Andrew Malta’s CPSC490 Project Proposal

Advisor: Dragomir Radev

There exists a vast amount of source code publicly available on Github; however, there do not exist many easy ways to search for relevant code to particular tasks. In this project I plan to build productivity tools for programmers and computer scientists, leveraging source code found on Github. First, I plan to identify repositories within a particular area of computer science; for example, repositories that deal with natural language processing and related fields. Professor Radev has collected a set of resources on the following topics, which can be found at the All About NLP (AAN) database. I plan to use these repository links in the resources tab of AAN to acquire the source code to perform analysis on. Using this code, this project will build an end-to-end system that given a particular topic, potentially represented as a set of keywords, return a number of relevant code snippets that demonstrate how the specified topic can be implemented in code. If time permits, I this project will also explore various static analyses on the code I collect to build models allowing programming editors to do auto-completion at the line and multiline level.

Some of the initial challenges will be to figure out how to interface with Github’s API to collect the code. This would include figuring out how to download a large volume of source code without making it look malicious to Github’s rate limiting services for its API. One course of action could be to explain my situation to Github support and see if they would grant me a temporary increase to my base rate limit. Another course of action that seems more promising would be to find archives of repositories hosted on mirror sites to download and parse locally. This would require some additional code to unzip the archives, recurse through the directory structure of the repository, and identify the source files I would like to save for analysis. Once this is done I can move onto methods for indexing and searching this code.
After the code is collected, I will need to do some static analysis to extract variable names, function names, and some overall statistics about the code. There are a number of static analysis libraries, especially for popular languages like Python, which fortunately happens to be the main language of implementation across the repositories posted on the AAN database. In addition to looking at the code for featurization, it might be helpful to also look at the README in the Github repository to understand if the code is applicable to the given task at hand. These later NLP tasks I plan to run by my advisor, Dragomir Radev, for suggested techniques.

After the featurization of the code, and potentially some surrounding contextual information about where the repository that the code was pulled from, the next step would be to creating a scoring metric to sort and display the results. I believe that some unsupervised learning could be useful here to group similar code based on the features I choose. This would constitute the last step of the original problem I posed, and I expect to explore a few methods for performing these last few steps to get some better qualitative performance.

As I stated earlier, if time permits I would also like to explore using the code I collect, and potentially some more that I can take from popular repositories on Github to build source code language models that could be useful for a number of reasons. One such reason that I alluded to would be to extend auto completion in programming editors to be able to predict and recommend common lines and sets
of lines of code. While this application is certainly different from the first task I stated above, they are both rooted in the same data and geared towards a similar goal of enhancing productivity of programmers and computer scientists.

Lastly, the deliverables of the project will include a set of code datasets published to Github for others to use for various purposes, a module that allows easy collection of source code from Github, a module that encapsulates the static analysis and scoring for the code search given a set of repositories, and, finally, a blog post describing the techniques I used throughout the completion of the project. Any additional work I discussed in this proposal will be completed if time permits.

References

