Nicholas Duncan
Ethan Holen
Andre Leautaud
October 21, 2021

MapReduce Analysis of Github Commit Data

Introduction

We propose to analyze GitHub commit messages across multiple different programming languages. We assume that a significant amount of information can be gleaned about programmers and the languages they use by analyzing the documentation (or lack thereof) that they provide when contributing their code. Initially we plan on running a toxicity detector against each commit message to see what links we can find between expletives and the languages that they are used in (also to find funny commit messages :)). We believe this will give us a great starting point to begin extracting data from this set.

Along with the expletive analysis we will also be checking the length of commit messages. This might also give some valuable information in seeing which projects and languages value more descriptive (or at least lengthy) documentation or concision. There is also the opportunity to analyze any interactions between collaborators and members of a team working in a repository; the interactions we'd be interested in occur within Github's issue/pull request method of workflow organization and collaboration.

This project could offer insight to companies looking to find bugs or challenging areas in their frameworks and technologies. The more expletives and annoyed commit messages from developers in a certain area, the more they might be irritated with the programming language, technology, or team. Importantly, this information can be extrapolated without relying on direct developer feedback and instead looking at the general trends of annoyance in different areas of projects. These trends have been shown to be useful in terms of predictive analytics in gauging the effectiveness of a team based on early signs of cohesiveness (Gitinabard, p.6).

The problem at hand readily exemplifies the 3 Vs of big data. A company of considerable size will have many repositories acting to organize the workflow of various interrelated projects, each with its own team of programmers, with each programmer working through multiple tasks each day consisting of many commits, issue discussion, and pull request communication. Anyone looking to use this activity on a continual basis would expect a constant flow of data, especially so for the case of those most likely to be looking to provide a solution as a service in respect to this problem. Corporations are always looking for ways to improve and predict the performance of their workers as well as identifying active problem areas or identifiable patterns of problem areas. There are many corporations that already outsource to services that provide a constant analysis of similar statistics that they can then report on.

Procedure

The data exist in a repository separated by hour. We will first fetch the json data and then use a script to translate it into a useful format. We will also design a scheme to store the data.

Our procedure for the generation of relevant statistics will include an analysis of the GitHub activity of user's public repositories in respect to certain quanta. These will include: the appearance of expletives (or imprecates), the average length of commit messages, and average time from push request to inclusion.

We will calculate these in MapReduce. The averages will be found using the summarization design pattern and a combiner. The expilcitivity will be found using the IBM MAX toxic comment classifier in the mapper. Ultimately these quanta will be normalized against the volume of code relating to each programming language across the whole dataset.

Similarities between these quanta in respect to the languages will be calculated by aspects of cosine similarity, Euclidean distance, and purely magnitudinal differences in any relevant dimensional space. Furthermore these statistics may be calculated with weighted imprecates that will reflect the relative severity of their use, where some imprecates are considered more vulgar than others. With the $\mathbb{R}^3$ space offering a good basis for the calculation of weighted variables as well as room for dimension reduction through projection for the calculation of statistics based on a subset of the variables.

Dataset

The dataset we will be using for this project is the GitHub Archive. This is a database which records the GitHub timeline formatted into distinct events and makes it accessible for download. The data provided for commit messages is in json format which we should be able to easily translate into any format we need for processing in MapReduce. The size of the data is around 100GB since it includes activity since 2015. We will take a selection of around 2GB.

Missing data should not be a problem as we can simply set the message to length 0 and either discount 0 values on messages or take them into account in our analysis. Regardless, the missing data should not make the data processing any more challenging. It may also be interesting to see which projects or languages have the most length 0 or missing commit messages in their projects.

Timeline

Week	Task	Team Member
9	Proposal	All
10	Decide on data input formatting and build parser (JSON→plaintext)	All
10	Implement data aggregation framework	Nicholas
10	Generate data storage/access scheme	All
11	Update procedures in report	All
11	Build job handling automation for mass processing	Nicholas and Ethan
11	Develop and implement relevant algorithms for first stage calculations	Andre and Ethan
12	Finish data collection	All
12	Graph results for visualization	Andre and Nicholas
13	Include results and conclusion in report	All
13.5	Polish report if necessary	All / None

Bibliography

Git. (n.d.). *GH Archive*. GitHub. Retrieved October 21, 2021, from http://www.gharchive.org/.

Gitinabard, Niki, et al. "Student Teamwork on Programming Projects: What can GitHub logs show us?." *arXiv preprint arXiv:2008.11262* (2020).

IBM. (n.d.). IBM/max-toxic-comment-classifier: Detect 6 types of toxicity in user comments. GitHub. Retrieved October 21, 2021, from https://github.com/IBM/MAX-Toxic-Comment-Classifier.

Milovidov, A. (2020). *GHE Clickhouse*. Everything You Always Wanted To Know About GitHub (But Were Afraid To Ask). Retrieved October 21, 2021, from https://ghe.clickhouse.tech/#repositories-with-the-maximum-amount-of-pull-requests.