UDK: 659.123:519.682:Python-nlpheart=861
COSATI: 12-05; 12-09
doi: 10.5937/str2003041O

A Python Package for Text Processing for Serbian – *nlpheart*

Stevan Ostrogonac [1)]
Borko Rastović [1)]
Elizaveta Liliom [1)]

Within the past two decades, text processing became an important part of most state-of-the-art advanced automation systems. However, for many under-resourced languages it is still challenging to perform textual data preparation, due to the lack of adequate tools. In this work, a python package for text processing for Serbian called *nlpheart* is presented. This package has been developed in industry and it is planned to be released as an open-source text processing tool for Serbian for academic purposes as well.

Key words: **— NLP, machine learning, text processing, Serbian.**

Introduction

RECENT years have brought rapid expansion of data science (DS) due to the need for data utilization for business purposes. Today, many large companies, as well as medium and even small businesses have reached a stage of data maturity that implies a systematic approach to data acquisition, preprocessing, structuring, storing, analyzing and decision making [1]. Furthermore, structured data is often utilized within machine learning (ML) models in order to automatize parts of business processes. Different types of data can be important for a particular business – numeric, text, audio or image (video) etc. Many algorithms have been developed for data processing and machine learning and they have been implemented and are open for use for both academia and industry. Currently, the most popular language for machine learning and data science is python. Logically akin algorithms or tools are usually joined within python packages that are easy to install and use. They are usually used as a black box, even though source code is available and can be modified.

Textual data represents an unstructured source of information that can be converted to structured data by applying algorithms for text preprocessing and later adequate tools for interpreting the text. There are many available python libraries for natural language processing (NLP). Some of them are general purpose (in the context of NLP), such as NLTK [2], TextBlob [3] or CoreNLP [4], which include text preprocessing techniques and many algorithms for different text analysis and modeling procedures (named entity recognition – NER, topic modeling, sentiment analysis, POS tagging, noun phrase extraction etc.). Other libraries are specialized for a particular task, such as Gensim [5], which provides tools for determining similarity between documents. Most of these libraries were not developed for production environment, although there are some that have been developed recently in order to address the speed issue, for example spaCy [6]. Another popular library for machine learning that offers many text processing tools is scikit-learn [7].

While machine learning is mostly a language-independent field of research, natural language processing requires a certain degree of adaptation to a particular language. Some of the mentioned libraries support many languages. However, there are still under-resourced languages, for which adequate NLP tools are yet to be implemented. For the Serbian language, efforts have existed in the past within different academic institutions to develop NLP tools and high-quality language resources. An accentuation-morphological dictionary was created at the Faculty of Technical Sciences in Novi Sad [8] and it currently contains over 4 million entries. It is currently being used for context analysis within text-to-speech and automatic speech recognition systems for Serbian [9], along with other language resources and text processing tools [10]. Furthermore, textual corpora with over 20 million tokens have been collected and processed in order to train language models that can be used as a basis for grammatical and semantic error detection and correction in text in Serbian [11]. Significant work has been conducted in the field of NLP on the Faculty of Philology in Belgrade, among which the most recent research was related to NER [12] and diacritization of text in Serbian [13]. However, the tools and language resources have not been open for research or application in industry. Regional Linguistic Data Initiative (ReLDI) has conducted the most recent and comprehensive development of NLP tools and language resources and has opened most of them for research [14-16]. Near the end of 2020, an extended version of the Stanza Python library under the name CLASSLA was released [17]. This library offers text processing pipelines for Serbian, Croatian, Slovenian, Macedonian and Bulgarian. The functionalities of this library are similar and have been developed in parallel to those offered by the library that will be described in this paper. However, the language resources and the NLP tools within CLASSLA are suitable mostly for general-purpose tasks. Furthermore, text processing output of CLASSLA is in the form of a list of annotated entities, while our tools provide processed text in the form of a sequence of words, which can be more suitable for practical purposes.

