Guidance and Communication in Mentorship Relationship - Research Project

Directed Studies Course COSC448

Akshaj Srinivasan

Supervisor: Dr. Gema Rodríguez-Pérez

University Of British Columbia

Introduction

In this endeavour, my focus will be on exploring the factors behind a successful mentor-mentee relationship within the realm of Open-Source Software. At the core of this project lies the primary goal of conducting textual analysis. This analysis will be based on a dataset that encompasses details about instructional documentation and the ensuing conversations that transpire during mentorship interactions. The results possess the capacity for future application, offering valuable conversational indicators that can be identified early in the mentorship process, aiding in the prevention of mentorship breakdown.

Background

The study builds upon Jasmin Mishra's honours thesis [1] project on mentorship relationships, accessible at: https://github.com/coffeehousejazz/honours_thesis.git. In her thesis, she used data from open-source software development projects which incorporated mentorships. The data was collected from Google Summer of Code projects hosted on GitHub during the years 2022 and 2021. Only certain participating organizations were selected based on popularity.

The mentor/mentee relationship data from the projects was mined using Python and the GitHub REST API. It made use of the user, issues, and comments APIs. The GitHub API was located on the Google Summer of Code website and calls with it require the mentee username, mentor username, owner name, and repository name. The publicly available data about the mentors and mentees was obtained via the user API. All closed pull requests and issues in the repository that the mentee was mentioned in or authored were obtained using the issues API. All the comments from the issues and pull requests that were acquired using the issues API were obtained using the comments API.

The collected data encompasses information from both mentors and mentees. It also covers closed issues and pull requests associated with mentees, including those they created or were mentioned in. Moreover, the data comprises all the comments from these issues and pull requests. This dataset is structured in JSON format and is stored using MongoDB. The database architecture consists of a single database housing three distinct collections: one for user data, another for comment data, and the last for pull request data. Prior to analysis, the dataset underwent a cleaning and configuration process to enhance its usability and readability.

Drawing on previous research, Mishra identified three key factors that characterized a successful mentorship: the completion of projects, mentee satisfaction, and the involvement of the mentee in the project beyond its completion. To establish her metric for a successful mentorship, she quantified success based on the duration for which the mentee remained engaged with the project following the conclusion of that Google Summer of Code program. In this context, she considered a mentorship as successful if the mentee contributed a minimum of 5 merged pull requests to the project within 6 months after the conclusion of the Google Summer of Code. The selection of 5 pull requests was deliberate, to only focus on highly successful mentorships.

Research Questions

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 type of comments that a mentor and mentee have on each other affect the success of a relationship?

RQ2: Does guiding documentation matter?

Do pull request labels guide mentees? Are pull request labels helpful in success?

Data Sources

To connect to Mishra’s’s database, use the mongoDB Compass or web application with the following connection string and password:

connection string: mongodb+Sav://HonourThesis:@cluster0.no1barz.mongoDB.net/test
password: XZJXwB8NNdHloxGw

From the database, I extracted the ‘data.users’, ‘data.comments’ and ‘data.pulls’ files for the user, comments and pull request data for all projects.

On her project’s GitHub repositories are two Python notebooks named ‘GSOC’ and ‘GSOC22’. I ran these files to extract the lists of successful and unsuccessful mentors and mentees.

Data Collection

I cross-referenced the lists of successful and unsuccessful mentors and mentees with the ‘data.users’ file and assigned a new TRUE/FALSE value of ‘Success’ to each user in the ‘data.users’ file depending on whether the mentorship was successful//unsuccessful. This was done using a Python notebook on MATLAB. I stored the results in an Excel file.

Next, I extracted the comment body text and pull-request title and body text of all projects from the ‘comments.data’ file and the ‘pulls.data’ file to MS Excel files. I Cleaned up text data by removing any special symbols, hyperlinks, paths, extra spaces and other ungrammatical portions of text. I then ran each block of texts through different textual analysis models (mentioned below) using Python and added the resulting scores back to the original datasets (‘data.comments’ and ‘data.pulls’).

