Engineering Acceleration Projects Proposal

Author: Ben Zhang
Date: Aug 21, 2023

Executive Summary

As an engineering manager with strong interest and relevant experience building LLM AI applications and platform toolings, I have identified several key projects to accelerate engineering processes at companies like OpenAI using LLM AI. These projects aim to enhance productivity, efficiency, and collaboration within the engineering teams. This proposal outlines three distinct projects that address three different areas of improvement. Each project includes objectives, deliverables, and estimated timelines, providing a comprehensive roadmap for successful implementation.

Projects

Automated Code Review

Objective

Implement an LLM AI-powered code review system to automate and streamline the code review process, improving code quality, reducing manual effort, and increasing development speed. While this project may not replace human code reviews entirely, it could help code owners to discover issues early and ship code sooner.

Deliverables

1. Develop a robust training dataset comprising high-quality code samples and associated code review feedback.
2. Fine-tune LLM AI model to recognize code patterns, identify common code issues, and provide actionable suggestions for improvement.
3. Integrate the LLM AI model into our code review workflow, allowing developers to receive automated code review feedback.
4. Evaluate the effectiveness of the automated code review system through metrics such as code quality improvements, reduced code review time, and developer feedback.

Timeline

1. Dataset preparation and model fine-tuning: Month 1-2
2. Integration and testing: Month 3-4

3. Evaluation and refinement: Month 5-6

Semantic Search on Code

Objective

Implement a semantic search system using LLM AI to enable developers to find relevant pull requests in the past and code snippets quickly and accurately, improving code reuse, knowledge sharing, and development efficiency. This project helps reduce duplicate efforts and reduce time spent to research things that were already researched by other people.

Deliverables

1. Develop a comprehensive dataset of code snippets, libraries, and relevant resources with associated metadata.
2. Devise a comprehensive solution to create effective embeddings for code snippets, pull request and code-related metadata for the most accurate retrieval of related code and pull requests based on user queries.
3. Build a user-friendly interface that allows developers to input search queries and receive accurate and contextual code search results.
4. Integrate the semantic search system into our development environment and tools, enabling seamless access to code snippets and resources.
5. Evaluate the performance of the semantic search system through metrics such as search accuracy, retrieval speed, and user satisfaction.

Timeline

1. Dataset preparation and metadata collection: Month 1-2
2. Search results optimization: Month 3
3. Interface development and integration: Month 4
4. Evaluation and refinement: Month 5

Developer Chatbot

Objective

Develop an AI-powered chatbot that provides developers with instant access to best practices, technical documentation, and development guidelines, improving knowledge sharing, onboarding efficiency, and developer productivity.

Deliverables

1. Design and build a conversational AI chatbot capable of understanding natural language queries related to software development.
2. Integrate the chatbot with relevant knowledge bases, technical documentation, and existing resources within the organization.
3. Develop the prompt for the chatbot to provide accurate and helpful responses to developer queries, leveraging LLM AI for context-aware recommendations.
4. Implement a user-friendly interface that allows developers to interact with the chatbot seamlessly and to gather feedback about the quality of chatbot answers.
5. Evaluate the chatbot's performance through measures such as response accuracy, user satisfaction, and the number of queries resolved.

Timeline

1. Chatbot design and architecture: Month 1-2
2. Knowledge base data ingestion and bulk ingestion tooling: Month 2-3
3. Interface development and integration: Month 3-4
4. Evaluation and refinement: Month 4-5

Conclusion

These three projects aim to leverage LLM AI to accelerate and optimize engineering processes at OpenAI. By automating code review, enabling semantic search on code, and providing developers with instant access to best practices and development guidelines, we can enhance productivity, reduce manual effort, and foster collaboration among our engineering teams. The proposed timelines outline an estimated schedule for completing each project given 2 engineers for each project, but they are subject to adjustments based on resource availability and project dependencies. Successful implementation of these projects will significantly augment our engineering capabilities and drive OpenAI's mission forward.