Course Recommendation Chatbot

Stanford CS224N Custom Project

Naama Bejerano
Department of Computer Science
Stanford University
n20bej@stanford.edu

Emma Troast
Department of Computer Science
Stanford University
ebtroast@stanford.edu

Abstract

In this project, we aim to analyze and compare two leading LLM chat APIs to build a helpful and informative course recommendation chatbot trained on Stanford's course catalog data. We built two chatbots using the leading LLM chat APIs, OpenAI and Google's Natural Language API. We retrieved course data from the online course catalog, experimented with different means of preprocessing the training data to build our models, and created UIs for students to interact with the chatbots. We conducted in-depth, reference-free quantitative and qualitative analyses of each chatbot to evaluate performance and compare the two models. We compared our implementations against each other and a baseline model of OpenAI's ChatGPT 3.5. Both models demonstrated significant improvements over the baseline by our quantitative metrics. OpenAI answered more questions correctly; however, it also produced more false information. Findings from our qualitative user testing showed high variability and showed positive sentiment towards the technology. We conclude that students would benefit greatly from a course recommendation chatbot and that the Google-based chatbot should be improved upon to increase accuracy and implemented.

1 Key Information

- Mentor: Rashon Poole
- External Collaborators (if you have any): N/A
- Sharing project: N/A
- Group Contributions: Emma was responsible for developing the OpenAI API chatbot model and Naama was responsible for the Google Natural Language API chatbot model, and thier respective UIs. We both worked on data collection and preprocessing, development of evaluation metrics, testing, and analysis.

2 Introduction

When Stanford students look for information to select their courses for the upcoming quarter, they turn to ExploreCourses [1]. This website contains all the necessary information for students to select their classes; however, the user experience leaves much to be desired. The website is difficult to navigate, returns hundreds of results per query with dense information, and does not have natural language processing capabilities. One may turn to an existing large language model to assist in a task like course selection; however, most popular LLM models, like OpenAI's ChatGPT 3.5, are not trained on updated information, rendering them effectively useless for this particular use case.

We set out to analyze and compare two leading LLM chat APIs to build an alternative way for students to engage with Stanford's course catalog. This seemed like an interesting NLP project due to the specificity of the task, the availability of relevant natural language data, and the prevalence of competing chat APIs. The use of chatbots in an educational setting is well-established and well received, as evidenced in the papers detailed in the literature review.

Our chatbots aim to give students an avenue to directly ask questions about their courses and get accurate and helpful answers. For example, a student may input "I have taken CS106B and am interested in machine learning. What class should I take this Spring?" to which the chatbot may output "CS 129: Applied Machine Learning with prerequisites programming at the level of CS106B or 106X, and basic linear algebra such as Math 51."

We approached this task by building two chatbots using the leading LLM chat APIs, OpenAI's chatGPT and Google's Natural Language API. The first aspect of our work was accessing all of the relevant data and experimenting with different methods of parsing to improve the accuracy of the chatbots. Next, we trained both models on the data and created a user interface for students to interact with the chatbots. Then, we conducted thorough quantitative testing, evaluating the chatbot on all possible query types supported by the training data to determine the accuracy of the chatbot's responses. Simultaneously, we conducted qualitative analysis through user testing. Students were asked to interact with both chatbots, ranking and comparing their performance in categories such as correctness, fluency, consistency, and style. Finally, we used these metrics to evaluate which chatbot performs best on our task and to provide future recommendations for the foundations needed to build a successful course recommendation chatbot for Stanford students.

3 Related Work

AI chatbots are gaining popularity in academic settings to support students and administrators. Chatbots in education are primarily used to provide students with information at all hours and can customize the experience for each student.

Abu-Rasheed et al.[2] detail their work on a similar learning recommendation chatbot that utilizes LLMs integrated with a knowledge graph to create personalized student recommendations. The authors use prompt-context construction from various information sources such as conversation history, user data, and learning materials to improve the accuracy of the chatbot. Additionally, the chatbot classifies user queries into predefined categories to dictate what requests are within the chatbots scope and which should be redirected to a human mentor. However, the researchers list some limitations including a limited sample of users, a need for more extensive testing, and reports the chatbot's inability to properly contextualize at times. This study details the closest parallel to what we are trying to create; however, the specific educational training data and use context are ambiguous and the results leave something to be desired. Further, this paper is very recent and there is no mention of developing this concept further or actualizing the chatbot.

