Nicholas Garrett
garrnic3@isu.edu
Professor Zibran
CS 4416
10/17/2021

Reading/Critique: On The Naturalness Of Buggy Code

Paper:

On The "Naturalness" of Buggy Code

Authors: Baishakhi Ray, Zhaopeng Tu, Vincent Hellendoorn, Alberto Bacchelli, Saheel Godhane, and Premkumar Devanbu

Summary:

In standard programming samples, good programming practices occur more often then bad ones, and bad programming practices are often the ones that cread and lead to buggy code. This paper tries to use a system for identifying how the "naturalness" of a block of code, or how likely it is to be written; less natural code will be flagged for the programmer to look at more closely for bugs.

Strengths:

1. The process of how "good code" was identified was described in the paper. I liked this because it helped to describe their process in defining how their program operated.

2 The paper used a mathematical equation to model the bug entropy, which I liked as it gave more validity to the relationship.

3 I liked the use of visual representations for some of the things described in the paper. They helped to better describe, or rather show, the principles.

Weaknesses:

1. The bugfixes used to train the algorithm might have contained errors themselves, making the analysis inaccurate. A way to solve this problem would have been to go through each bugfix to be used and manually ensure it is bug-free.

2. The code repositories used to train the natural language engine could mostly contain bug fixes of similar things, so outside of a reasonably narrow range of situations the system may not have sufficient data to operate. Unfortunately, I do not know what they could have done to solve this problem.

3. The language model assigned a probability to each work, though it seems unreasonable to expect for a "word's" probability to be exact. Due to the multiplication operator, any inaccuracies expand into a larger and larger error. A way to solve this problem would be to use a more stable system for calculating the probability of a line.