Stackbot

Julian Ball[1], Ankush Patel[2], Chase Brunette-Pak[3], Michaela Shoffner[4], and Sirajum Munira[5]

***Abstract*— StackOverflow is used widely by computer scientists for sharing code so that others may help with debugging, improving efficiency, styling, etc. This project intends to integrate StackOverflow into live programming rather than searching for posts manually. Source code will be scraped from StackOverflow, organized, labeled based on error types, and then be fed into a machine learning model to find patterns between source code and specific labeled errors. The model then can be added to a GUI that will work alongside the programmer's text editor or IDE and, in real time, search StackOverflow for labels that are predicted based on the source code the programmer is writing.**

I. MOTIVATION

Many programmers find themselves looking to StackOverflow for advice/solutions to specific programming issues. Having a tool that could guess what issues you may be experiencing while you are coding may be useful for programmers who would otherwise be searching for issues on StackOverflow anyways.

II. PROCEDURE FOR DATA GATHERING AND DATA USAGE

A. Web scraping of StackOverflow.com

The Python module called Robobrowser which is built on BeautifulSoup4 will be used to scrape StackOverflow.com. A Python program that the team has previously written uses this module and can scrape StackOverflow for source code on any language specified. The team will focus on the programming languages Java, C, and Python to train the machine learning models on. This program will be run inside of a tmux session on a server hosted at UTK so that the scraper can run as long as it is necessary to keep gathering data. As soon as a question or answer is scraped, the source code will be directly pipe lined to a server holding a Mongo database and inserted into a collection named on the source code's language. It is important to note that many various datasets will be involved in the process of creating the model, thus it is important to organize the various data into one cohesive format. Therefore, there will be a general script in the background that transforms a base set of datasets (related to StackOverflow). Once a cohesive format is obtained - i.e. labels are coherent among all questions/answers - will be able to be used the machine learning model. Our hope is to automate this data collection process so that our models can be retrained periodically and provide the best user experience in regards to bug searching.

B. Organization of the Data

Once the source code is put into collections specified on the programming language, shell scripts that have already been written will be run on the source code. Shell scripts have only been written fro Java, C, and Python, however, the team will create as many shell scripts needed for other languages. These shell scripts will attempt to compile the code. The output of these shell scripts will be parsed for errors and generalized so that a one to two word label can be generated for each document within the collections. A document will simply be source code from the web scraper. The label generated by the shell scripts will be inserted into the document of source code as a string. If the shell scripts do not generate errors (the code compiles successfully), then the code that is a question (bad code) will be labeled with "logic-error". Code that is an answer will be labeled "good" if the shell scripts successfully compile, however, if they cannot then they will also be labeled with the compiler error that was generated even though the code was an answer. These labels generated will then be used for the training stage.

III. RESPONSIBILITIES OF EACH MEMBER

Specific responsibilities of each member are outlined below; however, there are a core set of expectations that will be of consideration to each member. Each member will be responsible for gathering datasets related to Stackoverflow programming questions/answers. Furthermore, they should be able to provide correct licensing information for each data set that they provide. In addition to gathering datasets, each member will be responsible for presenting the data in such a way that it conforms to the layout of the mongoDB database. In other words, there should be a collective script that is designed in part by each member such that all data pipe lined through the script will be transferred in such a way that conforms to standard input of the model. Additional responsibilities include but are not limited to notifying each member of individual progress, creating/closing issues on the repository, and being present at group meetings.

- Julian Ball will be responsible for data gathering by improving the data scraper so that it is pipe lined directly to Mongodb. Furthermore, he will be responsible for cleaning and maintaining the data on the mongoDB database. He will also be responsible for experimenting with different machine learning algorithms.
- Ankush Patel will be responsible for designing a plugin on top of Atom that will utilize the Stackbot itself. This plugin will enable automated usage of the model and connect to related Stackoverflow issues. Furthermore, he will help label the data on the mongoDB database.
- Chase Brunette-Pak will be responsible for automating the scraping process and connecting it to the mongoDB database.

- Michaela Shoffner will be responsible for automating the machine learning model (re-training) periodically based on data coming in from the scraping pipeline.
- Sirajum Munira will be responsible for filling in the gaps between the processes themselves i.e. transferring the mongoDB scraped data to the machine learning model. Furthermore, she will be responsible for maintaining the overall structure of the data transformation script.

IV. TIMELINE

- 9/29-10/6: Setup mongoDB database with 5-10 collections of programming languages with labeled source data. At this point the labels are relatively simple i.e. Question and Answer.
- 10/6-10/13: More labels will be generated based on how the code compiles. The labels may include segmentation errors, logic errors, and/or syntax errors.
- 10/13-11/10: Ankush begins work on the plugin (GUI) that will be integrated with Atom. The rest of the team will begin experimenting with which machine learning models will be best for training on the data that we gathered from the prior weeks.
- 11/10-11/24: At this point, we want to automate the data gathering and training process for our ideal model so that the models can iteratively improve. Furthermore, the model should be sufficient and capable enough to integrate within Ankush's plugin interface on Atom.

V. EXPECTED OUTCOME

By the end of the semester, the team should have a few different machine learning models trained on all of the data gathered and a GUI that can run alongside the user's programming session. The team will also have developed programs that streamline the data gathering and training process so that it will be fully automated. The team will have models that learn iteratively meaning that as data is gathered, the models will be able to learn from it periodically so that the programmer does not have to manually do this. The GUI will run through Atom and therefore be able to run on any operating system that can run the IDE Atom. The end product will be an Atom plugin that can be downloaded and integrated into the programmers instance of Atom.