

Project Proposal: Swift GPT Chatbot

Project Description

Problem to be Solved

- Current chatbot has to call the API everytime a student asks a question even if it is a repeatedly asked question.
- Create a scoring method to identify the similarities between question asked and question stored.

App Functionality

Although the GPT API has already been effectively integrated, we aim to enhance its capabilities through the incorporation of a caching system and integration with Piazza. This new feature will enable the chatbot to save previously posed questions in a cache, enabling it to retrieve these stored responses when the same questions are asked again, thus avoiding the need to generate fresh responses.

Features to work on:

- Establishment of caching systems to store user's question and AI response.
- Ability to recognize if a question has previously been asked.
- Ability to fine-tune the GPT-3 model to better understand and answer questions in the ECE2026 domain.
- Implementation of methods to verify the accuracy of the answers provided by the GPT-3 API.
- Security and privacy considerations for handling sensitive student data, such as their questions and answers.

Benefits Of This App

- Convenience: Quick and easy for students to use since the majority of them are on their phone already.
- Utilizes the powerful GPT-3 to help make tough concepts very simple. Since professors and lecturers may not always be available to make students understand the concepts.
- Cut down API call expenses by storing and retrieving past questions in a cache instead of generating new responses.

Potential Problems and Pitfalls/Other Areas for Research

1. Training GPT-3 with the material specific to the textbook provided. Although GPT-3 has a considerable amount of knowledge even in signal processing, it needs to narrow down to the textbook information. However, the textbook has a lot of words and it may be difficult to train the model with so many words at once.
2. GPT-3 is also not a free API. Every request has a small fee (approx 0.002\$ per 1000 tokens). Hence we may shift to a new model, if the model is free, and gives a similar amount of accuracy as GPT-3.
3. Matching a previously asked question might not be an easy task as it is possible for the same question to be asked with different arrangements. This is the area of NLP that is outside of our expertise as of now.

Implementation

Software and Development Tools

- File management and version control
 - Git/GitHub
- Programming languages
 - Mostly Swift
- Database management
 - Cloud-based MongoDB
- API for communication
 - NodeJS

Project Goal For This Semester

We want to create a convenient chatbot that can answer student questions based on ChatGPT's answering mechanism. By creating it in Swift, we could merge it into the main QuizApp to reach a wider number of students.

Anticipated Milestones	Date Due
Team Accustomization	Week 3
Swift Research	Week 4
ChatGPT Research	Week 5
First Line of Code	Week 6
Working on prototype	Week 7
Working on prototype	Week 8
Prototype	Week 9
Working on Main Project	Week 10
Working on Main Project	Week 11
Main Product	Week 12
Refactor and Working on integration	Week 13
QuizApp Integration	Week 14
Presentation	Week 15

Group Membership

Task Assignment and Participation

Responsibilities

- Alvin Fabrio
 - NodeJS Backend, IOS frontend, API integration, etc.
- Meenakshi Prabhakar
 - Backend, potentially frontend
- Nico Jaimes
 - NodeJS Backend, Piazza/OpenAI API Integration
- Duy Nguyen
 - API integration
- Wei Tan
 - NodeJS, iOS

Member Skill Sets

Member	Time Commitment and Credits	Skills and Interests
Alvin Fabrio	6-8 hours/week (2 credits)	Java, JavaScript, React, Python, HTML/CSS, Android Studio, Swift
Wei Tan	6-8 hours/week (2 credit)	Java, JavaScript, React, Python
Duy Nguyen	2-4 hours/week (1 credit)	Java, Python, React

Communication Resources

- Weekly meetings: Tentative
- Google Drive: meeting notes, learning resources, and other shared documents
- Microsoft Teams for quick communications and meetings
- Trello Board: to keep track of tasks