Methodology

Regarding the first research question, I decided to carry out a sentiment and tone analysis of comment text for all comments made on each project in the database. I analyzed different types of sentiment and tone scores for each of the successful and unsuccessful mentorships to check for any correlation. The sentiment analysis used the following:

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

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

I also wanted to check whether the number of questions in comments had any correlation to a successful mentorship. This analysis used the following:

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

For the second research question, I mainly wanted to check whether coherence, clearness and conciseness in pull request descriptions benefited the success of the mentorship. I 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 I analyzed different types of pull request clarity scores for each of the successful and unsuccessful mentorships to check for any correlation. The analysis used the following:

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

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

Model Selection

To analyze text, I used different textual analysis models from the website huggingface.com, published for free use. The selection of models was conducted meticulously, taking into account their popularity and the frequency of downloads. For each of these shortlisted models, I also conducted multiple tests to evaluate their accuracy in generating results for the specific types of text that I intended to process. The outcomes from the chosen models exceeded expectations and were highly reliable.

For sentiment analysis of text, I ran 10 preprocessed comment texts through a sentiment analysis model and cross-referenced their sentiment scores with scores from other sentiment models. Some models gave erroneous results due to having too many variables of sentiment, while others did not give accurate results due to a lack of training. In the end, it was a trade-off between the number of sentiment variables and accuracy. In the end, I found that the 'jhartmann/emotion-english-distilroberta-base' model gave the most consistently accurate results.

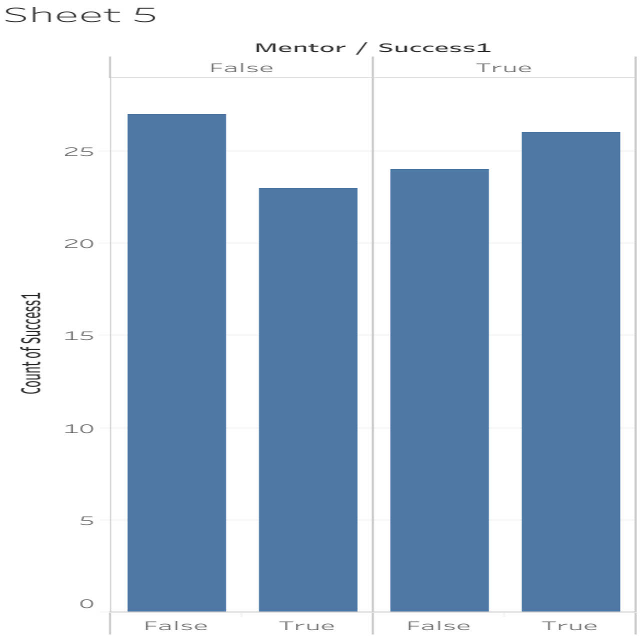
For tone analysis of text, the 'cardiffnlp/twitter-xlm-roberta-base-sentiment' model had been the most extensively trained using a wide array of text types. It was also the most downloaded model for tone analysis. I ran similar tests as I did for sentiment analysis models and it proved to be highly accurate every time.

The hardest model to choose was one for the clarity analysis of text. I wanted to find a model that really could tell whether a comment or title was a meaningful one or not. I defined a meaningful text to be one that is not verbose but also at least somewhat grammatically sound. The best one I could find was the 'https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457' model for gibberish detection. After running numerous different types of text through it, I realized that it was quite sound in its analysis for comment and pull request text. Those comments that the mentee easily understood received good scores and those that the mentee had to ask follow-up questions for received slightly worse scores.

Lastly, the question/statement textual analysis model, 'shahrukhx01/question-vs-statement-classifier', was also an easy selection as the model could tell a question from a statement with 100% accuracy during testing.

Sampling

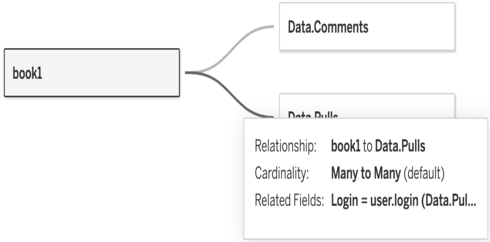
Utilizing the 'data.users' file, which contained labels indicative of success for each user according to their mentorship experience, I used of Python and MATLAB to generate samples. These samples were crafted to ensure a balance in terms of both mentor and mentee representation, as well as an equitable distribution of successful and unsuccessful users. These thoughtfully constructed samples played a pivotal role in my subsequent analysis and visualization efforts, ultimately yielding a more precise assessment when comparing the results of text analysis.



