The Framework for Hypothesis Verification and Analysis of Natural Language Processing for the Russian Language

Ekaterina V. Politsyna[1][0000−0002−9313−4766], Sergey A. Politsyn[1][0000−0002−0744−6035], and Alexander S. Porechny[1]

Moscow Aviation Institute (National Research University), 125993, Russia, Moscow, Volokolamskoe shosse, 4, kathrin.beaver@mail.ru

Abstract. The paper brings together a number of recent works on developing the single extensible programming framework for the Russian language covering all steps of natural language processing(NLP) with minimum requirements to end-users. Issues addressed include the theoretical significance of the framework and its structure. The framework covers first three phases of NLP (graphematic, morphological and syntactic) taking into account that the results on every phase could be uncertain and clarified later. (i.e. homoforms of a verb and a noun have different syntactic relations). The purpose of this framework is to verify ideas of linguists with minimum development time to estimate their viability along with solving practical tasks of NLP in various applications. The last part contains examples of tasks we have solved with this framework and some ideas for further development.

Keywords: NLP framework · programming framework · morphological analysis tools · syntactic analysis tools · java tools.

1 Introduction

Despite the actuality of automatic natural language text processing, the problem of formalizing the natural language is not fully resolved. As a result, it is impossible to "understand" completely a natural text by a computer program. Of course, there are many hypotheses and methods that can solve some problems of text analysis. For example, to remove lexical and morphological ambiguity using the methods of Synan, Trigra and Accopost. The accuracy can reach more than 90% [1], For example, the method implemented in the Russian-English and vise versa phraseological machine translation system RETRANS, which allow to remove the homonymy of words with 99% accuracy. The limitation is a basic set of structures, the extension of which requires manual search and insertion into the system [2, 3].

Thus, the problem of accuracy of NLP entails the creation of various text analysis tools such as Lemmatizer [4], Greeb [5], Stemka [6], pymystem3 [7], TreeTagger [8], etc. as well as many toolsets: AOT [9], GATE [10], LingPipe [11],

UIMA [12], in the commercial segment: Tomita-parser [13], SpeechKit [14] and the other tools from Yandex, Google's core, ABBYY [15]. Such a variety of solutions is stipulated by the complexity of NLP and the lack of simple and unambiguous solutions of computer text analysis problems. To validate new approaches and hypotheses, special tools are needed to create new programs that comprehensively take into account the previous experience accumulated in the field of computer linguistics.

At the same time, advanced technologies in a various fields increasingly demands the application of automatic text analysis tools in many software systems. Various tools are used in many commercial media monitoring software products (Interfax SCAN [16], products of Medialogy [17]), trends tracking, antiplagiarism systems (Antiplagiat [18], Rukont [19]) and others. This requires special development tools (libraries and frameworks) that support popular programming languages of industrial software (Java, .NET, etc.) and solve application problems, i.e. are sufficiently high-level in terms of text processing algorithms.

2 Overview of text processing tools

The majority of new hypotheses and ideas of NLP need preliminary verification and fast computer implementation for confirmation or refuse of their viability. Almost all new algorithms often suppose some already conducted preliminary analysis (for example, for semantic analysis at least pre-processing of the text and morphological analysis should already be done).

There are frameworks designed for automatic text analysis (GATE, LingPipe, UIMA) [10–12]; portals containing large data collections and providing tools for its processing (META-SHARE, Lapps Grid, Language Grid, CLARIN) [20, 21]; tools provided in the form of program interfaces (Google, Yandex, ABBYY, RCO [22]).

The British system GATE (General Architecture for Text Engineering) developed more than 15 years ago is actively used for all types of NLP tasks. It is positioned as a tool for developers and researchers in the field of automatic text processing [10]. It supports English, French, German, Spanish, Italian, Chinese, Romanian, Hindi, Arabic, and some other languages. But for the Russian language only the morphological analysis module and the Ontotext repository are implemented. A similar approach is used by the UIMA and the American LingPipe systems.

