Group Project Proposal

Team Composition

Andrew Berg	Raidel Hernandez	Tyler Kelly
FSUID: aab13j	FSUID: rh13k	FSUID: tck13

Assigned Paper

- Sentiment analysis of commit comments in GitHub: an empirical study

Programming Language(s):

- Python
- Java
- SQL

Project Plan *(Team-member Responsible)*

- Sentiment Analysis ***(Andrew)***
    - Setup core SentiStrength utility for finding the sentiment scores of the rest of the project
- Emotions In Commit Comments ***(Andrew)***
    - Figure 1: emotion score average per project along with the graphic
    - Figure 2: proportion of positive, neutral and negative comments per project
- Emotions and Programming Language ***(Raidel)***
    - Table 2: Emotion Score average grouped by programming language
        - Calculate the average emotion score of all programming languages
            - Determine projects base programming language. Determine average sentiment of commit messages for said project. Do this for each project analyzed then properly group projects by their language and average sentiment, and then average all the sentiments for a particular group.
- Emotions, Day and Time of the Week ***(Raidel)***
    - Table 3: Emotion score average of commit comments grouped by weekday
        - Group all of the commits into separate categories, each representing a different day of the week, when the message/commit was written.
        - Calculate the average emotion score of commit messages grouped by weekday.
    - Table 4: Emotion score average of commit comments grouped by time of the day.
        - Analyze timestamp and emotion score for each commit message.
        - Separate the commits by time of day:
            - Morning - [6:00-12:00)
            - Afternoon: [12:00-18:00)

            - Evening: [18:00 - 23:00)
            - Night: [23:00-6:00)
        - Print the average emotion score for commits by time of day committed.
- Emotions and Team Distribution ***(Raidel)***
    - Figure 3: Relationship Between continent distribution and positive emotion score average (only one project is distributed in 2 continents).
        - Study the relationship between the emotion in a commit message, and the geographical distribution of the projects.
        - In the project they found no 'significant correlation' between the emotion score of the commit message and the number of countries/continents in which each project was distributed.
        - Use Spearman correlation between the average of all positive comments.
- Additional Research Question ***(Tyler & Andrew)***
    - For users with personal projects, who have made commits, we will examine said projects, and determine their overall code quality. What will a users sentiment on a git commit message reveal about the quality of their personal open source projects? Will this contributor even have personal coding projects?
    - This question relies on two main data points being defined/obtainable. One, can we link a committer to one of the open source projects, to a user with their own set of personal projects? A possible approach, on a central repository database like github, would be to identify the commiter's userid, and then attempt to look up that userid's account, and analyze their personal projects (to be more precisely defined later, but for now can be defined as projects in which they are the major contributor).
    - Two, once we obtain this link between committer and user, will we be able to properly analyze the quality of their personal projects if they exist? Our code analysis will be strictly static, as dynamic analysis would be beyond the scope of this course. There is a large number of static analyzers available to us, as seen here. We will need to determine the most prominent languages that we wish to examine, download their respective analyzer, and install it properly on the machine.
    - If a language does not have a convenient tool to properly analyze its code quality, we will need to look at more traditional methods to determine the quality. One possible approach would be to simply analyze the ratio of comments to code in a project, and possibly examine the length, and complexity(length, markdown techniques) of the Readme as this can often be a good tell for the quality of a project.
    - Candidate static analyzers:
        - C:
            - https://github.com/dspinellis/cqmetrics
        - Ruby:
            - https://github.com/whitesmith/rubycritic