NATURAL LANGUAGES AND PROGRAMMING LANGUAGES: A CASE STUDY

Boško Nikolić [1], Dražen Drašković [1], Vuk Batanović[2]

[1] School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11000 Belgrade, Serbia
e-mail: nbosko@etf.bg.ac.rs , drazen.draskovic@etf.bg.ac.rs
[2] Innovation Center of the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11000 Belgrade, Serbia
e-mail: vuk.batanovic@ic.etf.bg.ac.rs

Abstract:

For understanding software, comments and functional specifications are the second most-used documentary artifact, after the code itself. Our aim is to examine the relationship between such descriptions of code, databases, and models, written in a natural language, and the objects they describe. We explored the extent to which the semantics of a textual description reflect the semantic and structural characteristics of the described object, and work on bridging the gap between natural languages and the programming languages in which the objects are defined. Our AI solutions utilize cross-level semantic similarity and software comment categorization as components of an intelligent system which leverage the correlation between object similarity and comment similarity for the tasks of object clone detection and semantic code search. The necessary datasets for machine learning models (the first models and datasets for both cross-level semantic similarity and semantic code search in Serbian) were be produced via linguistic annotation and analysis. Innovations in the realm of natural language processing technologies for the Serbian language are especially important in the local context.

Keywords: natural language processing, cross-level semantic similarity, semantic code search

1. Introduction

Natural Language Processing (NLP) is a growing area of artificial intelligence, and due to its widely used applications in day-to-day life, its significance is also getting recognized by the society. NLP allows computers to resolve ambiguity in natural language and adds useful numeric structure to the language data [1-4]. The software industries, particularly software development and use, are becoming an increasingly important part of the global, European and Serbian industry, society, and economy. With this growth comes an increase in the number of employees in the software industry, the amount of code and the size of software. As a negative consequence, there is a lot of re-writing of already implemented pieces of software, and it is often necessary to use someone else's code, databases and models, making software more difficult to understand, and more complicated to maintain and test. Another critical issue in the software industry is detecting copyright infringement, i.e. determining if two pieces of software are the same. Currently, there is no effective way for a company to verify whether someone is using parts of its software [5], [18].

In light of these facts, the aim of our project AVANTES (Advancing Novel Textual Similarity-based Solutions in Software Development - Program for Development of Project in the field of Artificial Intelligence, funded by Science Fund Republic of Serbia) is to utilize AI and NLP to help developers in identifying similar parts of code, models and databases in

order to reduce the maintenance effort, reuse test cases for them or fix similar bugs. The broader, overarching goal of the AVANTES project is to determine the extent to which the semantics of a textual description reflect the semantic and structural characteristics of software objects (code, databases, and models), and to bridge the gap between natural languages and programming languages, in which the objects are defined. The project team is multidisciplinary and consists of researchers from the School of Electrical Engineering, University of Belgrade, the Faculty of Philology, University of Belgrade and the Innovation Center of the School of Electrical Engineering.

The project targets the above-mentioned research challenges through three main objectives: 1) a new software similarity tool that relies both on code similarity and comment similarity and is applicable to multiple natural languages and programming language; 2) a new semantic search algorithm for exploring and analyzing general and database-related programming code using natural language input; 3) new natural language processing datasets and models for the Serbian language, including the first cross-level semantic similarity and semantic code search systems for Serbian.

2. Concept and methodology

One of the objectives of our AVANTES project is to define new software similarity tool that rely both on code similarity and comment similarity and is applicable to multiple natural languages and programming languages. For achieving this objective, multiple steps can be performed. The first step involved creating a dataset of programs suitable to be analyzed. At the beginning, a set of publicly available programs, libraries, and data structures with their course codes were collected and analyzed. The second step was focused on using existing and creating new methods for source code analysis using existing machine learning (ML) and AI techniques. This could help in solving the first problem of finding software clones type one to three. Detecting semantic software clones, the second problem, is a much harder problem than detecting the software clones [15]. After this, existing solutions are evaluated.

As the next step, determining the level of similarity between blocks of code in the absence of their source codes was performed [12]. The programs from the dataset were compiled using selected compilers and compiling options. Information related to source code were preserved for evaluation purposes. These compiled programs (e.g. Java byte code, machine codes, CIL) were evaluated [21]. The evaluation could include, but was not limited to, static code analysis, dynamic code analysis, execution trace analysis, memory access analysis. The analysis found that machine learning techniques could complement existing methods of code comparison. The results from this step on code comparison at this intermediate level were compared with the results obtained at the programming language level.

