Software Engineering Final Project Report

Abstract— **This report presents the findings of a comprehensive study conducted on the usage of large language models like ChatGPT by software developers. We focused on three primary research questions addressing the types of issues developers present to ChatGPT, the common types of prompts they use, and the predominant use cases for seeking assistance from these models. Utilizing the DevGPT dataset (snapshot_20231012), which includes conversations between developers and ChatGPT, we employed text processing and machine learning techniques, including SVM and Logistic Regression models, to analyze and categorize the data, followed by performance evaluation. The results indicate that "Bug" categories are the most frequent issues, contextual prompts are predominantly used, and "Content Creation" emerges as the primary use case for developers. These insights contribute to a deeper understanding of how software developers interact with and utilize AI language models in their workflows.**

I. Introduction

The advent of language models like ChatGPT has revolutionized the way developers approach problem-solving, code generation, and even bug fixing. As these models become more sophisticated, understanding their practical applications and limitations in software development is crucial. Our study is motivated by the need to empirically analyze and understand the types of interactions that developers have with ChatGPT. This involves examining the nature of queries posed by developers, the context in which they seek AI assistance, and the overall impact of such interactions on their workflow. Our research is grounded on three primary objectives: first, to categorize the types of issues developers present to ChatGPT; second, to identify the common types of prompts used in these interactions; and third, to ascertain the predominant use cases for AI assistance in software development. By analyzing the DevGPT dataset, a collection of real-world interactions between developers and ChatGPT, we aim to provide valuable insights into how AI is shaping the future of software development.

II. RESEARCH QUESTION 1

What types of issues (bugs, feature requests, theoretical questions, etc.) do developers most commonly present to ChatGPT?

Bugs: Developers ask for assistance in locating and resolving coding errors that result in unexpected behaviour or application malfunctions.

Feature Requests: To address changing user demands and market trends, developers seek advice on adding new features or enhancing current ones.

Theoretical Questions: In order to make wise decisions for projects, questions centre on understanding programming concepts, algorithms, design patterns, and software architecture.

Security: In order to safeguard their applications from cyber threats, developers consult with experts on how to put strong security measures in place, fix vulnerabilities, and follow best practices.

A. METHODOLOGY

Our research methodology is structured to systematically analyze the DevGPT dataset, a comprehensive collection of interactions between software developers and the ChatGPT language model. The methodology encompasses several key stages. The stages are as follows:

- **Data Preparation -** We utilized various JSON files containing user prompts. Each file represented a different context, such as discussions, issues, and commits.
- **Preprocessing -** The prompts were preprocessed using natural language processing techniques. This included:
  - Removing punctuation and converting text to lowercase.
  - Eliminating English stopwords using NLTK's stopwords corpus.
  - Applying lemmatization to the words to obtain their base form.
- **Keyword-Based Categorization -** We established a dictionary of keywords for each category (Bug, Feature Request, Theoretical Question, and Security). Prompts were initially categorized based on the presence of these keywords. We have explored various resources relevant to our keyword categories and the resources are as follows:

  1) Bug :
  - GitHub: Repository hosting bug reports, discussions, and issue tracking.
  - Stack Overflow: Q&A platform for programming-related queries and bug discussions.

  2) Feature Request:
  - Product Forums (e.g., Adobe, Microsoft): Discussion forums within specific software product sites.
  - Developer Surveys (e.g., Stack Overflow Developer Survey): Surveys collecting insights into feature demands from developers.

  3) Theoretical Question:
  - ResearchGate: Academic platform hosting research papers and scholarly articles.
  - EEE Xplore: Digital library for research papers and publications in engineering and technology.

4) Security Category:

- CERT (Computer Emergency Response Teams):* Provides reports on cybersecurity incidents and vulnerabilities.
- Reddit NetSec: Subreddit dedicated to network security discussions and news.

