousands" and 10 to 6^{th} power for the "millions" – for example, five hundred ($5*10^2$) and 6 000 (six thousand, $6*10^3$). The multiplier 10^2 has been assigned for the suffixes „*ста*" and „*стотин"*.The multiplier 10^3 has been assigned for *thousands (хиляди)*, and 10^6 for *millions (милиони)*.

The rule for the formation of tens (десетици) (from 20 to 90 with a multiplier 10^1) cannot be applied to this example, since "ten" (10) is not a possible word and a multiplier.

Table 2. Quantitative numerals formed by multiplication

Quantitative numerals formed by multiplication		
Hundreds		
Digit	Word	Quantitative numerals
200	Двеста	*„Две“* * ***„ста“*** ($2*10^2$)
300	Триста	*„Три“* * ***„ста“*** ($3*10^2$)
400	Четиристотин	*Четири****сто*****тин** ($4*10^2$)
500	Петстотин	*Пет****сто*****тин** ($5*10^2$)
600	Шестстотин	*Шест****сто*****тин** ($6*10^2$)
700	Седемстотин	*Седем****сто*****тин** ($7*10^2$)
800	Осемстотин	*Осем****сто*****тин** ($8*10^2$)
900	Деветстотин	*Девет****сто*****тин** ($9*10^2$)
Thousands		
2 000	Две хиляди	*„Две“* * ***„хиляди“*** ($2*10^3$)
3 000	Три хиляди	*„Три“* * ***„хиляди“*** ($2*10^3$)
.....		
100 000	Сто хиляди	*„Сто“* * ***„хиляди“*** ($100*10^3$)
200 000	Двеста хиляди	*„Две“* * ***„ста“*** * ***„хиляди“*** ($2*10^2*10^3$)
.....		
900 000	Деветстотин хиляди	*„Девет“* * ***„стотин“*** * ***„хиляди“*** ($9*10^2*10^3$)

– **Conjunctions** for quantitative numerals which are formed by addition (Table 3)

The defined *conjunctions* are “и” (and) (for hundreds and thousands) and “на” (to) (for tens). The first one is defined as a whole word and requires white space on both sides, e.g. „сто *и* девет“ (one hundred *and* nine). The second one is not defined as a whole word and it is used for numerals formed by combining two simple numerals into one compound word. For instance, the numerals from 12 to 19 are formed when the lower class is pre-positioned to the upper and they are joined with the preposition “на” (to) – 17 седем***на***десет (seventeen) (7-на-10).

Table 3. Quantitative numerals formed by addition

Quantitative numerals formed by addition		
Tens		
Digit	Word	Quantitative numerals
12	Дванадесет	„Два“ + „на“ + „десет“ (2+10)
13	Тринадесет	„Три“ + „на“ + „десет“ (3+10)
14	Четиринадесет	„Четири“ + „на“ + „десет“ (4+10)
15	Петнадесет	„Два“ + „на“ + „десет“ (5+10)

16	Шестнадесет	„Шест“ + „на“ + „десет“ (6+10)
17	Седемнадесет	„Седем“ + „на“ + „десет“ (7+10)
18	Осемнадесет	„Осем“ + „на“ + „десет“ (2+10)
19	Деветнадесет	„Девет“ + „на“ + „десет“ (2+10)
21 – 29	Двадесет и едно, ..., Двадесет и девет	„Двадесет“ + „едно“ (20+1), ...,,, Двадесет“ + „девет“ (20+9)
31 – 31	Тридесет и едно, ..., Тридесет и девет	„Тридесет“ + „едно“ (30+1), ..., „Тридесет“ + „девет“ (30+9)
41 – 49	Четиридесет и едно, ..., Четиридесет и девет	„Четиридесет“ + „едно“ (40+1), ..., „Четиридесет“ + „девет“ (40+9)
51 – 59	Петдесет и едно, ..., Петдесет и девет	„Петдесет“ + „едно“ (50+1), ..., „Петдесет“ + „девет“ (50+9)
61 – 69	Шестдесет и едно, ..., Шестдесет и девет	„Шестдесет“ + „едно“ (60+1), ..., „Шестдесет“ + „девет“ (60+9)
71 – 79	Седемдесет и едно, ..., Седемдесет и девет	„Седемдесет“ + „едно“ (70+1), ..., „Седемдесет“ + „девет“ (70+9)
81 – 89	Осемдесет и едно, ..., Осемдесет и девет	„Осемдесет“ + „едно“ (80+1), ..., „Осемдесет“ + „девет“ (80+9)
91 – 99	Деветдесет и едно, ..., Деветдесет и девет	„Деветдесет“ + „едно“ (90+1), ..., „Деветдесет“ + „девет“ (90+9)
Hundreds		
101 – 199	Сто и един, ..., Сто девет-десет и девет	„Сто“ + „един“ (100+1), ..., „Сто“ + „деветде-сет“ + „девет“ (100+90+9)
201 – 299	Двеста и един, ..., Двеста деветдесет и девет	„Две“ * „ста“ + „един“ (2*102+1), ..., „Две“ * „ста“ + „деветдесет“ + „девет“ (2*102+90+9)
301 – 399	Триста и един, ..., Триста деветдесет и девет	„Три“ * „ста“ + „един“ (3*102+1), ..., „Три“ * „ста“ + „деветдесет“ + „девет“ (3*102+90+9)
.....	.....	.....
901 – 999	Деветстотин и един, ..., Деветстотин деветдесет и девет	„Девет“ * „стотин“ + „един“ (9*102+1), ..., „Деветстотин“ + „Деветдесет“ + „девет“ (9*102+90+9)
Thousands		
1001 – 1999	Две хиляди и един, ..., Две хиляди деветстотин деветдесет и девет	„Хиляда“ + „един“ (1000+1), ..., „Хиляда“ + „девет“ * „стотин“ + „деветдесет“ + „девет“ (1000+9*102+90+9)
2001 – 2999	Две хиляди и един, ..., Две хиляди деветстотин деветдесет и девет	„Две“ *„хиляди“ + „един“ (2*103+1), ..., „Две“ * „хиляди“ + „девет“ * „стотин“ + „де-ветдесет“ + „девет“ (2*103+9*102+90+9)
.....	.....	.....