[1)] Infostud 3 Ltd, Vladimira Nazora 7, 24000, Subotica, SERBIA
Correspondence to: Stevan Ostrogonac, e-mail: stevan.ostrogonac@infostud.com

While most NLP problems, such as sentence boundary disambiguation, lemmatization and stemming, morphological clustering, NER etc. have been tackled by different research groups in Serbia, there have been no significant efforts in industry to develop NLP tools or a collection of language resources and apply them to businesses. The need for such resources has become apparent with the acceleration of digitalization. Within Infostud Group, which is a collection of classifieds websites and e-commerce businesses, the utilization of textual data (for both Serbian and English) has been recognized as a high priority task. The aim of this research was to develop an NLP python package for internal use, but it is planned to be released for public use and contribution under the name *nlpheart*. In time, this package will be updated with different domain language resources and adequate processing options.

In the following section of this paper, text processing functionalities of *nlpheart* are presented and described in detail. The section after that provides information on language resources that are an integral part of *nlpheart*. This is followed by a brief introduction to practical issues related to this tool, and in the final section, future research and improvements to the python library are discussed.

Text Processing With *nlpheart*

The package *nlpheart* was primarily based on the needs of the Infostud job board Poslovi (poslovi.infostud.com). However, it has been constructed in a manner that allows great flexibility. This was accomplished by relocation of many parameters from code to a configuration file, which gives a user the control over functionalities, eliminating the need for introducing multiple versions of each functionality. Among these parameters are mostly lists of symbols, abbreviations or words that are related to some rules of text processing. For example, there is a need to keep a filename with an extension as an integral chunk after preprocessing, therefore in the configuration file there is a list of all common file type extensions. Some parameters that are more likely to change for each particular use-case are left as method arguments, e.g. "*to_lower*", which indicates if preprocessing should include converting all letters to lowercase. Text processing with *nlpheart* can be performed in an easy manner through interface methods. The main methods (functionalities) and a few of many auxiliary ones are described in more detail in the rest of this section.

Language detection and translation

Texts in Poslovi are mostly written in Serbian or English, with less than 1% of texts (job ads) being written in German, Hungarian or Russian. For these five languages, a method *detectLanguage()* exists for identifying the language of the input text, and *translate()* is used for translating between any two of the mentioned five languages. Language detection is based on dictionaries that contain between 70 and 100 thousand words for each of the languages. Translation is based on manually constructed parallel domain dictionaries that contain 40 thousand of the most frequent words and phrases that appeared in approximately 300 thousand job ads. Both of these methods have an argument that can be used to switch to Google Translate service if there is a need to detect a language outside of the five that were mentioned, or to translate text outside of job ads domain or with the accuracy that is provided by context analysis.

Conversion between Cyrillic and Latin alphabet

Serbian alphabet has Cyrillic and Latin varieties, both of which are used in everyday as well as in formal communication. For machine learning, however, the data needs to be normalized in this context. While converting from Cyrillic to Latin is trivial, vice versa requires context analysis due to the existence of digraphs. The methods for these conversions are *cyrillicToLatin()* and *latinToCyrillic()*. There are also auxiliary methods for determining the type of alphabet (*isLatin()*, *isCyrillic()*).

Normalization

The method normalizeText() performs conversion of different symbols for a same character into a unique one. For example, different Unicode symbols may be used as an apostrophe, comma or some other character. The same is true for letters. This normalization represents a first step in text preparation for machine learning, which is why within this step some symbols are simply removed from text, for example: "¬", "¡", "❴", "ⓘ" etc.

Splitting sentences

The method *splitSentences()* performs a rule-based context analysis and splits text into sentences. It can also join text from two adjacent rows if it forms a single sentence. The current estimated accuracy is 99.8%, obtained on a test set of 2000 sentences taken from job ads and inspected manually. However, the rules are based on several lists of words that can easily be updated (if necessary) for a special domain of implementation.

Automatic preparation of text for machine learning

There are three methods that perform automatic text processing that results in a sequence of words that is ready for training ML models – *prepareTextForML()*, *translateAndPrepareTextForML()* and *prepareTextForMLSBD()*. These are basically the same method and the differences are related to the inclusion of additional processing steps, which could not be elegantly included within one method for certain technical reasons. The second function, therefore, performs language detection and translation to Serbian (if necessary) before proceeding with text processing. The third one separated output sequence of words to sentences. The basic method, *prepareTextForML()*, consist of several steps of text processing:

- Normalization. This process was described earlier in this section.
- Separation of text from punctuation. This is performed through context analysis in order to keep punctuation marks within text chunks where necessary – e.g. within names of files with extensions, e-mail addresses, web addresses etc.
- Making the text compliant with GDPR. The process of anonymizing personal or other sensitive data is very complex and it should be often revised and updated. Currently, *nlpheart* replaces with adequate tags all e-mail and web addresses, and phone numbers. Further development of this process will lean on rules or lists of words or phrases for specific domains. For job boards, that could include pay ranges, names of companies, perhaps locations, depending on a particular use etc., some of which are already in testing phases.
- Removing punctuation and other non-word chunks. This may optionally include removing isolated numerics.

