Using Deep Learning to Improve the Accuracy of Requirements to Code Traceability

Yu Zhao, Tarannum S. Zaman, Tingting Yu, Jane Huffman Hayes
Department of Computer Science
University of Kentucky
Lexington, Kentucky, 40506, USA
yzh355@g.uky.edu, tarannum.zaman@uky.edu, tyu@cs.uky.edu, hayes@cs.uky.edu

***Abstract*—Context and motivation. Information retrieval (IR) techniques have been used to recover traceability links between natural language requirements and source code. However, IR techniques are often lack of accuracy. To address this problem, research has shown that mining software repositories and using the mined results combined with the IR techniques can improve the accuracy [1], [4]. For example, Histrace [1] identifies traceability links between requirements and source code through CVS/SVN change logs using a Vector Space Model (VSM). The log messages are tied to changed entities and, thus, can be used to infer traceability links.**

Problem statement. While these approaches are promising, they rely on the assumption that different types of knowledge (e.g., commit messages, code comments) of the repositories exist. In many cases, however, such knowledge may not be available. For example, code commenting has been a standard practice in software development. Despite the need and importance of code comments, many code bases do not contain adequate comments [3]. Another type of knowledge involves commit messages, which have been used to document changes of software in version control systems. However, research [5] have shown that 14% of the commit messages are empty and 66% of the messages contain fewer words than a typical English sentence. To address the above problems on inadequate documentation, research on automated natural language text generation in software repositories have been proposed. For example, Wong et al. [7] generate comments automatically by mining Question and Answer (Q&A) for code-comment mappings. However, this approach has several drawbacks. First, it cannot handle cases in which the text descriptions do not exist in the mapping database. Second, there is not a notion of semantic similarity between words when generating the comments. Third, this approach is not scalable in the presence of large amount of data involving the code-comment mappings. Regarding the commit message generation, ChangeScribe [2] generates commit messages by taking into account the change types, such as file rename and deletion. However, this message generation approach is based on pre-defined templates and thus may not represent the real meanings of the changes.

Ideas and results. In this research, we propose an approach to automatically generate natural language texts that can build the bridge to recover traceability links between requirements and code. We focus on commit message and code comments generation. To address the aforementioned challenges imposed by existing techniques, we employ the deep neural network (also known as deep learning), featured by its ability of learning highly complicated features automatically [6]. We propose to leverage recurrent neural networks (RNNs), which are suitable for modeling texts (i.e., a sequence of characters) by its iterative nature. Natural language generation using RNNs differ from text mining and retrieval systems; the generated descriptions are different from any existing commit messages or comments, which are more flexible and may accurately reflect the semantic meanings. We will use Web Crawler to craw HTMLs in the Question and Answer (e.g., StackOverflow) and tutorial web sites (e.g., W3C). We can then utilize the natural language processing method to obtain the mapping between code and its corresponding descriptions. Next, we will train the RNNs by using these mappings. Specifically, The source code is the input to the RNNs and the text description (i.e., commit messages or comments) are the labels. Since today's software artifacts have become "big data", the training data is sufficient. As such, it is possible to train a generative text model based on the source code. Finally, given a code segment, the trained model can generate the corresponding text descriptions.

Contribution and future direction. In this research, we propose to train deep neural networks for generating text-based knowledge in software repositories to improve the accuracy of traceability links recovery. We will perform an empirical study to evaluate our proposed approach. We envision several scenarios where deep neural networks may address long-standing software engineering research challenges, including automated program generation from natural languages and test oracle generation.

REFERENCES

[1] N. Ali, Y. G. Guéhéneuc, and G. Antoniol. Trustrace: Mining software repositories to improve the accuracy of requirement traceability links. *IEEE Transactions on Software Engineering*, 39(5):725–741, 2013.

[2] L. F. Cortés-Coy, M. Linares-Vásquez, J. Aponte, and D. Poshyvanyk. On automatically generating commit messages via summarization of source code changes. In *Source Code Analysis and Manipulation (SCAM), 2014 IEEE 14th International Working Conference on*, pages 275–284, 2014.

[3] S. C. B. de Souza, N. Anquetil, and K. M. de Oliveira. A study of the documentation essential to software maintenance. In *Proceedings of the 23rd annual international conference on Design of communication: documenting & designing for pervasive information*, pages 68–75. ACM, 2005.

[4] B. Dit, A. Holtzhauer, D. Poshyvanyk, and H. Kagdi. A dataset from change history to support evaluation of software maintenance tasks. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 131–134, 2013.

[5] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 422–431. IEEE Press, 2013.

[6] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[7] E. Wong, J. Yang, and L. Tan. Autocomment: Mining question and answer sites for automatic comment generation. In *Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on*, pages 562–567, 2013.