CEN5035 – Software Engineering – Fall 2023

Final Project Report

Anonymous Author(s)

ABSTRACT

In this project, we use DevGPT[4] datasets to analyze ChatGPT and developer interactions to answer three research questions.

ACM Reference Format:
Anonymous Author(s). 2023. CEN5035 – Software Engineering – Fall 2023: Final Project Report. In *Proceedings of MSR '24: Proceedings of the 21st International Conference on Mining Software Repositories (MSR 2024).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 RESEARCH QUESTIONS

The different research questions we are exploring in this work are as follows,

(1) Do ChatGPT conversations tend to deviate?
(2) Which programming languages do developers ask ChatGPT for help with, and how does this pattern change over time?
(3) While interacting with ChatGPT, what emotions do developers generally display?

2 MOTIVATION

The motivation for researching the questions are explored in this section.

2.1 Do ChatGPT conversations tend to deviate?

Users can start a conversation with ChatGPT and the entire conversation can be about the same topic or they might move on to different topics as the conversation proceeds. Through this research question, we want to identify if conversations more often tend to deviate as it progresses. This information can be used to better equip AI tools to handle out of context questions.

2.2 Which programming languages do developers ask ChatGPT for help with, and how does this pattern change over time?

Developers working on different technologies and programming paradigms use ChatGPT for assistance. In this research question, we want to identify for which programming languages users take ChatGPT's assistance frequently and how this trend unfolds over time. By identifying this trend, future AI tools can be equipped with specialized knowledge about the most common programming languages to better assist its developer userbase.

MSR 2024, April 15–16, 2024, Lisbon, Portugal

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.3 While interacting with ChatGPT, what emotions do developers generally display?

Developers might display various emotions in the prompts given to ChatGPT. In this research question, we want to classify these emotions. Understanding the most common emotions and tone of developers will allow us to refine and develop AI tools that can better interpret and respond to users.

3 METHODOLOGY

In this section we explore the datasets used for the research questions, data pre-processing steps, and the approach taken to analyze the data.

3.1 Datasets

The DevGPT dataset types used for the research questions are listed un the below sections. All the questions use the following snapshots,

- `snapshot_20230727`
- `snapshot_20230803`
- `snapshot_20230810`
- `snapshot_20230817`
- `snapshot_20230824`
- `snapshot_20230831`
- `snapshot_20230907`
- `snapshot_20230914`

3.1.1 *Do ChatGPT conversations tend to deviate?*

Data types used are,

- `hn_sharings`: Hacker News thread.
- `pr_sharings`: GitHub Pull Requests.
- `discussion_sharings`: GitHub Discussions.
- `issue_sharings`: GitHub Issues.
- `commit_sharings`: GitHub Commits.

3.1.2 *Which programming languages do developers ask ChatGPT for help with, and how does this pattern change over time?*

Data types used are,

- `pr_sharings`: GitHub Pull Requests.
- `discussion_sharings`: GitHub Discussions.
- `issue_sharings`: GitHub Issues.
- `commit_sharings`: GitHub Commits.

3.1.3 *While interacting with ChatGPT, what emotions do developers generally display?*

Data types used are,

- `hn_sharings`: Hacker News thread.
- `pr_sharings`: GitHub Pull Requests.
- `discussion_sharings`: GitHub Discussions.
- `issue_sharings`: GitHub Issues.
- `commit_sharings`: GitHub Commits.

3.2 Data Pre-Processing

In data pre-processing, we describe the steps taken to extract relevant data from the raw DevGPT datasets and to clean the data to prepare it for further analysis.

3.2.1 ***Do ChatGPT conversations tend to deviate?*** The goal of pre-processing is to collect all the prompts and responses in the datasets.

- Removed conversations which are less than two pair of prompts and responses.
- Removed erroneous data instances Example - 404Status.

The processed data contains the prompts and responses and is stored in a file.

3.2.2 ***Which programming languages do developers ask ChatGPT for help with, and how does this pattern change over time?*** The raw datasets is parsed to collect the programming language and the ChatGPT conversation date. The steps taken to process the datasets are,

- Remove records which do not have programming language or date information.
- Convert ChatGPT conversation date from `"month_name date, year"` format to `"MM/DD/YYYY"`.
- Remove records with invalid date.

The cleaned data of a particular data type, from all the snapshots, are collated and written to a file for further processing.