- Diacritization. This feature has not been fully implemented yet, which is why it does not exist as a separate method at this time. However, a special case of the letter "*đ*", which is commonly written incorrectly as a digraph "*dj*" in text in Serbian, is treated by a rule-based subsystem which also depends on lists of exceptions that are provided within the configuration file. This subsystem determines where the construction "*dj*" should be converted to "*đ*" – e.g. in the word "*urađen*" (done), and when it simply represents two adjacent letters – "*d*" and "*j*", e.g. in the word "*odjednom*" (suddenly).
- Optional conversion to lowercase.

Lemmatization and inflection

The methods *getLemma()* and *getInflectedForms()* provide lemmatization and inflected forms of single words. There are also auxiliary methods such as *lemmatizeText()* that provide additional convenience for a user. These methods rely on lookup dictionaries that contain around 1 million entries. At this time, simple lookup is performed, without context analysis for semantic disambiguation.

Similarity calculation

Levenshtein distance calculation is wrapped within the method *minimumEditDistance()*. It provides a basis for fuzzy matching, which is important for some of the previously mentioned processing techniques.

Language Resources For Serbian

Text processing represents an initial step in exploiting valuable information through machine learning or expert systems. The next step is converting text to numeric representation. In order to perform these steps, language resources are needed. Some of them have been mentioned in section 2. Here, they will be described systematically and in more detail, since they are a part of the *nlpheart* library.

Dictionary for lemmatization and inflection

Lemmatization and inflection are very important for addressing the problems that are related to the very complex morphology of the Serbian language. Reducing the number of dimensions for numeric representation of text is of great importance in the context of the accuracy of ML models. The lemmatization and inflection features of *nlpheart* rely on word-lemma and lemma-inflections maps that contain around 1 million entries. These maps were constructed form the language resources of Infostud as well as from various chunks of data that were available online. However, currently both features are based on simple look-up, which introduces a certain percentage of errors. Plans for future research include the development of an expert system for context analysis.

The mentioned maps practically contain a general-purpose Serbian vocabulary as well. It can be used for many applications together with domain-specific vocabularies, e.g. for diacritization, spell-checking etc.

Domain vocabulary of Poslovi

Domain vocabulary of Poslovi was derived from the texts of job ads and it originally contained just over 170 thousand words. A significant percentage of those were erroneously written words or rare words that are not typical for the domain of interest, and they were removed initially by applying a threshold for a minimal frequency, which was 90. Later, stop-words were removed, and in the final stage of the preparation manual inspection was performed in order to remove non-word constructions or remaining typing mistakes that somehow occurred more than 90 times in the corpus. Two final versions of the domain vocabulary have been produced, which contained 40 and 80 thousand words. Both of them are included in *nlpheart*. Furthermore, a domain vocabulary of 30 thousand lemmas has been added to the library as well.

Language detection

For language detection, general-purpose vocabularies for Serbian, English, Russian, Hungarian and German are used. These vocabularies were obtained from open GitHub repositories, e.g. [18]. Each of the vocabularies contains between 80 and 100 thousand words. Words from different languages that have the same transcription were removed, since they do not contribute to the language detection task. Detection of the five languages in which the job ads on poslovi.infostud.com are written was tested with these vocabularies and it was 100% after the ads that contained paragraphs in multiple languages were removed from the test set. Naturally, the accuracy may not be the same within other domains of use.

Language translation

Even though Google Translate is a good service for text translation, our idea was to create a very simple system that would not rely on the availability of an external service. Therefore, the domain vocabulary of Poslovi, which was described in the *B* subsection of this section, was translated with the Google service to the four remaining languages that the *nlpheart* currently features. The same was done with the significantly smaller domain vocabularies for German, Russian and Hungarian, and for the English Domain Dictionary, which is a bit closer to the size of the vocabulary in Serbian. The results were collected and, after duplicates were removed, the size of the vocabularies was just over 45 thousand words. Later, some bad translations were manually removed and the final count of entries in the vocabularies was similar to the number of entries of the original domain vocabulary for Serbian.