Millions		
1 000 000	Един милион	„милион“ (1000000)
2 000 000	Два милиона	„Два“ *„милиона“ (2*106)
.....	.....	.....

4. Results

A set of one hundred and three numerals in Bulgarian (Table 4) was used in the GATE pipeline to test the results of the proposed approach. The words are written with no spelling mistakes and their combinations comply with the rules for the formation of quantitative numerals. The numerals included in the set of sentences are ones that vary according to gender – masculine, feminine and neuter gender.

Table 4. Results of the GATE pipeline

N	Number in Bulgarian words	Digit Number	Correct
1	Един	1	✓
2	Една	1	✓
3	Едно	1	✓
4	Два	2	✓
5	Две	2	✓
6	Три	3	✓
7	Трима	3	✓
8	Четири	4	✓
9	Пет	5	✓
10	Шест	6	✓
11	Седем	7	✓
12	Осем	8	✓
13	Девет	9	✓
14	Десет	10	✓
15	Единадесет	11	✓
16	Дванадесет	12	✓
17	Тринадесет	13	✓
18	Четиринадесет	14	✓
19	Петнадесет	15	✓
20	Шестнадесет	16	✓
21	Седемнадесет	17	✓
22	Осемнадесет	18	✓
23	Деветнадесет	19	✓
24	Двадесет	20	✓
25 - 33	Двадесет и едно – Двадесет и девет	21, …, 29	✓

34	Тридесет	30	✓
35 – 43	Тридесет и едно – Тридесет и девет	31, …, 39	✓
44	Четиридесет	40	✓
45 – 53	Четиридесет и едно – Четиридесет и девет	41, …, 49	✓
54	Петдесет	50	✓
55 – 63	Петдесет и едно – Петдесет и девет	51, …, 59	✓
64	Шестдесет	60	✓
65 – 73	Шестдесет и едно – Шестдесет и девет	61, …, 69	✓
74	Седемдесет	70	✓
75	Осемдесет	80	✓
76 – 84	Осемдесет и едно – Осемдесет и девет	81, …, 89	✓
85	Деветдесет	90	✓
86	Сто	100	✓
87	Двеста	200	✓
88	Триста	300	✓
89	Четиристотин	400	✓
90	Петстотин	500	✓
91	Шестстотин	600	✓
92	Седемстотин	700	✓
93	Осемстотин	800	✓
94	Деветстотин	900	✓
95	Деветстотин тридесет и три	933	✓
96	Хиляда	1000	✓
97	Две хиляди	2000	✓
98	Три хиляди	3000	✓
99	Шестнадесет хиляди и три	16 003	✓
100	Сто хиляди	100 000	✓
101	Деветстотин осемдесет и седем хиляди шестстотин петдесет и четири	987 654	✓
102	Един милион	1 000 000	✓
103	Два милиона	2 000 000	✓

The generated *Number* annotations cover all numerals with standard feature *value*, which has stored the actual value of the number that has been annotated as a Double (Figure 3).