3.2.3 ***While interacting with ChatGPT, what emotions do developers generally display?*** We can determine the developers' emotions by analyzing their prompts to ChatGPT. The objective of the pre-processing is to produce a file that contains only the developer prompts from the data sets and snapshots. The steps taken to lessen the noise in the data are,

- Remove prompts that have less than five characters.
- Remove Unicode data in the prompts.
- Remove HTML code snippets in the prompts.
- Remove extra white spaces in prompts.

The final processed file contains the prompt sentences in an array. Data from all the snapshots are merged into a single respective data type file.

3.3 Data Analysis

The pre-processed data is analyzed to answer the research questions.

3.3.1 ***Do ChatGPT conversations tend to deviate?*** To identify if a conversation has deviated, we use cosine similarity[3] to identify how similar the ChatGPT answers are within a conversation. Rather than analyzing user prompts, we focus on ChatGPT's answers for similarity computation. This approach mitigates the risk of misclassifying deviations that might occur due to generic prompts aligning with the previous context. By prioritizing ChatGPT's answers, we ensure a more accurate assessment since responses pertaining to the same topic exhibit similarity, even if prompted differently. For instance, while a prompt like "Tell more" maintains the conversation's context, similarity models might overlook this continuity, whereas ChatGPT's answers remain consistently akin when exploring the same theme. The average of the cosine similarity values computed by comparing consecutive responses is taken, and if this average falls below 0.5 (1 indicating most similar and 0 being most dissimilar), the conversation is classified as deviated. This information is plotted on pie charts and bar graphs.

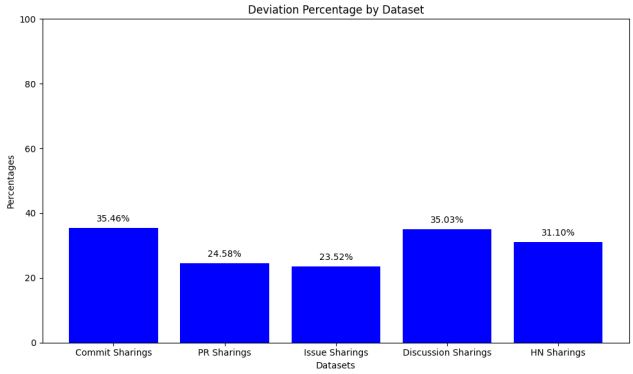


Figure 1: Research question 1 results

3.3.2 ***Which programming languages do developers ask ChatGPT for help with, and how does this pattern change over time?*** The data within each category is organized by programming language to gather insights on the top 5 languages. These insights are visualized via scatter plots, with the y-axis depicting the count and the x-axis representing time. While plotting the best-fit line to capture the trend accurately, a moving average[2] is utilized to smoothen the line, effectively normalizing sudden data fluctuations.

3.3.3 ***While interacting with ChatGPT, what emotions do developers generally display?*** Sentiment analysis is one type of natural language processing that helps identify the polarity of a document. We are attempting to classify the data along a spectrum of sentiments, such as joy, sadness, rage, fear, surprise, and love. A pre-trained sentiment categorization model based on Bert is used for the sentiment classification[1]. The user prompts are run through the sentiment analyzer, which gives a list of scores for all the emotions. The prompts are classified by the highest scoring emotions. The resulting classification are plotted on pie charts and bar graphs.

3.4 Results

3.4.1 ***Do ChatGPT conversations tend to deviate?*** The bar graph illustrating deviation percentages, plotted with the y-axis displaying deviations and the x-axis indicating various datasets, is depicted in Figure 1. Among these, commit sharings and discussion sharings exhibit the highest deviation rates, approximately reaching 35% of the conversations. Following closely, HN sharings demonstrate a deviation of around 31%, while PR sharings show approximately 25% deviation. Lastly, issue sharings display a deviation rate of roughly 24%.

3.4.2 ***Which programming languages do developers ask ChatGPT for help with, and how does this pattern change over time?*** The scatter plots representing the results for the various categories are shown in Figure 2. Within the PR sharings dataset (Figure 2a), the utilization of the top 5 programming languages—namely *TypeScript, JavaScript, Python, Go, Java*—has notably increased over time. In the commit sharings dataset (Figure 2b), the usage of the leading 5 programming languages—*CSS, HTML,*

Python, Jinja, JavaScript—has predominantly shown an upward trend or remained relatively constant. Within the issue sharings dataset (Figure 2c), the utilization of the top 5 programming languages—*Python, TypeScript, JavaScript, Java, C++*—has witnessed a decline over time. As for the discussion sharings dataset (Figure 2d), the trends among the top 5 programming languages—*Python, TypeScript, PHP, C, JavaScript*—are varied: most languages have either increased or remained stable over time, while the usage of the C programming language has experienced a decrease..