- **TF-IDF Vectorization -** The preprocessed texts were converted into a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) features. This transformation turns the textual data into a format suitable for machine learning models.
- **Label Encoding -** We encoded the categorical labels into numerical form using Scikit-learn's LabelEncoder.
- **Model Training -** We employed a Support Vector Machine (SVM) with a linear kernel, a popular choice for text classification tasks, to train our model. The dataset was split into training and testing sets, with 80% of the data used for training and 20% for evaluation.
- **Evaluation and Visualization -** The trained model was evaluated on the test set. We also visualized the distribution of prompts across different categories using bar charts.
- **Prediction -** Finally, we implemented a function to predict the category of new, unseen prompts using the trained SVM model and TF-IDF vectorization.

B. RESULT

The analysis of the DevGPT dataset revealed that the most common category of issues presented to ChatGPT by developers was "Bug", followed by "Feature Request" and "Theoretical Question". The category with the fewest counts was "Security". The prediction is also shown in the figure below. The sample prompts results are available at https://github.com/siddharthdhola/Group_5_Software-Engineering_Project/tree/main/Results/Question1.
Coming to the accuracies : SVM model gave accuracy of 95.634%. The accuracies per category are given in the table below.

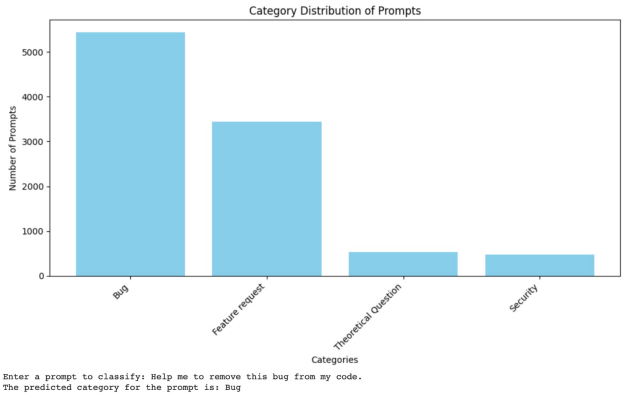

Fig. 1. Distribution of issue types

The example prediction for the prompt is also included in the figure.

Now, coming to the accuracies : SVM model gave accuracy of 89.20%. The accuracies per category are given in the table.

Prompts	Correct	Total	Accuracy
Bugs	17	20	85%
Feature Requests	18	20	90%
Theoretical Questions	17	20	85%
Security	19	20	95%

TABLE I
ACCURACY TABLE PER CATEGORY WITH CORRECT AND TOTAL PREDICTIONS MADE

III. RESEARCH QUESTION 2

What are the most common types of prompts (Single turn, Multi turn and Contextual prompts) that developers use when interacting with ChatGPT?

Single-turn prompts: These are short, direct inquiries seeking immediate responses from ChatGPT without continuation, ideal for quick, specific information or isolated problem-solving.
Multi-turn prompts: Involving a series of linked questions and responses, these prompts enable ongoing conversations with ChatGPT, allowing for deeper exploration and step-by-step issue resolution.
Contextual prompts: Used to maintain conversation flow, these prompts refer back to previous interactions with ChatGPT, ensuring continuity and leveraging prior information for a seamless, connected discussion.

A. METHODOLOGY

The Methodology is as follows:

- **Data Preprocessing**
  - **Standardization:** This stage involves cleaning, normalizing, and standardizing the text data from the DevGPT dataset. Utilize tools from NLTK and spaCy for tokenization, lemmatization, and removal of stopwords.
- **Prompt Type Identification:**
  - **Prompt Categorization:** Classify prompts into types such as Single turn, Multi turn and Contextual prompts. This categorization is based on the analysis of sentence structure, technical terminology, and contextual framing.
- **Quantitative and Comparative Analysis:** Conversation id's are used for this step as they are the most important aspect.
  - **Quantitative Analysis:** Perform a statistical analysis to determine the frequency and distribution of different prompt types within the dataset, identifying trends and correlations.
  - **Comparative Analysis:** Compare the responses generated by ChatGPT for each prompt category, analyzing their accuracy, relevance, and completeness. Assess whether certain types of

prompts yield more effective responses from the AI model.