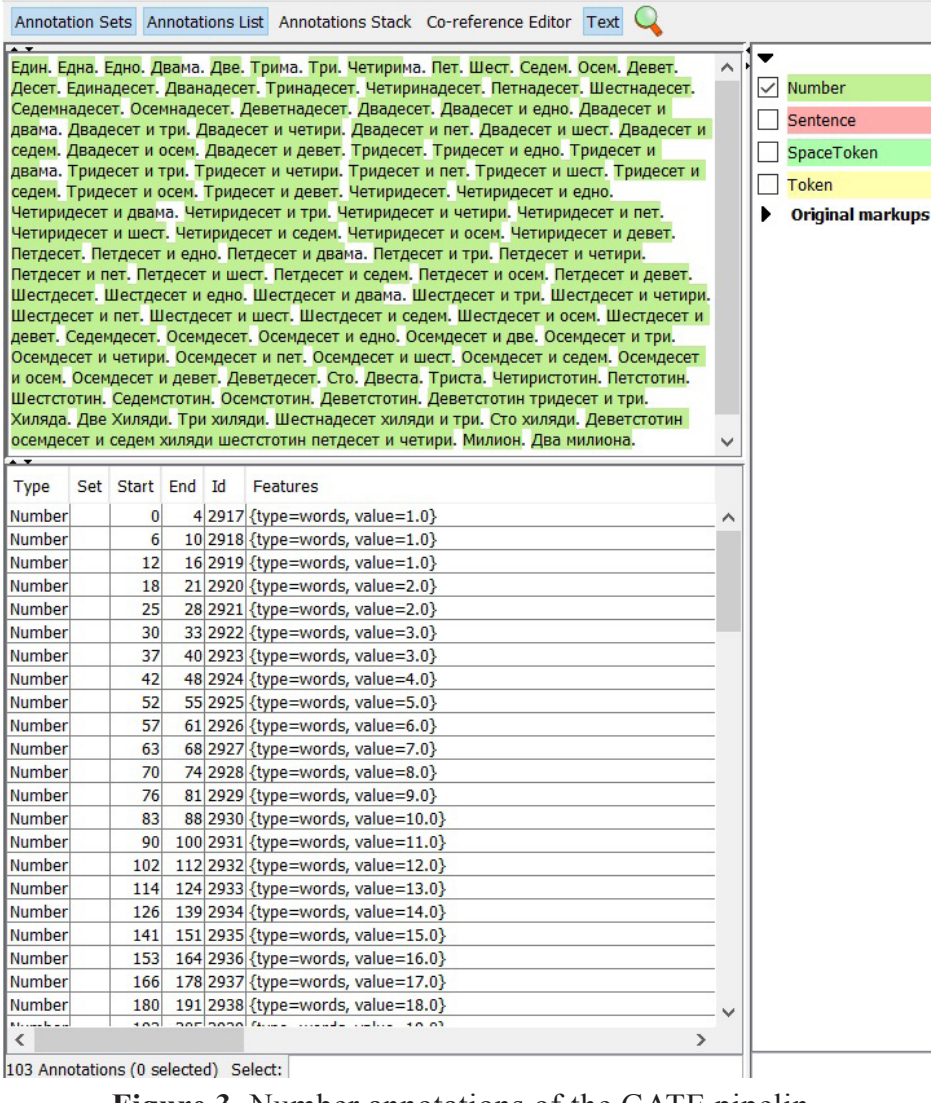

Figure 3. Number annotations of the GATE pipeline

5. Conclusion

The development of web technologies aims to extract information through its meaning regardless of the ways and languages in which it is presented. In a great number of domains, a significant part of the information includes numeric data presented in various formats.

This paper provides an approach for annotating numbers represented as words in Bulgarian texts with the open source software GATE. The approach follows specificities of the Bulgarian language and annotates numbers by determining their numeric value. The algorithm is aimed to work with natural text in Bulgarian. For this purpose, in addition to the standard components for text analysis in GATE, a configuration file for Bulgarian has been integrated. As a result, the generated *Number* annotations covered all numerals with standard feature *value*, which has stored the actual value of the number annotated as Double.

In future, we plan to extend the functionalities of the presented approach for producing more complex annotations, called numeric characteristics extraction.

Acknowledgments. This work was supported by Project No. RP-A4/22 – "Data mining of students' behavior in distance e-learning systems: case study BLACKBOARD", South-West University "Neofit Rilski" Blagoevgrad, Bulgaria.

REFERENCES

ALKHATEEB, F., BARHOUSH, M. & AL-ABDALLAH, E., 2016. Developing a System for Converting a Numeral Text into a Digit Number: Abacus Application. *Journal of Intelligent Systems*, **25** (4), 611 – 628. Available from: doi.org/10.1515/jisys-2015-0029.

AL-TAANI, A. T., WEDIAN, S. A. & DARWISH, O. M., 2009. Arabic numerals checker: checking agreement between numerals and counted objects in the Arabic language, *Int. J. Comput. Process. Lang.* **22**(2009), 341 – 357.