3.4.3 ***While interacting with ChatGPT, what emotions do developers generally display?*** The outcomes of the sentiment analysis are visualized in the bar graph displayed in Figure 3. The y-axis denotes the count, while the x-axis illustrates distinct datasets. Among the user emotions, joy emerges as the most prevalent across all datasets. Following joy, anger takes the position as the second most common emotion, trailed by sadness and fear. Love and surprise emotions occur infrequently. Figure 4 shows the pie distribution of the emotions in commit sharings dataset. Similar pie charts for the other databases can be found in the codebase.

3.5 Software and Tools

The software, libraries and tools that were used for the project are as follows,

- Python3
- pip
- GitHub
- VSCode
- Matplotlib
- Pandas
- numpy
- seaborn
- tensorflow
- transformers
- scikit-learn
- statistics

4 CONTRIBUTIONS

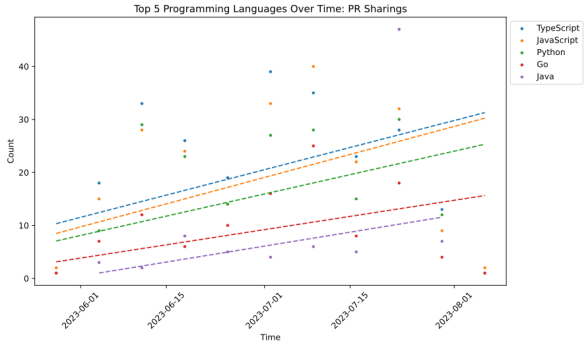
The contributions done by individuals are mentioned in this section.

4.1 Akhil Gorthi Bala Sai

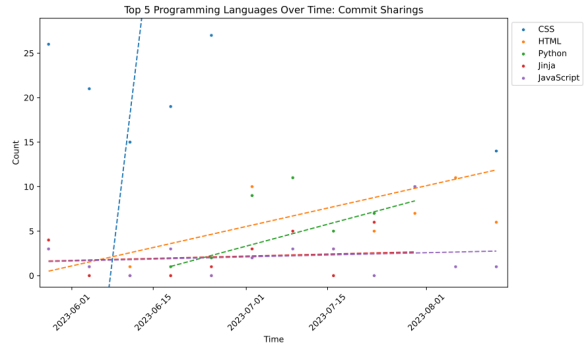
- Primarily worked on research question 1.
- For research question 1, pre-processed the datasets and removed the irrelevant data, combined the datasets from all the snapshots and categorized them.
- The files for research question 1 were pushed into GitHub accordingly.
- Worked on analyzing the data for research question 1, explored cosine similarity models, and different threshold values to identify deviation.
- Plotted pie chart and bar graphs for the results of research question 1.
- Contributed to writing all the reports and video updates.

4.2 Avaneeshakrishna Shastry Chakracodi

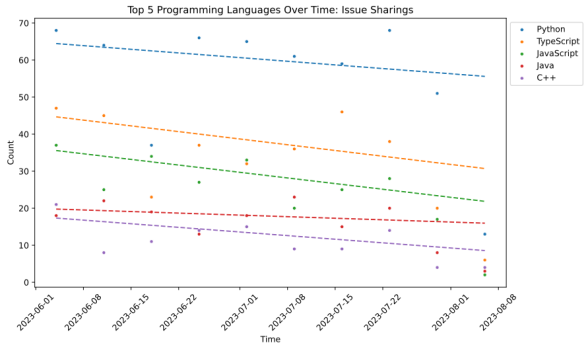
- Primarily worked on research question 3.
- Pre-processed all the datasets as mentioned for research question 3.
- Utilized Python to read, examine, and remove erroneous data from the JSON file, including HTML tags, unicodes, and