- **Feature Extraction and Vectorization:**
  - **Vectorization Technique:** Use CountVectorizer from Scikit-learn to convert the categorized text data into numerical feature vectors, focusing on word frequency.
- **Machine Learning Model Training and Evaluation:**
  - **Model Selection:** Trained Logistic Regression model, as this is effective for text classification tasks.
  - **Model Training and Testing:** Divided the dataset into training set (80%) and testing set (20%), train the models on the training data, and evaluate their performance on the test data.
  - **Performance Metrics:** Assessed the models using accuracy scores and classification reports, and conduct an error analysis to understand misclassifications.
- **Visualization and Communication of Findings:**
  - **Data Visualization:** Utilize visualization tools like matplotlib to graphically represent the distribution of prompt types and the results of the comparative analysis, aiding in the communication of findings.

B. RESULT

The analysis of interaction patterns with ChatGPT revealed that the most frequent type of prompt used by developers was the Contextual prompt. This was followed by Multi-turn prompts, with Single-turn prompts being the least common. The prediction is also shown in the figure below. The sample prompts results are available at https://github.com/siddharthdhola/Group_5_Software-Engineering_Project/tree/main/Results. Coming to the accuracies : SVM model gave accuracy of 95.634%. The accuracies per category are given in the table below.

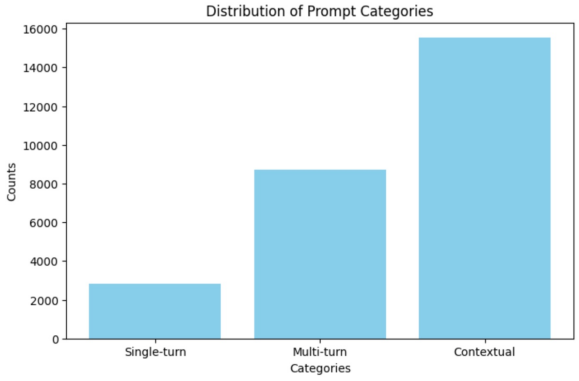

Fig. 2. Distribution of prompts types

Prompts	Correct	Total	Accuracy
Single-Turn	19	20	95%
Multi-Turn	16	20	80%
Contextual	18	20	90%

TABLE II
ACCURACY TABLE PER CATEGORY WITH CORRECT AND TOTAL PREDICTIONS MADE

Now, coming to the accuracies : Logistic Regression model gave accuracy of 88.63%. The accuracies per category are given in the table.

IV. RESEARCH QUESTION 3

What are the most common use cases (Content Generation, Information Retrieval, Natural Language Understanding and Language Translation) for which developers turn to ChatGPT for Assistance?

Content Generation: Developers often seek ChatGPT's help in generating various types of content, such as articles, stories, code snippets, or creative writing prompts.
Information Retrieval: Developers use ChatGPT to retrieve specific information or facts on diverse topics, seeking concise, accurate responses akin to a search engine.
Natural Language Understanding: ChatGPT assists developers in understanding and interpreting complex natural language queries, providing contextualized responses or aiding in language comprehension tasks.
Language Translation: Developers utilize ChatGPT for language translation tasks, requesting translations between languages for text or phrases.

A. METHODOLOGY

- Data Preprocessing and Environment Setup
  - **Libraries and Resources:** Utilized NLTK for natural language processing and Scikit-learn for machine learning. Download necessary NLTK resources like stopwords and wordnet.
  - **Data Cleaning and Normalization:** Applied regex and NLTK tools for tokenization, lemmatization, and removal of stopwords to clean and standardize the text data from the DevGPT dataset.
- Categorization
  - **Use Case Categorization:** Classify interactions based on application contexts, such as Content Generation, Information Retrieval, Natural Language Understanding and Language Translation , using a combination of keyword analysis and context interpretation. We have explored various resources relevant to our keyword categories and the resources are as follows:

1) Content Creation:

· Towards Data Science, OpenAI Blog: Analyzed blogs discussing AI content generation for relevant terms.