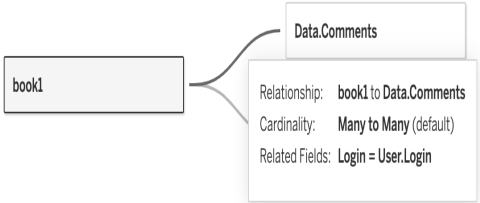
The above is a representation of one of the samples. I will be using data from this sample to explain my findings in the following section. On the Y-axis, is the number of successful/unsuccessful users in the sample. On the X-axis at the top of the graph, the true and false sections are representative of successful users (True) and unsuccessful users (False). On the X-axis at the bottom of the graph, the true and false values indicate whether the bars are representative of mentors (True) or mentees (False).

Results

To visualize my results, I used Tableau Public. I used the three files, ‘data.users’, ‘data.comments’ and ‘data.pulls’ as my data sources by exporting them as tables and creating graphs to display my findings. I created relationships between the tables by relating the login IDs of each of the users to the login IDs of the commenters and pull requesters.



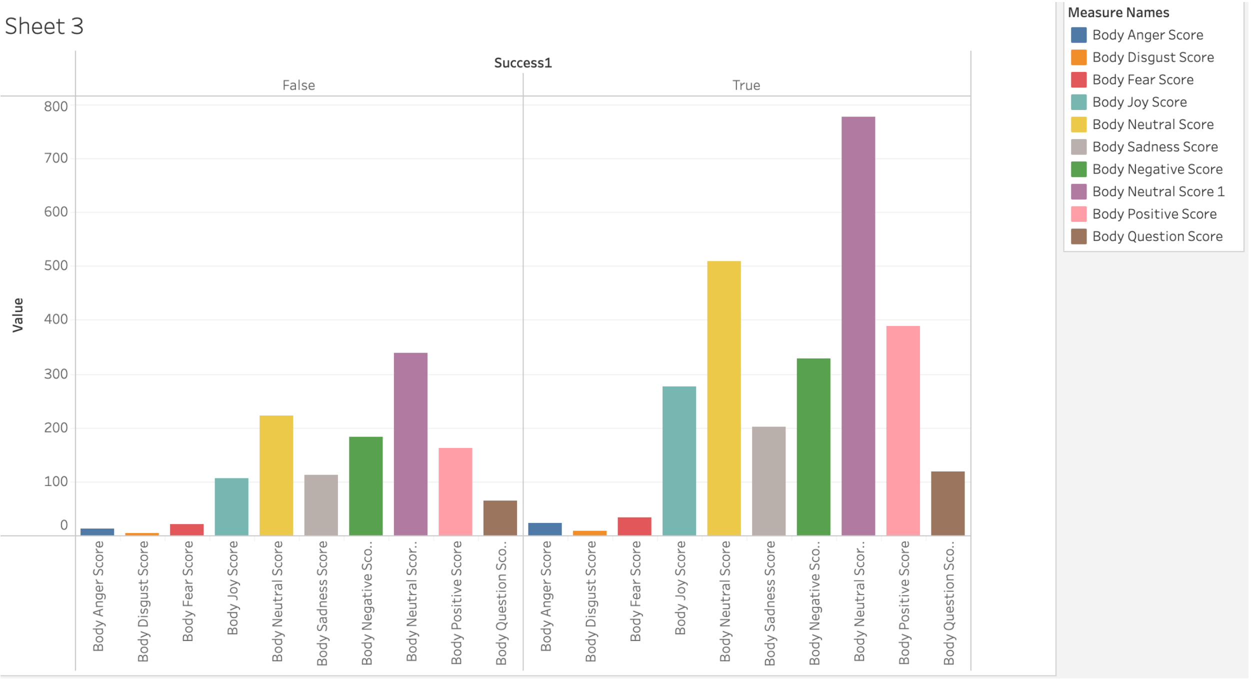
Relationship: book1 to Data.Pulls
Cardinality: Many to Many (default)
Related Fields: Login = user.login (Data.Pul...



Relationship: book1 to Data.Comments
Cardinality: Many to Many (default)
Related Fields: Login = User.Login

‘Book1’ contains the sample data.