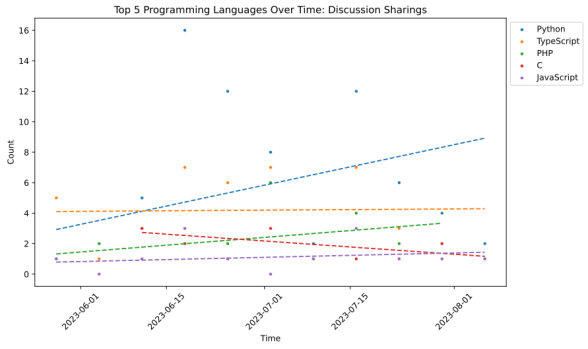
(a) PR Sharings



(b) Commit Sharings



(c) Issue Sharings



(d) Discussion Sharings

Figure 2: Research question 2 results

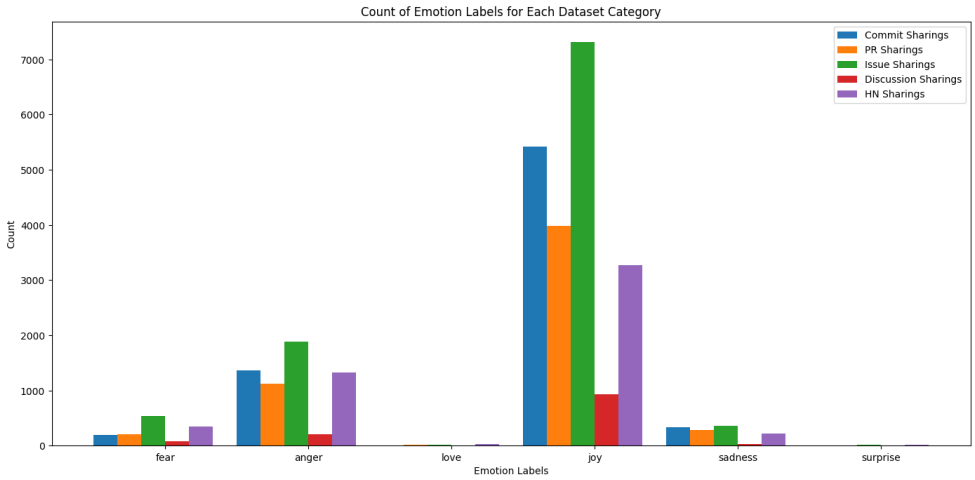


Figure 3: Research question 3 results

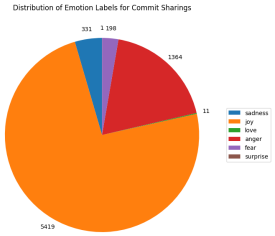


Figure 4: Emotions in commit sharing dataset

empty strings, to address research question 3. Additionally, limited the prompt to a minimum length of five characters.
- Categorized the processed datasets for research question 3 and stored on file.
- Used pre-trained Bert-based sentiment classification model on all the preprocessed datasets. The sentiment results are stored on file.
- Based on the highest value of all the emotions for a prompt, the emotion label in the graphs are counted.
- Pie charts and bar graphs were used as part of the sentiment analysis visualization to show the patterns in the overall number of emotions across all datasets.
- Pushed the Python code artifacts for research question 3 into GitHub.
- Contributed to writing all the reports and video updates.

4.3 Rohit Natesh

- Performed intial codebase setup & GitHub repository setup.
- Primarily worked on research question 2.
- Developed a dynamic data files reading technique for research question 2.
- Pre-processed the datasets for research question 2.
- Stored the pre-processed data on file to be able to run data pre-processing and analysis steps individually for research question 2.
- Performed data analysis on research question 2.
- Explored different plotting techniques for research question 2 and plotted the results using scatter plots with best fitting lines.
- Fixed an issue in research question 1 where the graphs were not being plotted correctly when done for all the datasets.
- Contributed to writing all the reports and video updates.

5 PROJECT ARTIFACTS

The project's repository can be accessed on GitHub. Link to repository is https://github.com/rohitnatesh/software-engineering-project.

6 CONCLUSION

In conclusion, these research questions shed light on some of the trends of developer conversations with ChatGPT. These findings can aid in improving AI tools to better handle developer conversations.

REFERENCES

[1] 2023. Sentiment Analysis BERT Model - bhadresh-savani/bert-base-uncased-emotion. https://huggingface.co/bhadresh-savani/bert-base-uncased-emotion
[2] 2023. Simple Moving Averages Make Trends Stand Out. https://www.investopedia.com/articles/technical/052201.asp
[3] 2023. Understanding Cosine Similarity in Python with Scikit-Learn. https://memgraph.com/blog/cosine-similarity-python-scikit-learn
[4] Tao Xiao, Christoph Treude, Hideaki Hata, and Kenichi Matsumoto. 2024. DevGPT: Studying Developer-ChatGPT Conversations. In *Proceedings of the International Conference on Mining Software Repositories (MSR 2024).*