Using Weka Machine Learning to Detect Syntax Errors in GitHub Commits

Hannah Stormer[1], Eisha Ahmed[1], Monica Bui[2], Joshua Campbell, Dr. Abram Hindle
Department of Computing Science University of Alberta [1]WISEST Student [2]HIP Student

Research Question

Can machine learning be an effective tool for finding syntax errors in Python code in commits on GitHub?

Introduction

- Syntax errors (fig. 1) are common, but it can be difficult to detect their location in a program.
- Data about what kinds of errors programmers make in Python can be found by looking through GitHub commits to see if syntax errors are present.
- This data will be used in the creation of a program that will improve detection of these errors.
- In order to gather a sufficient amount of data, it is necessary to automate the process of looking for errors.

```
modentries = []

for entry in entries:
        store = entry.get(mod, None)
        print store
        if store != None
                entry.pop(mod)
                entry[rep] = store
                print entry.get(rep)
        else:
                print "Key is already equal
        modentries.append(entry)
```

Figure 1: An example of a syntax error in Python. Shown here is a colon missing from the end of the line.

Methods

- Python syntax errors were searched for manually by looking at commits on GitHub and also with a program that looked for keywords in the commit message.
- The data was uploaded into Weka and used to train a machine learner to decide whether a change was a syntax error.

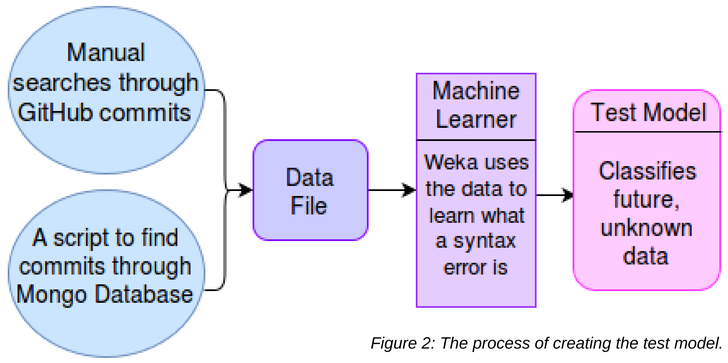

Figure 2: The process of creating the test model.

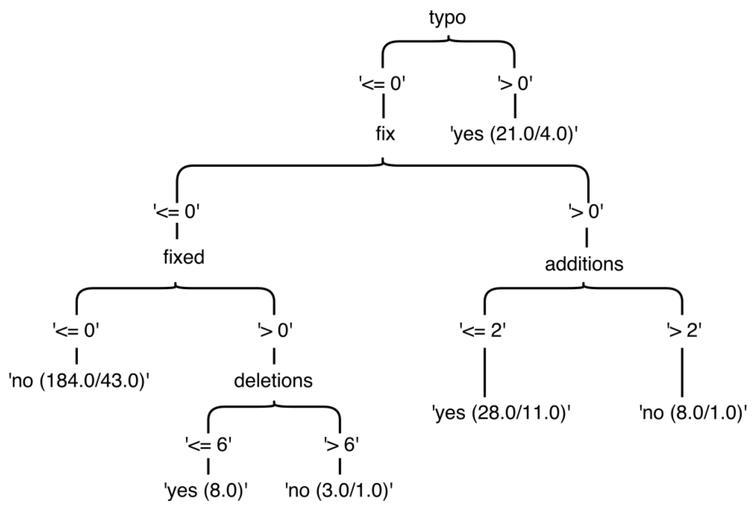

Figure 3: The decision tree that the J48 algorithm uses to determine if a commit has syntax errors.

Findings

- Out of all the algorithms tested, the most accurate was NaiveBayesMultinomial.
- It finds 65% of mistakes in a given set of commit data from GitHub.
- These results indicate that the commit message is not an adequate indicator of whether a commit contains a syntax error.
- The data collected could be used to improve detection of the more common types of syntax errors.

Algorithm	Percent Accuracy on Training Data	Percent Accuracy on Test Data
J48	71.1207%	55.00%
NaiveBayes	69.36966%	60.00%
NaiveBayesMultinomial	72.8448%	65.00%
IBK	54.3103%	50.00%
OneR	71.9828%	55.00%
JRip	75.0000%	55.00%
Random Forest	70.6897%	50.00%

Figure 4: Percentage of accuracy of different algorithms on training and test data.

Acknowledgements

The researcher would like to thank:
- WISEST Summer Research Student Eisha Ahmed, HIP Student Monica Bui, Principle Investigator Dr. Abram Hindle, Direct Supervisor Joshua Campbell
- Sponsor NSERC Promoscience
- The WISEST Student Summer Research Program 2016

Citations

Some images for this poster were created in collaboration with lab partners Eisha Ahmed and Monica Bui.