COSC448 Directed Studies - Reproducibility Package:

About:

This document provides a concise overview of my project and outlines the process for replicating our test outcomes using the materials included in the reproducibility package. The study builds upon Jasmin Mishra's honours thesis project, accessible at: https://github.com/coffeehousejazz/honours_thesis.git.

Building upon the findings of Mishra's project, I undertook an investigation into two research inquiries that were highlighted in the concluding sections of her thesis. To be more specific:

RQ1: Does communication matter? Does the number and type of comments that a mentor and mentee have on each other affect the success of a relationship?

We decided to carry out a sentiment analysis of comment text for all comments made on each project in the database. Then we analyzed different types of sentiment scores for each of the successful and unsuccessful mentorships to check for any correlation. Sentiment analysis included the following:

- The sentiment analysis of comment text for base emotion type scores - anger/disgust/fear/joy/neutral/sadness/surprise (using: https://huggingface.co/jhartmann/emotion-english-distilroberta-base)

- The tone analysis of comment text for general tone scores – positive/neutral/negative (using: https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment)

We also wanted to check whether the number of questions in comments had any correlation to a successful mentorship.

- Number of questions in the comment text (question/statement scores) (using: https://huggingface.co/shahrukhx01/question-vs-statement-classifier)

RQ5: Does guiding documentation matter? Do pull request labels guide mentees? Are pull request labels helpful in success?

We mainly wanted to check whether coherence and conciseness in pull request descriptions benefited the success of the mentorship. We decided to carry out a text analysis of the pull request title and body text for all pull requests made for each project in the database. Then we analyzed different types of pull request clarity scores for each of the successful and unsuccessful mentorships to check for any correlation. The analysis included the following:

- Pull request title text analysis for gibberish detection - clean/mild-gibberish/word-salad/noise scores (using: https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457)

- Pull request body text analysis for bugfix/non-bugfix scores
  (using: https://huggingface.co/neuralsentry/starencoder-git-commit-bugfix-classification) and gibberish detection scores

This was done in 3 steps:

1. Matched each mentor and mentee to success or failure based on the results of the thesis project. (let's call this 'users db').

2. Extracted comment body text and pull request title and body text for each project from data. Cleaned up text data by removing any special symbols, hyperlinks, paths, extra spaces and other ungrammatical portions of text. We then ran each block of texts through carefully chosen text analysis models published for free use on the internet, received scores for each text from different models and added the scores to each dataset. ('comments db' and 'pulls db').

3. We matched the comment and pull request data from 'comments db' and 'pulls db' to 'users db' based on the user (mentor/mentee) id and used Tableau to visualize our results.

Step 1:

- Save files 'data.users', 'data.comments', data.pulls' to the downloads folder on your computer.

- Execute file 'GSOC2021.ipynb' to store files 'success.csv', 'unsuccessful.csv', 'success_mentor.csv' and 'unsuccess_mentor.csv' to the downloads folder on your computer.

- Execute file 'GSOC2022.ipynb' to store files 'success22.csv', 'unsuccessful22.csv', 'success_mentor22.csv' and 'unsuccess_mentor22.csv' to the downloads folder on your computer.

- Execute the 'Mentorship Success Data Extraction.ipynb' file, to combine data from the preceding two stages and employ it to classify each user in the 'data.users' file as either successful or unsuccessful. take a random sample of 285 successful users and 285 unsuccessful users. Lastly, it takes another random sample of 50 mentors and 50 mentees from the previous sample in file 'book1.xlsx'.

Step 2:

- Extract comment body text, pull request title text and pull request body text from files 'data.comments.xlsx' and 'data.pulls.xlsx' and store them each in separate files. Clean up all text data by removing any special symbols, hyperlinks, paths, extra spaces and other ungrammatical portions of text.

- Utilize the 'huggingface text analysis.py' file to process all the text content from each file using the Hugging Face models. The file itself contains the model names for each specific model to be employed. To execute this file successfully, ensure that you have the necessary packages such as pandas, transformers, datasets, and pathlib downloaded and installed.

- You will receive test scores in the file ‘results.csv’ for each line of text in the file. Process the test score data to have a column name for each score label and just a numeric value under the labels for each line of text. Add these columns back to ‘data.comments.xlsx’ and ‘data.pulls.xlsx’. The order of results should match the order of comments and pull requests such that each score accurately pertains to the correct comment or pull request.

Step 3:

- Navigate to the website https://public.tableau.com/

- Create a new free account if you do not already have one and create a new project.

- Import data files ‘data.comments.xlsx’, ‘data.pulls.xlsx’ and ‘book1.xlsx’ in the data source section. The data will be saved as tables. Create relationships between the tables: relate the ‘Login’ field in the book1 table to the ‘user.login’ field in data.pulls table relate the ‘Login’ field in the book1 table to ‘User.Login’ field in the data.comments table

- On a new sheet, add the column ‘success’ (under book1) to the column section. Also, add the column names for the different test scores you want to check for success correlation for comment body data, pull request title data or pull request body data. Then in the rows section, add the same column names again, but sum all the scores. This gives us the ultimate label score for successful mentors and mentees and for unsuccessful mentors and mentees.

By following these steps, you should be able to successfully reproduce our findings.