· Articoolo, Writesonic: Extracted keywords from websites offering content generation tools.

2) Information Retrieval:

· Google SERPs: Scraped top search results for indexing, search algorithms, query optimization terms.

· IEEE Xplore, ACM DL: Structured queries to gather terms on information retrieval from academic databases.

3) Natural Language Understanding:

· Hugging Face, Google BERT: Extracted keywords from NLP models for text and sentiment analysis, entity recognition.

· Common Crawl, Wikipedia dumps: Analyzed datasets for terms on language understanding, POS tagging.

4) Language Translation:

· Europarl, United Nations documents: Analyzed bilingual corpora for translation-related phrases.

– Feature Extraction and Vectorization
  ∗ **TF-IDF Vectorization:** Employed TfidfVectorizer from Scikit-learn to transform the processed text data into TF-IDF feature vectors, crucial for converting textual data into a numerical format for machine learning.
– Machine Learning Model Training and Evaluation
  ∗ **Data Splitting:** Divided the dataset into training and testing sets using train_split_test.
  ∗ **Model Training:** Trained a Support Vector Classifier (SVC) on the training data, chosen for its effectiveness with high-dimensional data.
  ∗ **Model Evaluation:** Used accuracy score, classification report, and other relevant metrics to assess the model's performance. Conducted an error analysis to identify misclassification patterns.
– Quantitative Analysis and Cosine Similarity
  ∗ **Quantitative Analysis:** Analyzed the frequency and distribution of different use cases within the dataset to identify trends and common patterns.
  ∗ **Cosine Similarity Analysis:** Implemented cosine similarity measures to explore the relationships and similarities between different use cases in the feature space.
– Data Visualization
  ∗ **Visualization Techniques:** Utilized matplotlib to create visual representations, such as the distribution of use cases or results of cosine similarity analysis, aiding in better understanding and communication of the findings.

B. RESULT

The analysis shows that "Content Creation" was the most common use case for ChatGPT among developers, followed by "Language Translation" and "Information Retrieval", with "Natural Language Understanding" being the least used. The prediction is also shown in the figure below.

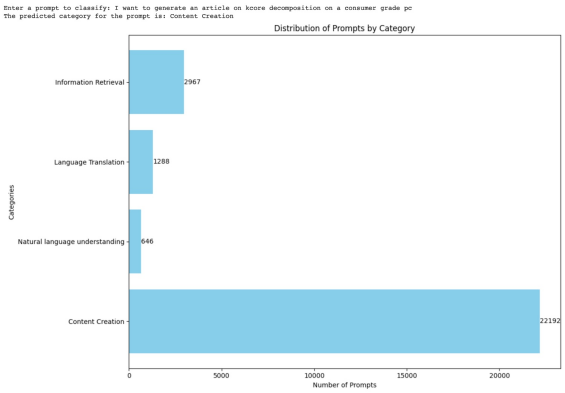

Fig. 3. Distribution of common usecases types

The sample prompts results are available at https://github.com/siddharthdhola/Group_5_Software-Engineering_Project/tree/main/Results. Coming to the accuracies : SVM model gave accuracy of 95.634%. The accuracies per category are given in the table below.

Prompts	Correct	Total	Accuracy
Information Retrival	19	20	95%
Natural Language Understanding	18	20	90%
Language Translation	19	20	95%
Content Creation	19	20	95%

TABLE III
ACCURACY TABLE PER CATEGORY WITH CORRECT AND TOTAL PREDICTIONS MADE

CONCLUSIONS

Our research shows how developers often use AI tools like ChatGPT to solve problems, especially for fixing bugs in software. We found that they prefer to ask detailed questions that need understanding of the background, rather than just simple, one-off questions. The main reason they use AI is to help create content and translate languages, but they don't use it much for understanding complex language. Our study also shows that using machine learning (like SVM and Logistic Regression models) to sort these questions works really well. This tells us that AI can be a big help in making software and can get even better in the future.