In a similar vein, Chia et al.[3] worked to create an intelligent course recommender chatbot using natural language processing. This chatbot is designed to assist students looking on the internet for courses that fit their goals and interests. The recommender engine processes students' inputs using neural networks for part-of-speech tagging to identify keywords that are matched with course descriptions using TF-IDF and cosine similarity. The user testing of this model showed favorable results, with users ranking the chatbot highly in terms of usability and functionality. However, the scope of this chatbot is quite surface level as it looks to match key phrases from the request to the course description, failing to provide more specific information on the course or allow any variations in terms of requests or inputs to the chatbot.

These two described chatbots differ in their use LLMs versus NLP, where LLMs provide wider context and NLP techniques focus on pattern matching and analyzing requests individually. The first chatbot by Abu-Rasheed is focused on understanding the student as a whole to provide a personalized experience, while the second focuses on answering specific questions without prior context or information about the user.

In the article, "How Universities Can Use AI Chatbots to Connect with Students and Drive Success" [4], the authors cite long-term usage since 2016 at many institutions. At Georgia State University, researchers and administrators reported chatbots significantly reducing "summer melt", a phenomenon when primarily low-income students get lost in the system between high school and college, by 22%. Further, the chatbots are useful in supporting first-generation low-income students who can feel their questions are not worth asking. Across the board, schools that utilize chatbots for administrative tasks have reported high rates of satisfaction and benefits to students. However, there must be an extremely high level of accuracy in these mechanisms to not be counterproductive.

Other researchers used Interpretative Phenomenological Analysis (IPA) with semi-structured interviews to explore students' perceptions regarding AI chatbots in higher education [5]. Largely, all sentiment was positive, conditioned on the accuracy of information. Other researchers examined the same question from a different perspective, by synthesizing the literature about AI chatbots in education to better understand uses, context, sentiment, and limitations [6]. They found numerous benefits, including personalized support for students and reduced administrators' workload. Their critical concerns pertained mainly to reliability and accuracy.

To summarize, it is clear that AI chatbots in educational settings are growing in popularity and perception. Stakeholders support the increased use of these chatbots to supplement learning and navigating the education systems. The biggest concerns and limitations of current AI chatbots are accuracy and hallucinations. Further, the literature on AI chatbots in education does not directly address the space we are attempting to address with our course recommendation AI chatbot, and there is plenty of room for growth in this area of research.

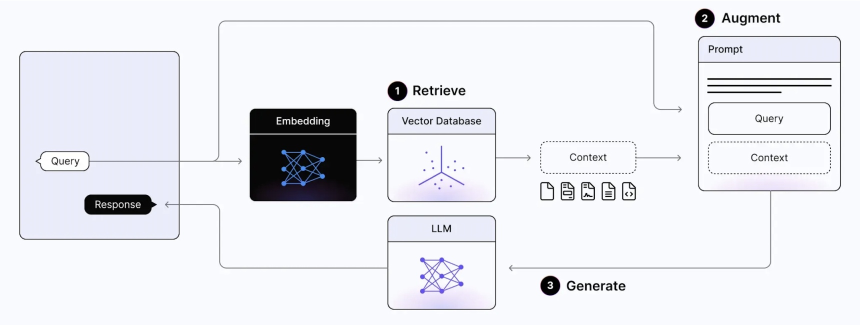
4 Approach

Our main goal for this project was to compare the performance of two existing LLMs on the task of providing course recommendations. We chose to work with two of the most popular LLM APIs, OpenAI GPT 3.5 and Google's Natural Language API. Both APIs allowed us to train on our large course information dataset and integrate a chatbot into an easy-to-use UI.

4.1 OpenAI API

When working with OpenAI API, we took inspiration from their Website Q&A with Embeddings [7] tutorial and its Python script [8] for building embeddings from a web crawl. We ran into obstacles with preprocessing the ExploreCourses webpage due to access errors. Our solution was to download data from the website as .txt files and train the model embeddings from this large dataset.

The basic architecture of the OpenAI model is RAG (Retrieve, Augment, Generate). The model trains vector embeddings for the data and query using OpenAI's Embeddings API with engine `text-embedding-ada-002`. It then builds a context for the query based on cosine similarity. Finally, it generates a response using `gpt-3.5-turbo-instruct` because of its compatibility with legacy completions endpoint. The model used in the tutorial, `text-davinci-003`, has been deprecated.