Apache UIMA (Unstructured Information Management applications) is an architecture and a set of libraries and tools for creating, exploring and using a wide range of different analysis models, as well as integrating them with information retrieval and storage technologies [11]. The open source UIMA framework is an implementation of the open UIMA architecture. UIMA provides handling of input information of any type: text, audio, video, its transformation and transmission in a structured form to storage systems. UIMA supports English, French, German, Italian, Portuguese, Russian, German, Swedish, etc. But the Russian

language is supported only by the keyword extraction module Apache UIMA AlchemyAPI Annotator.

LingPipe is a set of development tools for solving computer linguistics tasks [12]. Its architecture accounts the requirements to the efficiency, speed, scalability and multiple reuse of results, i.e. for industrial use, including unit tests for library frameworks. Arabic, Chinese, Dutch, English, German, Greek, Hindi, Japanese, Portuguese and Spanish are supported, but there is no support of the Russian language.

Therefore, the most popular frameworks (GATE, LingPipe, UIMA) developed for English and other languages are not suitable for the Russian language, not only because of the lack of Russian language support by authors (there are opportunities of adaptation), but often due to algorithms used. Methods or algorithms used for one language are not always applicable or give the same good results for other languages. Thus, for example, the Porter algorithm, which underlies the Snowball tool [23], is well suited to the English language but for the Russian has a relatively low accuracy due to the individuality of different natural languages.

Most commercial products (ABBYY, Yandex), which support the Russian language, are unreasonable to purchase for possible one-time utilization or unconfirmed hypothesis. Some companies (e.g. Yandex) offer trial or limited use of their tools and APIs, but not all vendors and for all tools. Often, commercial products are solid. They do not allow making a superstructure, for example, to add or modify an existing method or simply test a new hypothesis.

The AOT toolset designed for the Russian language, which consists of several related tools (for graphematic, morphological, syntactic and semantic analysis), but it doesn't support all popular software platforms. Primarily, there is no support for the Java language [9]. Of course, there are methods how to use libraries written in another language, however, such an implementation reduces the debugging. In addition, these methods significantly slow down the performance of the final system, in compare with one programming platform for all parts of the project is used.

There is one general drawback in application of individual tools implementing one of the NLP stages: most of the tools are scattered, the processing results of one have to be further transformed and adapted for the other tool. For example, there are many tools for text parsing (Lemmatizer, Greeb, etc.) and morphological analysis (Stemka, pymystem3, TreeTagger, etc.), but in order to unite them into one toolset it is necessary to learn each of the selected tools and then write a program for converting the results to "understandable" form for reading and processing for the next level tool.

3 Using the framework

There is always a need for new approaches and investigation methods during the research in the field of computer linguistics. Also, developers of various in-

formation systems are also encountered with the need to solve individual tasks of automatic text analysis or use the results of NLP.

Therefore, the problem of rapid hypotheses validation is relevant not only for researchers, but also for developers of industrial systems in which the demands related to NLP are increasingly growing. These requirements are usually not related to the main functionality of the system and should be as much as possible subject to the requirements for the architecture of the system, development time, reliability, scalability, etc., to achieve results acceptable for practical use.

The choice of programming language for the implementation of auxiliary tools is quite free, but the means used in the development of industrial software are regulated by the standards of organizations. As it was mentioned before, Java is the most widely used programming language according to some rating agencies [25, 27].

Thus, there is a need to design a developer-friendly open-source Java-tool for the Russian language, designed to conduct research in the field of linguistics, in which the main task is to make a computer implementation of a hypothesis to verify it as quickly as possible, and to assess the prospects for its further development.

4 The framework for implementing of text analysis algorithms

The framework with a simple program interface was developed implementing three stages of NLP: graphematic, morphological and semantic-syntactic.

