

Data Wrangling Report

Analysis of Developer Interactions with ChatGPT Using the DevGPT Dataset
4th Feb 2025

Introduction

This project examined the DevGPT dataset in order to understand how developers interact with ChatGPT. This data can help us understand what types of issues are most commonly faced by developers, and which of these issues developers choose to try to solve using ChatGPT instead of through other means. This is a publicly available dataset, found here:

<https://github.com/NAIST-SE/DevGPT>. It contains files with records of developers' interactions with ChatGPT in each of the following categories: Hacker News, conversations, mention, GitHub pull request, GitHub issue, GitHub commit, and GitHub code file.

Our study focuses on three key research questions:

1. What is the typical structure of conversations between developers and ChatGPT? How many turns does it take on average to reach a conclusion? Does this depend on the type of request?
2. What types of issues (bugs, feature requests, theoretical questions, etc.) do developers most commonly present to ChatGPT?
3. What is the most commonly used programming language in these interactions?

Methodology

Research Question 1: Conversation Structure Analysis

In order to approach this question, we decided to use the data from an issue and conversation file. First, the data was imported and normalized using the json Python package. We then investigated the characteristics of the data in terms of the tokens of prompts and number of prompts. By comparing these qualities of the files, we could understand how many turns developers usually take in conversation with ChatGPT, and how this differs between requests related to discussions and issues. The number of prompts can represent the number of turns a developer takes in the conversation, as each prompt represents one of their turns. The

discussion file had a mean of 5.27 prompts per conversation, while the issues file had a mean of 4.57 prompts per conversation.

We used matplotlib and seaborn to visualize the distributions of the number of prompts per conversation in the files. Both distributions were negatively skewed, with most conversations having less than 10 prompts, but some outliers reaching over 80 prompts.

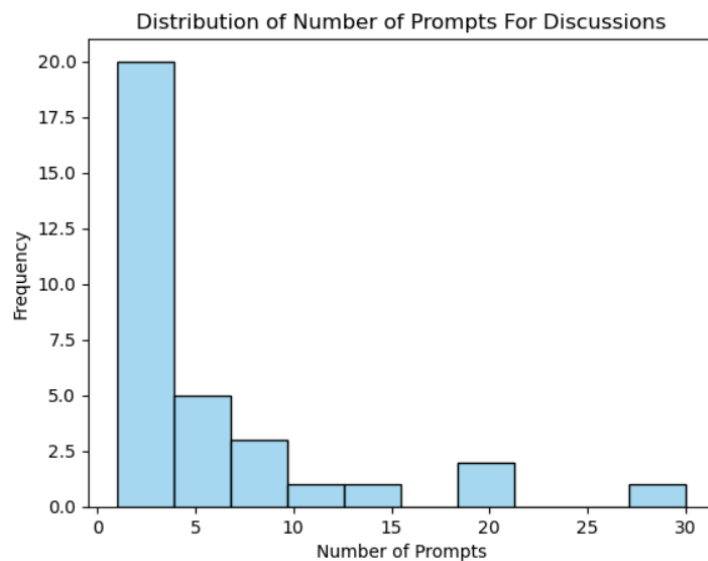


Figure 1: Distribution of the Number of Prompts for Discussions

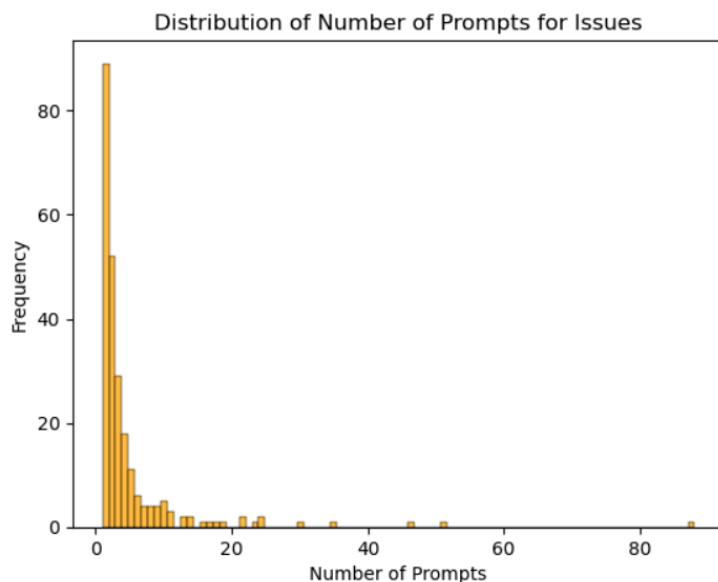


Figure 2: Distribution of the Number of Prompts for Issues

In order to determine if the difference in distributions was statistically significant, we decided to use a Mann-Whitney U test. This is because there were some extreme outliers, so comparing the means using a standard t-test may not have been as fruitful. After running the test as part of the SCIPY python package, the U-statistic was 4360, and the p-value was 0.40, indicating there is no statistically significant difference in the medians of the distributions. This suggests that developers' conversations with ChatGPT do not take a different number of turns when the conversation is related to issues compared to discussions.

