Exploring the Naturalness of Buggy Code with Recurrent Neural Networks)

Jack Lanchantin JJL5SW@VIRGINIA.EDU
University of Virginia, Department of Computer Science

Ji Gao EMAIL@COAUTHORDOMAIN.EDU
University of Virginia, Department of Computer Science

Abstract

Statistical language models are powerful tools which have been used for many tasks within natural language processing. Recently, they have been used for other sequential data such as source code. (Ray et al., 2015)

1. Introduction

Natural language is inherently very well understood by humans. There are certain linguistics and structures associated with natural language which make it fluid and efficient. These repetitive and predictive properties of natural language make it easy to exploit via statistical language models. Although the actual semantics are very much different, source code is also repetitive and predictive. Some of this is constrained by what the compiler expects, and some of it is due to the way that humans construct the code. Regardless of why it is predictable, it has been shown that code is accommodating to the same kinds of language modeling as natural language (**?**).

Recently, (Ray et al., 2015) showed that it is possible to predict buggy lines of code based on the entropy of the line with respect to a code language model.

2. Related Work

2.1. Bug Dection

For software bug detection, there are two main areas of research: bug prediction, and bug localization.
Bug prediction, or statistical defect prediction, which is concerned with being able to predict whether or not there is a bug in a certain piece of code, has been widely studied in recent years (**?**). With the vast amount of archived repositories in websites such as github.com, there are many opportunities for finding bugs in code.
Static bug finders, automatically find where in code a bug is located. There has been a wide array of recent work in this area (**?**).

2.2. Natural Language Processing

3. Methods

3.1. Recurrent Neural Network Language Model

3.2. Experimental Setup

4. Results

5. Threats to Validity

6. Conclusion

References

Ray, Baishakhi, Hellendoorn, Vincent, Tu, Zhaopeng, Nguyen, Connie, Godhane, Saheel, Bacchelli, Alberto, and Devanbu, Premkumar. On the" naturalness" of buggy code. *arXiv preprint arXiv:1506.01159*, 2015.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.