Figure 1 shows the structure of the framework. Each module is a separate standalone library that can be replaced by a third-party library with similar functionality without affecting the other modules.

Graphematic analysis stage is implemented in the Parser module which is based on regular expressions engine. The example of the module Parser provides the possibility of replacing the module with a third-party without interfering with the work of other tools. To do this a standard Java approach is used: you need to create your own class that implements the GamaParser interface and required methods, and then use it as an input parameter when initializing the framework or part of it.

Morphological analysis module is implemented as a *JMorfSdk* Java library. It was based on OpenCorpora [28] dictionary, which is one of the variants of the grammatical Russian language dictionary created by A. A. Zaliznyak. The OpenCorpora dictionary is constantly updated and maintained. It contains about 360 thousand lemmas, 5 million word forms, which include quite rare and new words, as well as the most typical typos to reduce their impact on the analysis of the whole text. The library's performance is rather high. The complexity of obtaining the morphological characteristics of the word is O(1), which is achieved through the use of *ConcurrentHashMap* together with bit operations and optimized storage of the most necessary characteristics. The library provides methods for both analysis and word generation, taking into account the

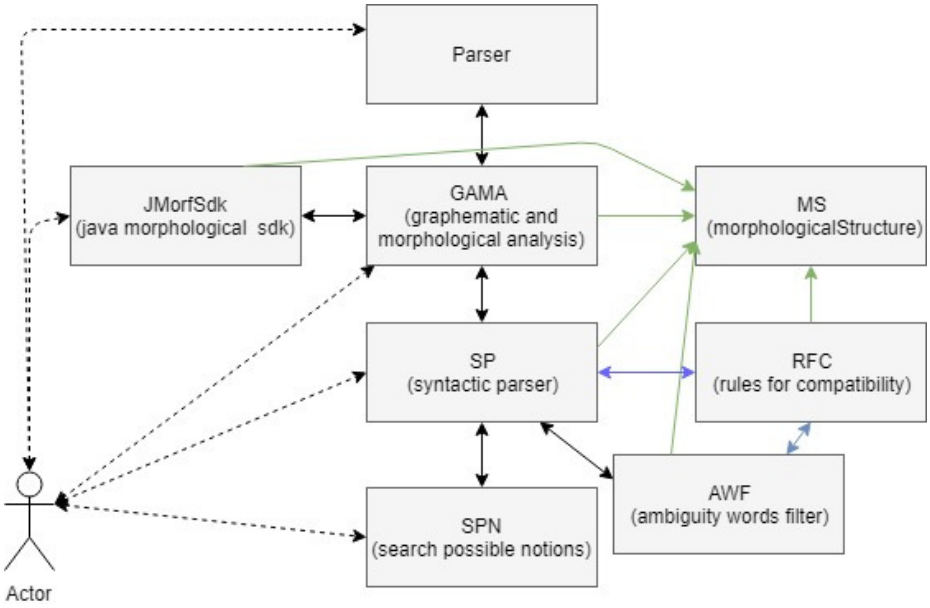

Fig. 1. Structure of the framework.

given morphological characteristics. The library is thread-safe and can be used in multi-threaded applications.

The GAMA module (graphematic and morphological analysis) expresses the idea of a "quick start". The module combines morphological and graphematic analysis of the text and has a simple program interface. User provides raw text and gets a set of homoforms for each word. Thus, an increase in the abstraction level is achieved, which reduces the need to study the API for graphematic and morphological analysis, and also eliminates the time costs for their integration and debugging.

The AWF(ambiguity words filter) module is designed to eliminate ambiguity. The filter is based on the approach that the word in a sentence with a single-valued(unambiguous) word form or exactly defined part of speech determinate a set of possible syntactic structures, which in their turn form a subset of possible parts of speech that can be used with the original words. Thus, part of the word homoforms are filtered and the word becomes unambiguous with high probability [29].

