

SOFTWARE ENGINEERING PROJECT PHASE 2

MSR 2024 Mining Challenge

AkhilRaj Tirumalasetty
Florida State University
at21bo@fsu.edu

Manikanta Mamidi
Florida State University
mm23bm@fsu.edu

Sruthijha Pagolu
Florida State University
sp23bu@fsu.edu

ABSTRACT

This report presents a research proposal aimed at exploring and improving ChatGPT's conversational capabilities. The study is motivated by the growing importance of AI chatbots like ChatGPT in various technical domains, especially programming. The research revolves around three key questions: predicting the length of a conversation with ChatGPT based on the initial prompt and context, assessing ChatGPT's understanding and response accuracy in different programming languages, and identifying the critical points in conversations involving coding languages where ChatGPT's performance diminishes.

Employing a methodical approach that encompasses the analysis of conversation JSON files, sentiment analysis, and the application of advanced neural networks and regression models, the study seeks to quantitatively and qualitatively evaluate ChatGPT's interactions. The project's scope includes data collection and cleaning, feature engineering, model training and evaluation, culminating in comprehensive documentation and presentation of findings.

The anticipated outcome of this research is to offer insights into the dynamics of ChatGPT's conversations in programming contexts, thereby contributing to its improvement and reliability as a tool for software developers. The findings are expected to pave the way for more intuitive, efficient, and safer AI-assisted programming environments.

1. INTRODUCTION

In the era of rapid technological advancement, AI-driven chatbots like ChatGPT have emerged as pivotal tools in various sectors, including software development. The ability of these chatbots to understand and interact in human language has opened new frontiers in enhancing productivity and solving complex problems. However, the

effectiveness of these AI systems, particularly in technical domains such as programming, is a subject that warrants thorough exploration. This research project is designed to delve into the intricacies of ChatGPT's interactions in software development contexts.

The primary objective of our study is to examine and improve the interaction dynamics of ChatGPT, especially when dealing with programming-related queries and conversations. To achieve this, we have formulated three research questions focusing on predicting conversation length with ChatGPT based on initial prompts, evaluating the chatbot's understanding and response capabilities in various programming languages, and identifying the critical junctures where ChatGPT's performance in coding conversations tends to falter.

Our research methodology combines a mix of qualitative and quantitative analyses, including the examination of ChatGPT conversation JSON files, sentiment analysis, and the implementation of machine learning techniques like neural networks and regression models. This approach allows us to model and understand the various facets of ChatGPT's conversational behaviors in a programming environment.

By addressing these research questions, we aim not only to enhance ChatGPT's functionality and reliability for developers but also to contribute to the broader field of AI and machine learning, particularly in the context of natural language processing and human-AI interaction. This study is poised to offer valuable insights into making AI chatbots more intuitive and effective, thereby fostering a more efficient and safe programming landscape.

2. RESEARCH QUESTIONS

the research aims to assess the capability of ChatGPT to comprehend and provide effective responses to a range of programming languages. By testing ChatGPT's ability to understand programming languages, it can become a

valuable tool for programmers, students, and anyone who needs help with coding. The study can also provide valuable insights for developers, students, and professionals working in the field of software development. Ensuring ChatGPT understands programming languages correctly is essential for safety, as it can help prevent incorrect advice or issues when people are working on important projects. Additionally, the study can help determine the threshold point where the conversation will fail beyond that point, making ChatGPT a more valuable tool for programmers, students, and anyone who needs help with coding. The study also aims to find out the average stage or point of failure of ChatGPT in a particular coding language. This can help developers pinpoint weaknesses in the model, leading to targeted improvements in future versions. Understanding the limitations of ChatGPT can help set realistic user expectations, and for educators using ChatGPT as a supplementary tool for coding / programming courses, it can help structure lessons or student interactions with the model more effectively. The research can contribute to making ChatGPT a more valuable and reliable tool for various users, while also ensuring that technology is both helpful and safe in our increasingly digital world.

We attempt to address the following research questions:

- 1) How accurately can we predict the length of a conversation with ChatGPT based on the initial prompt and context provided?
- 2) To what extent can ChatGPT correctly respond to various programming languages?
- 3) At what stage of the conversation on average does ChatGPT fail in particular coding language?

Up to the completion of phase 2 of the project, our focus has been on addressing research question 2. Moving forward, our next objective is to tackle research question 1, which is intrinsically linked to research question 3.

3. METHODOLOGY

This section involves the steps followed in order to address the research question 2.

3.1 Data cleansing

We obtained a dataset from the specified GitHub repository for our research. Upon examining the dataset, it became apparent that it contained a significant amount of extraneous information not pertinent to our study. Our focus

for this paper is solely on the conversational aspects within the dataset. To streamline the data and eliminate irrelevant content, we employed PowerShell technology. By executing a custom script, we effectively reduced the dataset size from 200MB to a more manageable 40MB. To further facilitate our analysis, we segmented the 40MB dataset into smaller, more manageable portions, each approximately 5MB in size.

3.2 Data integration

The initial step is to read the contents of a text file, which comprises a series of messages exchanged between the user and ChatGPT. The text extracted from this file is systematically structured and transformed into JSON (JavaScript Object Notation) format. After converting the data into JSON, it is placed into an object, enabling the extraction of the conversation between the user and ChatGPT. Subsequently, this extracted conversation is used to create a dataset. With the dataset in place, the JSON object allows for the necessary access and manipulation of the data as needed.

3.3 Core Algorithm

We proceed to perform sentiment analysis on the extracted data. This process is aimed at understanding user satisfaction or dissatisfaction within the conversation. However, analyzing each prompt in the conversation for sentiment is complex, so we adopt a strategy where we consider the last 10% of the conversation as representative of its overall tone. This approach is based on the assumption that the concluding part of the interaction most likely reflects the user's overall sentiment.

In this sentiment analysis, each message or object is evaluated and assigned a sentiment score that ranges from -1 to +1. This score is based on the likelihood of the sentiment expressed in the message, where -1 indicates a high level of dissatisfaction, and +1 indicates satisfaction. This method allows us to gauge the user's sentiment in a more manageable and focused manner, concentrating on the end part of the conversation to infer the general tone of the entire interaction.

RESULTS

Utilizing sentiment analysis, we have derived the accompanying graph. The data indicates that a significant number of users engaged with ChatGPT primarily on topics concerning the Python programming language. It is noted that the bulk of these users expressed satisfaction with the

responses provided by ChatGPT. In a comparable manner, this analysis can be extended to conversations involving other programming languages to assess the degree to which the chatbot delivered satisfactory outcomes.