Translation that leans on these vocabularies requires a logic that allows one word to become a sequence of words after translation. The quality of this rudimentary translation is not sufficient for anything but for text classification tasks, which was confirmed in some experiments that were conducted in another internal study within Infostud. However, this system can be further improved in the future in order to become useful for more sophisticated tasks.

Textual corpus of Poslovi job ads

After the full processing that was described within this paper, the collection of around 300 thousand job ads texts was added to *nlpheart* as a domain textual corpus that can be of use for researchers. Currently, this corpus contains 1,885,625 sentences, 23,164,550 tokens, and the full vocabulary contains just over 172,444 words.

Practical Use

Using *nlpheart* is fairly simple. It requires only three lines of code to enable text processing.

```
import nlpheart.src.NLPHeart as nlpheart
nlp = nlpheart.NLPHeart()
nlp.loadResources()
```

However, it takes around 5s to load all the language resources. In case there is no need for loading some of them, this can be modified by changing some of the flags at instantiation of the NLPHeart class.

```
nlp = nlpheart.NLPHeart(required_resources = {"lemma_inflection": True, "lt_vocabs": True, "txt_corpora": False, "vocab": False})
```

For example, if there is no need for lemmatization or inflection, the flag "*lemma_inflection*" can be set to "*False*". That would reduce loading time to several hundred milliseconds. As more resources will be added to this library in the future, this configuration will most likely need to be moved to the configuration file as well.

After initialization, any of the methods can be called on a text as in the following example:

```
output_text = nlp.prepareTextForML(input_text)
```

Text processing has been fairly optimized. Currently, 300 thousand ads can be processed for machine learning (*prepareTextForML()*) within around 10 minutes on a standard laptop.

For practical purposes, *nlpheart* can be deployed as a service in order to enable efficient online text processing [19]. In Infostud, it has been open for all the websites from the Group as a service within a Docker container [20].

Future Work

The hope of the authors is that *nlpheart* will gain contributors from other research teams that are willing to work towards developing open datasets and open tools for more efficient research and development and for bringing natural language processing for Serbian to the level at which this field is for some other languages [21].

Our own contributions to this library that are currently being discussed or implemented are:

- Introducing semantic classes by exploiting word vectors that have been opened to public [22],
- Improving translation vocabularies and logic,
- Adding domain corpora and vocabularies form other domains within Infostud Group,
- Developing sophisticated context analysis for word-sense disambiguation, either as an expert system, or as an artificial intelligence solution.

References

[1] Y. Zhu and Y. Xiong: "*Towards Data Science*", Data Science Journal, Vol. 14, No. 8, pp. 1-7, 2015, DOI: http://dx.doi.org/10.5334/dsj-2015-008

[2] E. Loper, S. Bird: "*NLTK: the Natural Language Toolkit*", In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, pp. 62-69, Somerset, NJ: Association for Computational Linguistics, 2002, DOI: 10.3115/1118108.1118117

[3] TextBlob: Simplified Text Processing [Online]. Website, Accessed on September 28th, 2020, https://textblob.readthedocs.io/en/dev.

[4] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky: "*The Stanford CoreNLP Natural Language Processing Toolkit*", In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60, 2014.

[5] R. Rehurek, P. Sojka: "*Software framework for topic modeling with large corpora*", In: Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks. ELRA, pp. 45-50, Valletta, Malta, May 22nd, 2010.

[6] M. Honnibal and I. Montani: "*spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*", to appear, 2020.

[7] F. Pedregosa et al: "*Scikit-learn: Machine Learning in Python"*, Journal of Machine Learning Research Vol. 12, No. 85, pp. 2825-2830, 2011.

[8] M. Sečujski: "*Accentuation Dictionary for Serbian Intended for Text-to-Speech Technology*", Proceedings of DOGS, pp. 17-20, Novi Sad, Serbia, 2002.