The RFC(rules for compatibility) module contains a set of word compatibility rules, including models of control. These rules were located in a separate module so that they could be expanded and modified without changing the other tools. Various rules and models of syntactic word binding in phrases and sentences are grouped in the module, and some statistical rules are realized.

The module SP(syntactic parser) is a tool for constructing the syntactic structure of a sentence. The raw input is a text coming as a string, GAMA

obtains homoforms, then AWF module is used to eliminate the ambiguity. After that, the tool uses an internal set of rules and RFC to build a syntactic structure. This approach reduces the number of methods to minimum.

In addition to functional modules, the framework includes MS (morphological structures) module. This module provides structures to store words, homoforms with their morphological characteristics, the word order in the sentence, etc. This module helps not to create structures for storing words in various forms during NLP. It also binds modules into a single framework.

The SPN(search possible notions) module is a tool for identifying concepts and creating a list of candidates for concepts (in terms introduced by G.G. Belonogov) [30]. In addition, there are methods for identifying word phrases from a text that can be used, for example, to search for keywords and key phrases.

5 Using the framework in text analysis tools

The framework is a part of NLP tools set. This set is presented on the web-portal "Automated text analysis" [31], and is used in many NLP services and tools. In addition, it has been successfully applied to the creation of research software for identifying named entities [32], the tool for obtaining full forms of abbreviations [33], the tool for creating a semantic web [34], the application for highlighting key words [35], the tool for identifying key elements of texts and the construction of abstracts [36], and others.

At present time, the framework is also used to develop and test algorithms of the automatic obtaining of synonyms in Russian texts by extracting similar phrases and forms of word combinations and their further analysis using the constructed semantic network [34].

The framework was used for developing a number of software tools and conducting research. First, basing on the framework we created a program for eliminating abbreviations and contractions. It was calculated that this program revealed more than 90% of the contractions and formed the correct wordforms for about 63% of different type contractions. Secondly, a name and entity extraction system has been developed, which showed an accuracy of about 94% for the named entities extraction, with the percentage of the first kind errors being less than 20%, and the percentage of the second kind errors being less than 5%.

Another application of the framework is the comparison between methods for identifying key words from the texts. A set of tools for the keyword extraction research was developed. The conducted research allowed creating an integrated approach which increases keyword extraction accuracy for more than 8% in comparison to the known extraction methods.

Based on the first results obtained using the SPN in extracting concepts [37],we started creating a database of word combinations and concepts that is being filled up after processing and manual approving lists of new phrases and candidates to concept extracted from a text. With the use of the framework, a tool was developed to identify key sentences from texts and obtain a brief presentation of the text. The research is continuing in this area.

The joint use of the framework with the previously developed semantic network based on various dictionaries(synonyms, antonyms, associations) will significantly expand the range of tasks to be solved. For the extension of database of concepts and phrases and work with the semantic network, a web interface and corresponding tools are being implemented to expand the database and search for existing concepts and their connections.

Based on the first results obtained by using the SPN in extracting concepts [37], we started creating a database of word combinations and concepts that is being filled up after processing and manual approving lists of new phrases and candidates to concept extracted from a text. Its joint utilization with the previously developed semantic network will significantly expand the range of tasks to be solved. To append the database of concepts and phrases and work with the semantic network, a web interface and corresponding tools are being implemented to expand the database and search for existing concepts and their connections.

6 Conclusion

The developed framework is a tool for fast implementation of NLP algorithms and testing of hypotheses in computer linguistics. The main task of the framework is to provide an extensible set of libraries that are united by a common architecture that meets the software development requirements and regulations. Due to this, it is possible to use them within a set of programs, as well as to solve the applied problems of computer linguistics.

The framework provides the ease of implementation within existing industrial solutions through the use of Java platform standards: the use of a "typical" object model and exceptions handling, the use of multi-threaded features and the overall modular architecture of the Java platform. The framework is maintaining and developing. It was tested while creating the other NLP applications.