As the final step, the developed code similarity and comment similarity algorithms were integrated. This step aims to discover the way in which the developed algorithms and methods for determining the level of similarity between two segments of programming code could be integrated with the developed algorithms and methods for determining semantic similarity between comments associated with the code segments.

The objective is to determine: Would it be adequate to use comment similarity as an additional element in the comparison vector when comparing the code [13]? Would it be adequate to use code similarity as an additional element in the comparison vector while comparing comments? What elements of the code similarity vector and the comment similarity vector are important in comparing commented code?

Software comments have long been identified as important sources of information regarding a software product, second only to the code base [6]. Several categorization systems and taxonomies have been presented over the years [16], [20], but they were all general. In this project, our goal is to design a comment taxonomy that is particularly suited to the downstream tasks of semantic similarity and semantic code search. In solving the categorization problem, we plan to rely on supervised machine learning algorithms, which have proven to be the optimal approach to text classification problems, and to create

appropriate annotated datasets according to the new comment taxonomy which we will design.

Cross-Level Semantic Similarity (CLSS) is the task of comparing the meaning of two texts of different lengths, e.g. a paragraph and a sentence, or a sentence and a phrase. This task is related to the problems of short-text semantic similarity and text summarization but is more difficult than measuring semantic similarity between texts of equal or similar lengths. The predominant methods of tackling the CLSS task, as is the case in the majority of NLP problems, rely on supervised machine learning models, necessitating the creation and annotation of datasets appropriate for the task. Due to this, previous work on CLSS has been rather limited and focused solely on English [10-11]. For the same reason, very little is known about the linguistic properties associated with different similarity scores in CLSS. Even though multiple linguistic factors have already been shown to affect short text similarity, some of them cannot be directly applied to comparisons of different sized textual units, which makes an interdisciplinary perspective reliant on a detailed linguistic analysis necessary for selecting linguistic features relevant for CLSS machine learning models.

The trans-disciplinary aspect of this project involves the analysis of the linguistic factors that create cross-level semantic similarity in natural language [7]. The phenomenon of semantic similarity is measured between items of different types: paragraphs, sentences, and phrases. We aim at establishing the extent to which the meaning of the larger item is captured in the smaller type. Special attention is given to the comparison between Serbian and English, as well as to the comparison between the language of code comments and the more general domain of newspaper texts [14].

Semantic Code Search (SCS) is the task of searching a code repository on the basis of a natural language query and retrieving blocks of code (e.g functions, classes, etc.) which are relevant to the given query. It is similar to the Information Retrieval task, but the results in SCS are code blocks, rather than documents in Information Retrieval. SCS is a recently conceptualized task, and research on it has so far focused only on queries in English ([8],[9]). The main idea of SCS models is to embed both natural language queries and blocks of code into a common space and to then perform natural language search using text embeddings and nearest neighbor search algorithms.

3. Data usage

The AVANTES project is strongly data-driven and includes numerous activities on compiling new datasets and using them to construct AI models, as well as to perform engineering and linguistic analyses.

In terms of data collection, we collect the following types of data:

- Software comments, written by software developers in a natural language - we consider two languages, English and Serbian. These comments were collected from various sources, including student projects, coursework, and final thesis at the School of Electrical Engineering, software projects developed at the Computing Center of the School of Electrical Engineering, and other interested industry partners, as well as public repositories such as GitHub and previously developed and publicly available comment-related datasets. These data sources were evaluated and filtered for comment quality. The collected comments were analyzed in order to create a comment type taxonomy.
- Newswire texts of different lengths, written in Serbian, and gathered from Serbian online news sources for the task of CLSS. We plan to consider only the Serbian language with regard to this part of data collection, since adequate CLSS newswire datasets already exist in English.
- A set of (software comment, code block) pairs, with comments written in Serbian, which will be used in the creation of a Semantic Code Search mechanism for Serbian. In the construction of this set, we considered the following programming languages: Java, JavaScript, PHP, C/C++/C#, SQL. Data collection sources for this set are the same as the ones listed for software comment collection. In the creation and

evaluation of an improved SCS model for English, we rely on the existing dataset of this kind in English.