Although each type of conversation did not differ in its number of prompts, perhaps the prompts themselves would differ in length. So, to further investigate the structure of the conversations, we compared the number of tokens per prompt in each of these files. Tokens represent parts of the prompts, such as words, or punctuation. The mean number of tokens for the issues file was 173, and the mean number of prompts for the discussions file was 215. However, again they were both heavily skewed to the left and contained some extreme outliers. To examine the difference in these distributions, we ran another Mann-Whitney U test. The U-statistic was 4160.5, and the p-value was 0.73, suggesting there is no relationship between the type of conversation and number of tokens per prompt.

Research Question 2: Identifying Common Developer Issues

To produce a larger dataset, we first combined all of the issue files from different dates. Then the data wrangling was performed to get a cleaned version of data then the Natural Language Processing (NLP) techniques were applied to find the categories.

Our categorization process is as follows. First, we defined the main issue categories, which included bugs, feature requests, theoretical questions, and optimization. Some keywords were separately given to each category. However, after investigating the proportion of prompts that included these keywords, we found that the vast majority of the prompts were not being detected. As a result, category-related keywords were expanded by using Masked Language Modeling (MLM), Context-Based Expansion, and Hugging Face Transformers. Then, these expanded keywords were applied to detect these kinds of issues within the cleaned version of the dataset. Finally, we visualized the proportions of each issue type to understand their distribution.

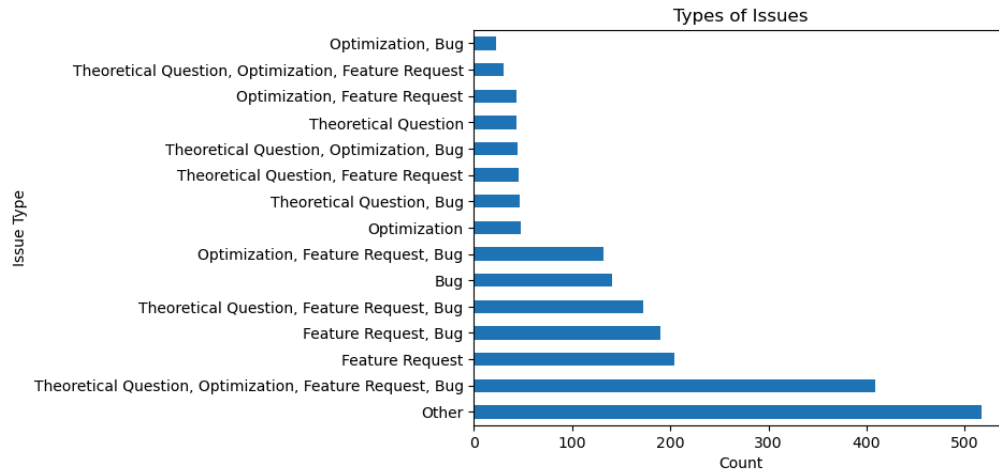


Figure 3: Types of issues.

Of the identified issue categories, those pertaining to a theoretical question, optimization, feature request and bug were most common. This was then followed by those containing a feature request, then those mentioning a feature request and a bug. The least common issue type we identified was issues related to optimization and a bug.

Research Question 3: Most Common Programming Languages Used

To determine the most frequently used programming languages in these datasets, we used simple keywords to detect the language. We chose to examine this question in the context of the “issues” files to understand which languages are most common when developers are discussing issues with ChatGPT. It is shown that the most commonly used languages were Python and JavaScript, with HTML being the least commonly used language.

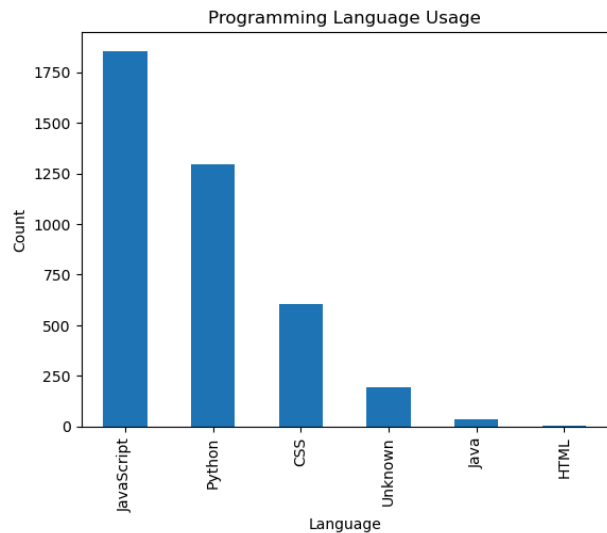


Figure 4(Right): Top using Programming Languages.

Results and Conclusion

- Conversation Structure: Conversations typically involve 4.57–5.27 prompts, with no significant difference between issues and discussions.
- Types of Issues: Developers most commonly seek ChatGPT's help for bug fixes, feature requests, theoretical discussions, optimizations and some combined questions.
- Top Programming Languages Used: Python and JavaScript

Based on these results, it is suggested that developers rely on ChatGPT for both conceptual and technical problem-solving, with Python and JavaScript being the most popular language for inquiries related to issues.

This project provides insights into how developers interact with ChatGPT, revealing patterns in conversation structure, issue types, and programming language usage. Our methodology, combining statistical analysis, NLP techniques, and data visualization, enabled a structured investigation of developer behavior in AI-assisted coding. In future work, we could explore how ChatGPT's problem solving efficiency changes across coding languages, or compare these developer interactions with other AI-assisted coding tools such as DeepSeek.

Reuse Statement

We used ChatGPT to assist in some code while making sure not violating any copyright policy. All code reused is subject to permissive license terms.