References

1. Sokirko A. V., Toldova S. Y.: Sravnenie ehffektivnosti dvuh metodik snyatiya leksicheskoj i morfologicheskoj neodnoznachnosti dlya russkogo yazyka (skrytaya model' Markova i sintaksicheskij analizator imennyh grupp). [Comparison of the effectiveness of two methods of removing lexical and morphological ambiguity for the Russian language (the hidden Markov model and the syntactic analyzer of name groups)]. Internet Mathematics Russia, Moscow, 80–94 (2005) (In Russian)
2. Belonogov G.G. and others:Avtomatizaciya sostavleniya i vedeniya slovarej dlya sistem frazeologicheskogo mashinnogo perevoda tekstov s russkogo yazyka na anglijskij i s anglijskogo na russkij [Automation of compilation and maintenance of dictionaries for systems of phraseology machine translation from Russian into English and from English into Russian]. Scientific and technical information, Series 2, no. 12, pp. 16–21. VINITI, Moscow (1993) (In Russian).
3. Belonogov G.G. and others:Interaktivnaya sistema russko-anglijskogo i anglo-russkogo mashinnogo perevoda politematicheskikh nauchno-tekhnicheskih tekstov.

[The interactive system of Russian-English and English-Russian machine translation of multi-topic scientific and technical texts] Scientific and technical information, Series 2, no. 12, pp. 21–27. VINITI, Moscow (1993) (In Russian).
4. Lemmatizer official website, https://wiki.de.dariah.eu/display/TextGrid/Lemmatizer. Last accessed 8 Apr 2018.
5. Library graphematic analysis of Russia – Greeb official website, https://github.com/dustalov/greeb. Last accessed 14 Apr 2018.
6. Kovalenko A.: Veroyatnostnyj morfologicheskij analizator russkogo i ukrainskogo yazykov [Probabilistic morphological analyzer of Russian and Ukrainian languages]. System administrator, no 1, pp. 66-75. (2002) (In Russian).
7. pymystem3 official website. https://pypi.python.org/pypi/pymystem3. Last accessed 14 Apr 2018.
8. TreeTagger. Ludwig-Maximilians-Universitt Mnchen (LMU). http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger. Last accessed 8 Apr 2018.
9. AOT official website http://aot.ru Last accessed 8 Apr 2018.
10. GATE – General architecture for text engineering, The University of Sheffield, 1995-2017. https://gate.ac.uk/. Last accessed 15 Apr 2018.
11. LingPipe official website http://alias-i.com Last accessed 15 Apr 2018.
12. Apache UIMA Project http://http://uima.apache.org Last accessed 11 Apr 2018.
13. Yandex Tomita parser official web site https://tech.yandex.ru/tomitaLast accessed 11 Apr 2018.
14. Yandex SpeechKit official web site https://tech.yandex.ru/speechkit Last accessed 11 Apr 2018.
15. ABBY official website https://www.abbyy.com/ru-ru/isearch/compreno Last accessed 8 Apr 2018.
16. "Medialogy" company software products https://www.mlg.ru Last accessed 21 Oct 2017.
17. SCAN - System of comprehensive news analysis by Interfax. https://scan-interfax.ru/ Last accessed 21 Oct 2017.
18. Internet service "Antiplagiat" by "Anti-plagiat" company https://www.antiplagiat.ru Last accessed 22 Oct 2017.
19. "RuContext" system by "RuCont" company http://text.rucont.ru Last accessed 22 Oct 2017.
20. The Language Applications Grid, US National Science Foundation Office of CyberInfrastructure. http://www.lappsgrid.org/ Last accessed 25 Oct 2017.
21. CLARIN - European Research Infrastructure for Language Resources and Technology, CLARIN ERIC. https://www.clarin.eu Last accessed 25 Oct 2017.
22. Products of RCO company. http://www.rco.ru/?page_id=4510 Last accessed 26 Oct 2017.
23. Snowball official website http://snowballstem.org/ Last accessed 13 Apr 2018.
24. Porechny A.S, Politsyna E. V., Politsyn S.A.: Sozdanie krossplatformennoj biblioteki morfologicheskogo analiza dlya russkogo yazyka. [Creation of a cross-platform library of morphological analysis for the Russian language] In: Proceedings of the XVIII International Scientific Conference "Computer science: problems, methodology, technology", pp. 51–56. VSU, Voronezh(2018) (In Russian).
25. Official site Tiobe Index. https://www.tiobe.com/tiobe-index/ Last accessed 15 Feb 2018.
26. Official site PYPL. http://pypl.github.io/PYPL.html Last accessed 15 Feb 2018.
27. Portal Statistic Times. http://statisticstimes.com/tech/top-computer-languages.php Last accessed 15 Feb 2018.