The datasets which we created during the course of this project are as follows:

- Two datasets of software comments, one containing comments written in English and the other containing comments written in Serbian. Comments in both datasets were manually labeled according to their category using a comment type taxonomy which were developed within the project. These datasets contain at a minimum of several thousand items. The primary use of this data is to train and evaluate machine learning classifiers on the task of categorizing software comments in English and Serbian.
- Three datasets of textual pairs of different lengths, manually annotated with semantic similarity scores for each pair. These datasets differ amongst themselves with regard to the language and the domain of the collected textual pairs, and will include the following:
  - A dataset of newswire text pairs in Serbian
  - A dataset of software comment pairs in Serbian
  - A dataset of software comment pairs in English

  The length of the texts included in all three datasets ranges from a phrase and sentence, to a paragraph. Texts of different lengths are paired in (phrase, sentence) and (sentence, paragraphs) groupings and manually annotated with fine-grained semantic similarity scores by five annotators. Each language-domain-pair length combination is represented by around a thousand examples, for a total of around two thousand items for each of the three datasets. The primary use of these datasets will be to train and evaluate machine learning models on the task of Cross-Level Semantic Similarity and to assess these models' behavior on different languages and domains.
- One dataset consisted of publicly available programs to be used as a benchmark for software similarity purposes. Based on this initial version of the dataset an updated version of dataset created by solving some set of problems multiple times by multiple persons. This were done by analyzing student projects, coursework, and theses at the School of Electrical Engineering.
- A dataset of (software comment, code block) pairs, with the comments written in Serbian. Its exact properties and size are difficult to estimate at this point, because it largely depends on the related dataset of software comments in the Serbian language and the set of data of publicly available programs.
- A collection of up to 80 natural language queries in Serbian is designed, and a small subset of the (comment, code) dataset is marked with relevance scores with regard to said queries, to enable subsequent evaluation of semantic code search models.

3. Conclusions

Our exploration of the CLSS task in Serbian, a resource-limited yet morphologically rich language, promote this task in the wider NLP community as well as push forward language technologies for Serbian. Furthermore, we aim to consider a completely new and highly specialized domain for CLSS in both languages - software comments - and to compare and contrast it to more general domains, such as newswire texts. We expect this line of research to lead to more robust CLSS models, and to tackle the issue of domain adaptation in the realm of CLSS. To further promote research in all these novel language/domain combinations, we will create and make publicly available new annotated CLSS datasets. Moreover, we will strengthen the linguistic component of the task by conducting an in-depth analysis of several levels of linguistic structure (morphosyntax, semantics, discourse), some of which are not typically taken into account in short text similarity tasks, but have important potential for CLSS. By doing so we will simultaneously work on the broader goal of bringing NLP and linguistics closer.

The aim of described AVANTES project is to develop SCS models for Serbian, tackling the resource limitations of this and many other minor languages on the SCS task. We will also endeavor to improve both Serbian and English SCS models using the developed code

similarity methods, for both general programming code and SQL database code. In particular, we will explore the extension of search results with additional code blocks found by using code similarity methods on the initial SCS results. We will also consider using code similarity to improve search result ranking. Finally, we plan to explore the multilinguality often present in software comments and its effects on the functioning of SCS systems, which is a hitherto unaddressed issue.

Innovations in the realm of Natural Language Processing technologies for the Serbian language are especially important in the local context, as this project will lead to the creation of the first models and datasets in Serbian for both Cross-Level Semantic Similarity and Semantic Code Search. This will provide a huge impetus for future work on these problems in Serbian and, due to the novelty and resource scarcity related to both of the aforementioned tasks, it will also have the potential to generate research interest in the global NLP community. Moreover, the main contribution of AVANTES at the international level is the synergistic effect that is expected from combining semantic analysis of both, comments and code.

Acknowledgment: This work was supported by the Science Fund of the Republic of Serbia, grant no. 6526093, AI-AVANT

References

[1] Batanović, V., and Bojić, D. (2015). Using Part-of-Speech Tags as Deep-Syntax Indicators in Determining Short-Text Semantic Similarity. Computer Science and Information Systems, 12(1), pp. 1-31.