[9] S. Ostrogonac, E. Pakoci, M. Sečujski, and D. Mišković: "*Morphology Based vs Unsupervised Word Clustering for Training Language Models for Serbian*", Acta Polytechnica Hungarica, Journal of Applied Sciences, Joint Special Issue on TP Model Transformation and Cognitive Infocommunications, Vol. 16, No. 2, pp. 183-197, 2018, ISSN: 1785-8860

[10] S. Suzić, S. Ostrogonac, E. Pakoci, M. Bojanić: "*Building a Speech Repository for a Serbian LVCSR System*", Telfor Journal, Vol. 6, No. 2, pp. 109-114, 2014.

[11] S. Ostrogonac: "*Automatic Detection and Correction of Semantic Errors in Texts in Serbian*", Primenjena lingvistika (Applied Linguistics), Vol. 17: 265-278, 2016, ISSN 1451-7124, UDK: 81'33.

[12] B. Šandrih, C. Krstev, R. Stanković: "*Development and Evaluation of Three Named Entity Recognition Systems for Serbian - the Case of Personal Names*", In Proceedings of the International Conference Recent Advances in Natural Language Processing - RANLP 2019, eds. G. Angelova et als., pp. 1061-1068, Varna, Bulgaria, 2019. DOI: 10.26615/978-954-452-056-4_122

[13] C. Krstev, R. Stanković, D. Vitas, "*Knowledge and Rule-Based Diacritic Restoration in Serbian*", In Proceedings of the Third International Conference Computational Linguistics in Bulgaria (CLIB), The Institute for Bulgarian Language Prof. Lyubomir Andreychin, Bulgarian Academy of Sciences, pp. 41-51, Sofia, Bulgaria, May 27-29, 2018, ISSN 2367-5675 (on-line)

[14] V. Batanović, N. Ljubešić, and T. Samardžić: "*SETimes.SR – A Reference Training Corpus of Serbian*", In Proceedings of the Conference on Language Technologies & Digital Humanities (JT-DH), Ljubljana, Slovenia, pp. 11-17, 2018.

[15] M. Miličević, and N. Ljubešić: "*Tviterasi, tviteraši or twitteraši? Producing and analysing a normalised dataset of Croatian and Serbian tweets*", Slovenščina 2.0 4(2), pp. 156-188, Slovenia, 2016.

[16] V. Batanović, B. Furlan, B. Nikolić: "*A software system for determining the semantic similarity of short texts in Serbian*", In Proceedings of the 19th Telecommunications Forum (TELFOR), pp. 1249-1252, Belgrade, Serbia, 2011.

[17] V. Batanović, N. Ljubešić, T. Samardžić, M. Miličević Petrović: "Otvoreni resursi i tehnologije za obradu srpskog jezika", Primena slobodnog softvera i otvorenog hardvera 2020 (PSSOH 2020), Belgrade, Serbia, October 2020, DOI: 10.5281/zenodo.4113229

[18] https://github.com/hingston/russian, accessed on September 30th, 2020.

[19] F. Armash Aslam, H. N. Mohammed, and P. S. Lokhande: "*Efficient Way of Web Development Using Python and Flask*", International Journal of Advanced Research in Computer Science, Vol. 6, No. 2, 2015, DOI: 10.26483/ijarcs.v6i2.2434.

[20] B. B. Rad, H. J. Bhatti, and M. Ahmadi: "*An Introduction to Docker and Analysis of its Performance*", JCSNS International Journal of Computer Science and Network Security, Vol. 17, No. 3, 2017.

[21] D. Sarkar: "*Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data, 2nd edition*", 2019, ISBN: 9781484243534

[22] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov: "Learning Word Vectors for 157 Languages", Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May, 2018.

Received: 17.02.20201.
Accepted: 24.03.20201.

Biblioteka za obradu teksta na srpskom jeziku pisana u programskom jeziku *python – nlpheart*

U toku protekle dve decenije, obrada teksta je postala značajna komponenta većine savremenih sistema za naprednu automatizaciju. Ipak, za mnoge jezike ne postoje kvalitetni jezički resursi i alati koji bi omogućili efikasnu pripremu tekstualnih podataka. U ovom radu je opisana biblioteka *nlpheart* za obradu teksta na srpskom jeziku, koja je pisana u programskom jeziku *python*. Ova biblioteka realizovana je za potrebe industrije, ali je u planu njeno objavljivanje u vidu alata za obradu teksta na srpskom jeziku koji bi bio dostupan za nadogradnju i za istraživanja.

Ključne reči: **obrada prirodnog jezika, mašinsko učenje, obrada teksta, srpski jezik.**