28. OpenCorpora official website . http://opencorpora.org/ Last accessed 15 Feb 2018.
29. Porechny A.S.: Razrabotka metoda razresheniya neodnoznachnostej pri semantiko-sintaksicheskom analize teksta na estestvennom yazyke [Development of a method for resolving ambiguities during semantic and syntactic analysis of a text in the natural language] In: Proceedings of "Gagarin's Readings" XLIII – International Youth Scientific Conference p. 1121. MAI, Moscow(2017) (In Russian).
30. Belonogov G.G. Teoreticheskiye problemy informatiki, tom 2, Semanticheskiye problemy informatiki [Theoretical problems of computer science, Vol. 2, Semantic problems of computer science], Plekhanov State University of Economics, Moscow (2008) (In Russian).
31. Web-portal "Automated Text Analysis". http://textanalysis.ru/ Last accessed 15 Apr 2018
32. Kuzmina A.I.: Instrument vydeleniya imenovannyh sushchnostej iz tekstov na russkom yazyke [The tool for extracting named entities from texts in the Russian language] In: Proceedings of the XVIII International Scientific Conference "Computer science: problems, methodology, technology"]. VSU, Voronezh (2018), pp. 33–36. (In Russian).
33. Sivovolova M.A.:Programma ustraneniya sokrashchenij v slovarnyh stat'yah [The application for resolving the abbreviations in the dictionary entries] In: Proceedings of the XVIII International Scientific Conference "Computer science: problems, methodology, technology", VSU, Voronezh(2018), pp. 61 – 65. (In Russian)
34. Kostichev D.A., Politsyna E.V.: Razrabotka algoritma avtomatizirovannogo postroeniya semanticheskoj seti na osnove tolkovyh slovarej [Development of the algorithm for the automated creation of the semantic network based on dictionaries]. In: Proceedings of the XVIII International Scientific Conference "Computer science: problems, methodology, technology", VSU, Voronezh (2018), pp. 245–249. (In Russian)
35. Ivashchenko M.V.: Analiz metodov avtomatizirovannogo vydeleniya klyuchevyh slov iz tekstov na estestvennom yazyke [Analysis of methods for automated extraction of keywords from texts in the natural language] In: Proceedings of the XVIII International Scientific Conference "Computer science: problems, methodology, technology" VSU, Voronezh (2018) pp. 19–24. (In Russian)
36. Pryazhentseva A.A.: Issledovanie algoritmov vydeleniya klyuchevyh slovosochetanij iz tekstov na russkom yazyke [Investigation of algorithms for highlighting keyword combinations from texts in the Russian language] In: Proceedings of the XVIII International Scientific Conference "Computer science: problems, methodology, technology"] VSU, Voronezh(2018), pp. 56–60. (In Russian)
37. Porechny A.S.:Realizatsiya poiska ponyatiy s pomoshch'yu vydeleniya slovasochetaniy iz teksta [Implementation of the search for notions by selecting word combinations from a text] In: Proceedings of conference "Problems of computer linguistics and typology" VSU, Voronezh(2017), pp. 108–118. (In Russian)