[2] Batanović, V., and Nikolić, B. (2017). Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization and Word Embeddings. Telfor Journal, 9(2), pp. 104-109.

[3] Batanović, V., and Nikolić, B. (2019). Using Language Technologies to Automate the UNDP Rapid Integrated Assessment Mechanism in Serbian. In Proceedings of the International Conference on Language Technologies for All (LT4All), Paris, France.

[4] Batanović, V., Cvetanović, M., and Nikolić, B. (2018). Fine-grained Semantic Textual Similarity for Serbian. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, European Language Resources Association (ELRA), pp. 1370-1378.

[5] Cvetanović, M., Radivojević, Z., and Milutinović, V. (2017, June). Restart optimization for transactional memory with lazy conflict detection. International Journal of Parallel Programming, Vol. 45, No. 3, pp. 482-507.

[6] de Souza, S. C. B., Anquetil, N., and de Oliveira, K. M. (2005). A Study of the Documentation Essential to Software Maintenance. Proceedings of the 23rd ACM Annual International Conference on Documentation (SIGDOC 2005), New York, USA.

[7] Furlan, B., Batanović, V., and Nikolić, B. (2013). Semantic similarity of short texts in languages with a deficient natural language processing support. Decision Support Systems, 55(3), pp. 710-719.

[8] Husain, H. (2018, September). Towards natural language semantic code search. URL https://githubengineering. com.

[9] Husain, H., Wu, H. H., Gazit, T., Allamanis, M., and Brockschmidt, M. (2019). CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. arXiv preprint arXiv:1909.09436.

[10] Jurgens, D., Pilehvar, M. T., and Navigli, R. (2014, August). Semeval-2014 task 3: Cross-level semantic similarity. In Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp. 17-26.

[11] Jurgens, D., Pilehvar, M. T., and Navigli, R. (2016). Cross level semantic similarity: an evaluation framework for universal measures of similarity. Language Resources and Evaluation, 50(1), pp. 5-33.

[12] Kamp, M., Kreutzer, P., and Philippsen, M. (2019, May). SeSaMe: a data set of semantically similar Java methods. In 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), pp. 529-533. IEEE.
[13] McBurney, P. W., and McMillan, C. (2016). An empirical study of the textual similarity between source code and source code summaries. Empirical Software Engineering, 21(1), pp. 17-42.
[14] Miličević, M., and Ljubešić, N. (2016). Tviterasi, tviteraši or twitteraši? Producing and analysing a normalised dataset of Croatian and Serbian tweets. Slovenščina 2.0: Empirical, Applied and Interdisciplinary Research, 4(2), pp. 156-188.
[15] Mostaeen, G., Svajlenko, J., Roy, B., Roy, C. K., and Schneider, K. A. (2018, September). On the Use of Machine Learning Techniques Towards the Design of Cloud Based Automatic Code Clone Validation Tools. In 2018 IEEE 18th International Working Conference on Source Code Analysis and Manipulation (SCAM), pp. 155-164. IEEE.
[16] Pascarella, L., Bruntink, M., and Bacchelli, A. (2019). Classifying code comments in Java software systems. Empirical Software Engineering, 24(3), pp. 1499-1537.
[17] Pejić, N., Cvetanović, M., and Radivojević, Z. (2019, November). Estimating similarity between differently compiled procedures using neural networks. Telfor XXVII, Belgrade.
[18] Radivojević, Z., Cvetanović, M., and Stojanović, S. (2016, January). Comparison of Binary Procedures: A Set of Techniques for Evading Compiler Transformations. Computer Journal, Vol. 59, No. 1, pp. 106-118.
[19] Ragkhitwetsagul, C., and Krinke, J. (2017, February). Using compilation/decompilation to enhance clone detection. In 2017 IEEE 11th International Workshop on Software Clones (IWSC), pp. 1-7. IEEE.
[20] Steidl, D., Hummel, B., and Juergens, E. (2013, May). Quality analysis of source code comments. In 2013 21st International Conference on Program Comprehension (ICPC), pp. 83-92. IEEE.
[21] Stojanović, S., Radivojević, Z., and Cvetanović, M. (2015, February). Approach for Estimating Similarity between Procedures in Differently Compiled Binaries. Information and Software Technology, vol. 58, pp. 259-271.