CS 410 Final Project Report

Captain: Aishani Pal - aishani2 , Alyxandra Merritt - merritt9
December 9, 2021

Overview

The main goal of our project is to analyze the emotions of GitHub commit comments associated with a person(s) over time for a single project(s). While this is one specific metric that we chose to analyze, the tool we have built can expand to other areas. For example, one can look at commit comment sentiment across time for specific languages or size of project (determined by numbers of lines of code). With the collected data, personalized sentiment tool, and graphical outputs, a project manager can use the analysis to better manage workflow and team dynamics. If he/she notices team members displaying more negativity during certain days of the week, the manager can encourage team members to focus on other non-code related work.

The paper our project is based on is included in this repository.

Demonstration Video

https://mediaspace.illinois.edu/media/t/1_t06bkpvd

Software Implementation

Data Collection

All data is coming from GHTorrent's MSR 2014 Mining Challenge Dataset, which includes data from the top-10 starred software projects, for a total of 90 projects and all their forks. After downloading the SQL dump and saving it locally, running `data_script.py` will run the required SQL queries to generate the data we need for this project in the appropriate format for our software.

Sentiment Analyzer

All code for the sentiment analyzer exists in `sentiment_analyzer.py`.

Data Preparation

We mostly followed this tutorial. We completed the following steps to prepare our data for the sentiment model.

1. Tokenize
2. Part of Speech tagging

3. Removed excess noise (i.e. removed stop words)

NLTK Pre-Trained Model

We used the NLTK's Vader sentiment analysis tool, which uses a bag of words. The tool gives 3 scores for negative, neutral, and positive. To achieve 1 sentiment tag for each github comment, we took the maximum of the three values and assigned that sentiment to the comment.

Baseline

It was extremely important to have a way to understand if the sentiment analysis tool we used was comparable with that of the paper we have based our project on. Thus, we ran a metric the paper examines using our tool. We chose to replicate the metric comparing emotions to the time of the day (section 3.3). We successfully got the same results as that of the paper, which demonstrated that regardless of the time of day, users tended to have more neutral comments than negative or positive.

Analyzing our Metric

Our main project goal was to analyze the emotions over time of one person on one project. This was our unique metric and extension of the paper we have been referencing. Once all the comments were tagged with their corresponding sentiment, the data for each comment was outputted to separate files based on the user_id of the author. This allowed us to group the data by user and create plots to analyze the trend of sentiment over time for an individual person. As the repositories from the dataset are quite large with many contributors, we had a variety of outcomes. For users with a small number of comments, like users 15246, 252614, and 4024, the results were inconclusive as there wasn't enough data to see a clear trend. For users that had a relatively large number of commits on the project, like users 506780 and 9883, it is easier to see some variation over time. The majority of comments for a particular user do tend to be positive or neutral, similar to the baseline analysis, but we did not find there to be a general trend over time.

Further Improvements

If this project was to be continued in the future, we would like to explore running our sentiment analysis on a larger portion of the dataset. In our proposal and throughout the implementation of the project, we focused on analyzing the emotions over time of one person on one project. This could be expanded to consider multiple projects or comparing multiple users on the same project. It would also be interesting if we could scrape data from personal GitHub repositories in order to perform this analysis on our own projects.

Software Usage

We have already prepped the data needed to run the sentiment analyzer tool for this project. There is no need to download the SQL dump from GHTorrent. You can ignore the `data_script.py` for the sake of testing.

1. Clone the repo and ensure that python3 is installed. After cloning, cd into GithubCommentsSentimentAnalysis.

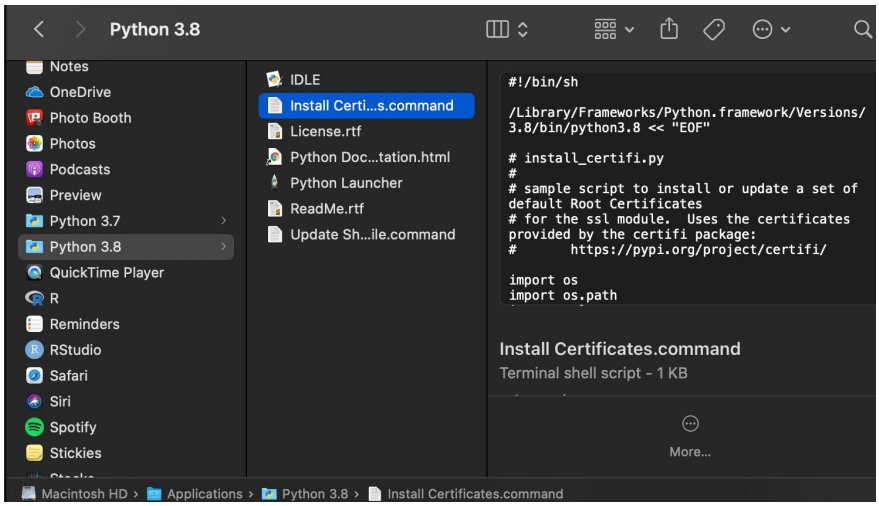
2. Run `pip3 install -r requirements.txt` to ensure you have the required packages to run this project.

3. To test the sentiment analyzer, run `python3 sentiment_analyzer.py`. The outputs of this script are all the comments with their timestamp and corresponding sentiment (saved in the sentiment_data/ folder) and the baseline graphs (saved in the plots/ folder). You will notice that each baseline graph shows significantly more neutral comments (than positive or negative) for any time of day, similar to that of the paper this project is based on (included in this repository). This tells us that our sentiment analyzer tool is similar to that of the paper's and we have successfully written our own version.

    a. You may see the following error message

```
[nltk_data] Error loading punkt: <urlopen error [SSL:
[nltk_data]     CERTIFICATE_VERIFY_FAILED] certificate verify failed:
[nltk_data]     unable to get local issuer certificate (_ssl.c:1108)>
```

To resolve this, navigate to your Python applications folder and run the `Install Certificate.command` (you may have to do this for all your Python3 versions).



4. To run the script that allows for the analysis of our sentiment results for users over time, run `python3 data_analysis.py`. At the top of this file, one can specify an array of user_ids that have contributed to the project with id 289. For each user, the script will graph the sentiment over time. The output of this script is the saved graphs in the plots/ folder for each user following the naming convention "user{id}.png".

Team Members Contribution

Both Aishani and Alyx worked on creating the sentiment analysis tool including data preparation with steps such as POS tagging and removing stop words. We also worked together to use the nltk toolkit and run the sentiment analysis on the comments. All of the reports and final video were completed together as well.

For our separate responsibilities, Aishani primarily worked on the data collection by writing SQL queries to populate the csv files. She also ran the baseline analysis which organized the data to fit the paper's metric. Alyx primarily worked on grouping the comment data by users and analyzing the trends over time which serves as our unique metric for the final project.