Stackbot

Julian Ball[1], Ankush Patel[2], Michaela Shoffner[3], and Sirajum Munira[4]

December 10, 2019

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 tags, and then be fed into a machine learning model to find patterns between source code and specific tags. 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.

1 MOTIVATION

Many programmers find themselves looking to StackOverflow for advice/solutions to specific programming issues. Having a tool that could find tags related to source code may be useful for programmers who would otherwise be searching for issues on StackOverflow anyways.

2 PROCEDURE FOR DATA GATHERING AND DATA USAGE

2.1 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. In order to avoid being IP banned, the team will create another web scraper that scrapes available proxies from freeproxylist.net and feed that list to the StackOverflow scraper. The team will focus on the programming language Python3 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 MongoDB database and inserted into a collection and 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 benefits of iterative learning.

2.2 Organization of the Data

Once the source code is put into collection, Python scripts that generate abstract syntax trees (AST) from source code will be run and the AST in the form of a string will be inserted into each document corresponding to the source code that it came from. If the script fails then this means that the source code had a compilation error and will consequentially not be trained on.

3 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 helping gather data related to StackOverflow programming questions/answers. If other data sets are available outside of the scraped data they should be added to MongoDB. Furthermore, they should be able to provide correct licensing information for each data set that they provide. In addition to gathering data sets, 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 if an acceptable model is made by the end of the semester. 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. Other than designing a plugin, Ankush will help find a suitable model for training.

- 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.

4 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.

5 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.

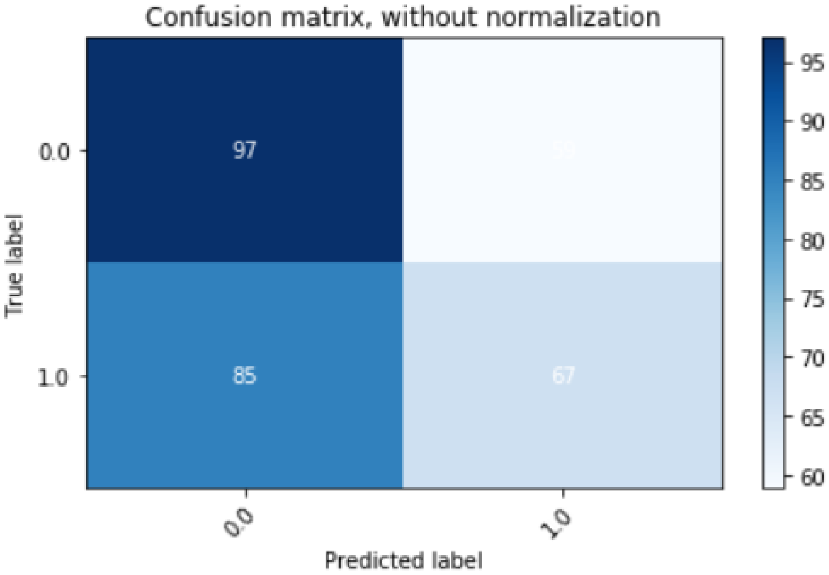
6 RESULTS AND FUTURE WORK

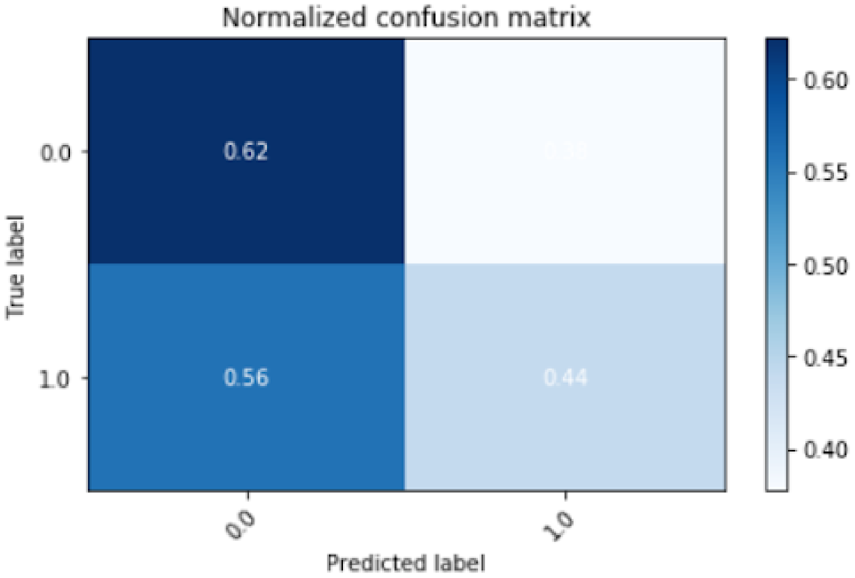
No acceptable model was produced from training on ASTs and tags. As a result, the team did not reach the phase of creating an Atom plugin. During the phase of gathering data, our team had not figured out how to avoid the issue of rate limiting and IP banning. In the final phase of the project, the team had figured out how to rotate IP addresses within the scraper so that more data could be collected at a faster rate. With only 2608 pieces of source code and 2341 unique StackOverflow tags, there is no way to expect a model to learn anything. The results showed this with the best model out of 10 models having less than 1 percent accuracy.

However, classifying whether a piece of code is in a question or an answer is more feasible. The team decided to train a model that will try and decide this when given an AST of python source code. Out of 10 models trained, the best model was only 53 percent accurate. With binary classifiers, this accuracy is also not helpful since it is only as good as random guessing.

Ultimately, the issue was not how the models were trained or the type of models themselves chosen, but the lack of data available to them. In order to improve the results, the first step would be to consistently increase the size of the data set iteratively and at each iteration create models that can be analyzed for improvements or deterioration.

The confusion matrices of the binary model are shown below.





7 APPLICATIONS

One application of the model would be to integrate it within an IDE so that it can suggest related StackOverflow posts to users when requested. The team will set out to develop an Atom plugin that would provide this functionality since the team has experience with Atom and making plugins within it. A user friendly plugin that runs within Atom could be very useful if the tool were accurate at determining related tags. In addition to creating the tags, it would display a list of links to these questions and allow them to be sorted as StackOverflow does. It would also show how many answers are posted on the questions.