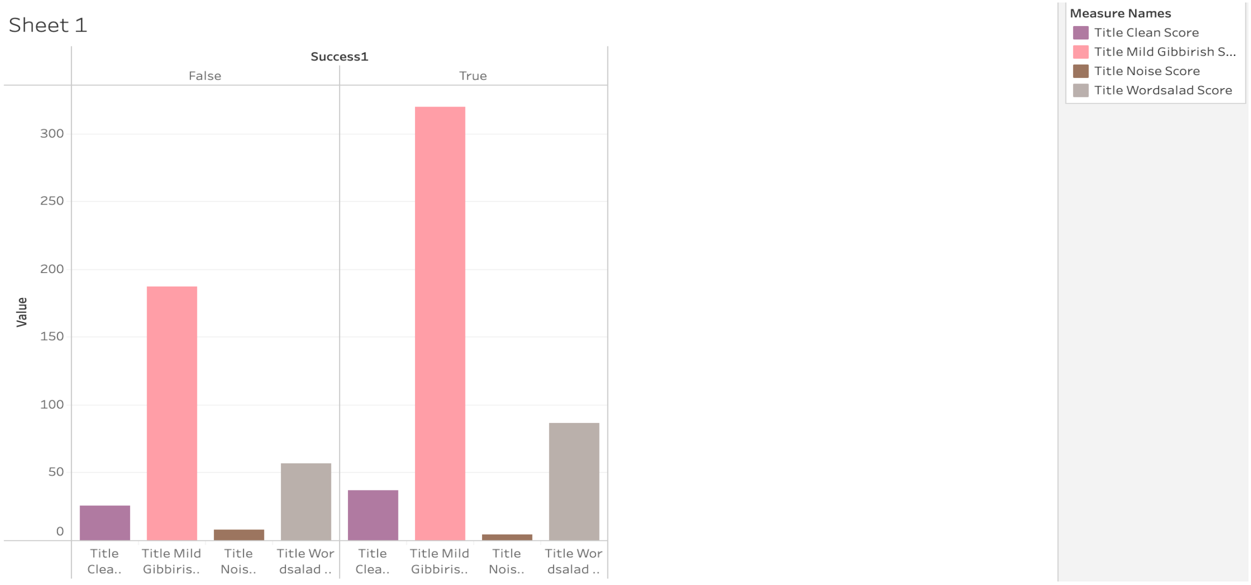
For the first research question, I graphed the summed test scores from all the textual analyses of comment data for successful and unsuccessful users. (Note: ‘body neutral score 1’ results from tone analysis and ‘body neutral score’ results from sentiment analysis)



Across many such samples, I found that results for Successful users produced higher scores for sentiment and tone analysis on average for all sentiments and tones. The graphs followed a similar shape for both unsuccessful and successful users. Since there were an equal number of successful and unsuccessful users in the sample and since the average number of comments by unsuccessful mentors/mentees (23.26/7.77 ~ 31.03) and successful mentors/mentees (13.10/17.11 ~ 30.21)[1] are approximately equal, it can be alluded that the models were able to assign scores with higher confidence to comments by successful users than to those by unsuccessful ones. Other than this, no clear distinction could be made between unsuccessful and successful mentees or mentors across the samples with respect to the different types of emotion and tone scores for comments.

However, the number of questions asked by successful users was always higher than the number of questions asked by unsuccessful users. This was reflected in most samples and rarely were they equal. This shows that successful mentorships always had more questions being asked.

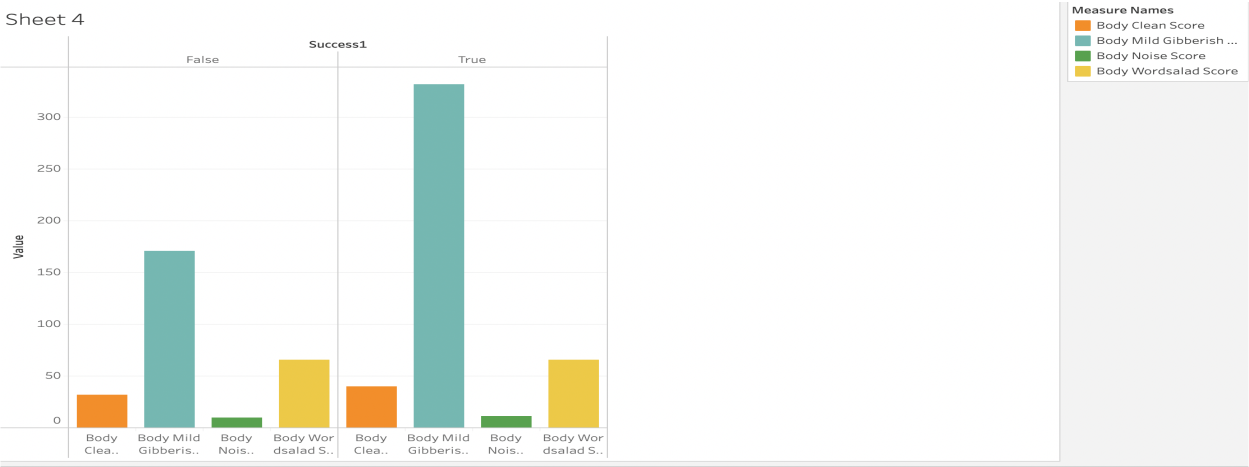
For the second research question, I graphed the summed test scores from the gibberish detection model for pull request title text data for successful and unsuccessful users. (Note: the order of high to low clarity of text is from Clean to Mild-Gibberish to Noise to Word Salad.)