ATANASOVA, I., A, 2019. University Knowledge Management Tool for the Evaluation of the Efficiency and Quality of Learning Resources in Distance e-Learning, *International Journal of Knowledge Management*, **15**(4), October-December 2019, ISSN: 1548-0666, EISSN: 1548-0658, DOI: 10.4018/IJKM, IGI Global. https://www.igi-global.com/article/a-university-knowledge-management-tool-for-the-evaluation-of-the-efficiency-and-quality-of-learning-resources-in-distance-e-learning/234740.

BERNERS-LEE, T., HENDLER, J. & LASSILA, O., 2001. The Semantic Web. *Scientific American Magazine.*

BORISOVA, N., ILIEV, G. & KARASHTRANOVA, E., 2013. On Detecting Noun-Adjective Agreement Errors in Bulgarian Language Using GATE. *Proceedings of the Fifth International Conference of FMNS.* Blagoevgrad, 180 – 187.

BORISOVA, N., 2015. An approach for ontology based information extraction. *Information Technologies and Control.* **1**, 15 – 20.

BOYADZHIEV, T., KUTSAROV, I. & PENCHEV, Y., 2004. *Suvremenen bulgarski ezik. Fonetika. Leksikologiya*. Morfologiya. Sintaksis. Sofia. ISBN: 954-321-070-5.

CUNNINGHAM, H., 2005. Information extraction, automatic. *Encyclopedia of language and linguistics*, **3**(8), 10.

GEORGIEV, G., NAKOV, P., GANCHEV, K., OSENOVA, P. & SIMOV, K., 2009. Feature-Rich Named Entity Recognition for Bulgarian Using Conditional Random Fields. – *In: Proc. of International Conference RANLP-2009*, September 2009, 113 – 117.

GATE, 2022: https://gate.ac.uk/sale/tao/.

Gramatika na savremenniya balgarski knizhoven ezik, 1983, Tom II. Morfologia. Sofia: Izdatelstvo na BAN.

HRISTOVA, G., 2021. Text Analytics in Bulgarian: An Overview and Future Directions. *Cybernetics and Information Technologies*, **21** (3), 3 – 23. Available from: doi.org/10.2478/cait-2021-0027.

ILIEV, G., BORISOVA, N., KARASHTRANOVA, E. & KOSTADINOVA, D., 2015. A Publicly Available Cross-Platform Lemmatizer for Bulgarian. *Proceedings of the Sixth International Scientific Conference – SWU, FMNS 2015*. Blagoevgrad, 147 – 151.

KARASHTRANOVA, E., ILIEV, G., BORISOVA, N., CHANKOVA, Y. & ATANASOVA, I., 2015. Evaluation of the Accuracy of the BGLemmatizer, *Proceedings of the Sixth International Scientific Conference – SWU, FMNS 2015*. Blagoevgrad, 152 – 156.

KOEVA, S. & STEFANOVA, V., 2022. *Report on the Bulgarian Language*, https://european-language-equality.eu/wpcontent/uploads/2022/03/ELE___Deliverable_D1_5__Language_Report_Bulgarian_.pdf.

NAIK, A., RAVICHANDER, A., ROSE, C. & HOVY, E., 2019. Exploring Numeracy in Word Embeddings. *In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3374 – 3380, Florence, Italy. Association for Computational Linguistics.

NAKOV, P., 2003. Building an inflectional stemmer for Bulgarian. *In: Proceedings of the 4th international conference on Computer systems and technologies e-Learning - CompSysTech '03.*

SIMOV, K., OSENOVA, P., SIMOV, A. & KOUYLEKOV, M., 2004. Design and implementation of the Bulgarian HPSG-based treebank. *Journal of Research on Language and Computation*, **2**(4), 495 – 522, December 2004.

THAWANI, A., PUJARA, J., SZEKELY, P.A. & ILIEVSKI, F., 2021. Representing numbers in NLP: a survey and a vision. *arXiv preprint arXiv:2103.13136*.

USKOV, I.N., YARKEEV, A. & TSOPA, E., 2019. Named Numeric Characteristics Extraction from Text Data in Russian. *In: CEUR WORKSHOP PROCEEDINGS "MICSECS 2019 – Proceedings of the 11th Majorov International Conference on Software Engineering and Computer Systems" 2020*.

WALLACE, E., WANG, Y., LI, S., SINGH, S. & GARDNER, M., 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. *In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5307 – 5315, Hong Kong, China. Association for Computational Linguistics.

✉ **Dr. Nadezhda Borisova, Assist. Prof.**
Researcher ID: F-4210-2014

✉ **Dr. Elena Karashtranova, Assoc. Prof.**
Department of Informatics
South-West University "Neofit Rilski"
66, Ivan Mihailov Blvd.
2700 Blagoevgrad, Bulgaria
E-mail: nborisova@swu.bg
E-mail: helen@swu.bg