[1] These findings are derived from the research presented in Jasmin's Honors Thesis Project.

We can see that the graph follows a similar trend as before where higher scores were more confidently assigned by the model to pull request titles made by successful than those made by unsuccessful users. Other than this, no clear distinction could be made between unsuccessful and successful mentees or mentors across the samples with respect to the different clarity scores for pull request titles.

I also graphed the summed test scores from the gibberish detection model for pull request body text data for successful and unsuccessful users.



From this, we can see that the graphs follow a similar trend. The only exception is the word salad score for successful and unsuccessful users. The total word salad score for unsuccessful users is higher than that for successful users. The same is reflected by the noise score for successful and unsuccessful users. The values for noise scores are almost the same with the successful users having a slightly higher score. This goes to show that many unsuccessful users had unarticulated or incoherently expressed pull request descriptions.

Discussion

From the results, we can infer the following:

- Models were able to confidently give higher scores across all labels for successful users than those for unsuccessful users.

- Unsuccessful users had unarticulated or incoherently expressed pull request descriptions when compared to successful users.

The second inference may suggest a potential connection between the quality of guidance and documentation offered by mentors or mentees and the overall success of the mentorship. It is plausible that mentors who provide more comprehensive guidance and instructions to their mentees could contribute to more successful mentorships. Conversely, it's also possible that well-supported mentees, benefiting from better mentorship, might exhibit increased confidence in their work, leading to more coherent descriptions of their task resolutions. In either case, this strongly indicates that the presence of clear and comprehensive guiding documentation may play a pivotal role in determining the success of a mentorship.

The initial inference doesn't directly address the issue of mentorship success, and it requires further examination to uncover the underlying reasons. This pattern was consistently observed across multiple samples, not limited to just one instance. One plausible explanation could be errors in the original dataset. For instance, in the 'data.comments' dataset, there might be comments from users not included in the 'data.users' dataset. The same could apply to the 'data.pulls' dataset concerning pull requests made by users who are not present in the 'data.users' dataset. Alternatively, there could be inaccuracies in the relationships between 'data.users' and 'data.comments'/'data.pulls'. However, in both scenarios, we would expect to see some unaccounted-for successful and unsuccessful users. Strikingly, the data indicates that if there are errors, they predominantly concern the underrepresentation of unsuccessful users. Therefore, it is imperative to conduct further analysis and provide a more comprehensive explanation for this pattern.

Despite the possibility of errors being present, the alignment of noise and word-salad clarity scores among the less-represented unsuccessful users with the values seen in the more-represented successful users across various samples provides strong support for the second interpretation mentioned earlier.

Conclusion

Mentorship stands as a critical field of study, with potentially important consequences if executed inadequately. While it's not always possible to perfectly match mentors with mentees and vice versa, we can proactively detect issues early on through textual cues and other indicators to gauge the smoothness of a mentorship relationship. The primary objective of this analysis is to assist organizations in pinpointing factors that can influence the success of mentorship programs. Textual cues within mentorship interactions offer valuable insights into whether a mentorship is likely to succeed or not. Leveraging these cues to adjust the mentorship dynamics can prove to be a time and resource-saving strategy. Guiding documentation and communication is crucial to the mentee's understanding of the task at hand and the type of work to be submitted. Additionally, it significantly influences the level of engagement and enthusiasm that both mentors and mentees invest in their mentorship experiences.

By recognizing the significance of clear, supportive guidance, organizations can institute practices that foster more productive and fruitful mentor-mentee relationships. Ultimately, this can lead to improved learning outcomes, professional development, and overall satisfaction for both mentors and mentees.

References

[1] Mishra Jasmin, Investigating Mentorship Relationships in Open-Source Software, 2022, https://github.com/coffeehousejazz/honours_thesis