RAG systems come with inherent advantages and flaws [9]. The RAG allows us to use our own data, something that is crucial for the task at hand. It is more resistant to hallucinations due to its reliance on context, which also improves transparency. However, the reliance on contexts increases the risk that the model will not come up with an answer, and performance can be dependent on prompting techniques. It is dependent on chunking method, something we explore along with prompting later. Moreover, there are a lack of useful metrics to evaluate model performance.

4.2 Google Natural Language API

When working with Google's conversational API, we looked to their educational information page, Google Cloud Skills Boost [10], and utilizing outside resources as tutorials to create the chatbot [11]. The Google Cloud interface to create a customizable chatbot is called DialogFlow which provides an interface for people, such as ourselves. Ultimately the chatbot is built on Vertex AI Conversation which is a "single powerful platform for building conversational AI solutions that use Generative AI". This centralized platform uses Google's various API's in collaboration, such as the DialogFlow CX API which manages the conversational agents, Google Natural Langauge API to process users natural language inputs, and other API's to support speech to text. Vertex AI works by first ingesting, analyzing, and then transforming the inputted data which is done using managed databases and labeled annotation[12]. From there the transformed data is trained using Auto ML which is an automatic process of model algorithm selection, hyper parameter tuning, iterative modelling, and model assessment[13]. Next, the model is built and evaluated using using Explainable AI which is a "set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms" [14]. Next, the model is deployed using scalable hardware which is needed for lower latency and for online predictions. Finally, the model provides predictions.

4.3 UI Building

Another important step in developing our project was building a simple and intuitive user interface. One of the key methods we use for evaluating and comparing our chatbots is user testing. Since we planned on sending our chatbots out to students for testing, we needed to build a UI beyond the command line or Google Colab. These environments were sufficient for development; however, since this is a tool we hope to be used by the general student population, having a UI that is easy to use was crucial. We used Gradio [15], an open-source Python package that allows for the quick development of web applications for ML models. For the deployment of the chatbot based on Google API we utilized additional educational resources to deploy the model successfully [16].

5 Experiments

5.1 Data

We used Stanford University ExploreCourses [1] course data on all classes offered at Stanford University during the 2023-2024 school year, excluding the GSB, School of Education, BOSP, and School of Medicine. This data was downloaded, converted to a .txt file, preprocessed to remove unnecessary white space, and used to train model embeddings. The user will input a question and the model will process the query string and find its vector embedding to put through the model.

To get the best possible accuracy with our chatbot, we experimented with different datasets and techniques to process the training data and user input. We reached out to the managers of the ExploreCourses website to access the ExploreCourses API repository which provides public access to Stanford course data. We processed the data to extract all available information on courses, cleanly segment the information and the separate the courses from one another. However, performance on both models proved worse using this data, so we returned to the data files downloaded directly from ExploreCourses.

To train the models, the course data must be split into chunks of tokens. These chunks must include enough context to answer questions, must be small enough to be under the max token count for the model, and must be large enough to reduce training time to a reasonable length. We attempted various methods to divide the data, including dividing by course; however, the most efficient and effective method was dividing by sentence.

Moreover, we explored different means of processing user queries by breaking down prompts into ones that may be easier for the chatbot to parse. We arrived at the best overall results when the chatbot processes the natural language input directly from the user; however, there are some cases in which the model does not produce an answer to these prompts. In such a case, performance was improved by making an additional query with the user's same prompt. This solution works well for our task since the scope of prompts is limited to a few categories. For example, to the query "What quarter is JAPAN 24 offered?", the OpenAI API chatbot will reply "JAPAN 24 is not mentioned in the given context. Please rephrase your question or provide more information." However, if we break up the prompt on the back-end and query with "JAPAN 24 quarter", the reply is "The course JAPAN 24 is offered in the Winter quarter of the 2023-2024 academic year."

5.2 Evaluation method

Reference-free evaluation is the most relevant means of assessing the performance of our chatbots. While on a task such as language translation it makes sense to compare a model's translation against a human's, this is not the case for our recommendation chatbot. The key task in our project is information retrieval; we are less focused on the syntax and vocabulary in the response. While these kinds of reference-free scores were once nonstandard, they have since become increasingly popular.

We compared our models against ChatGPT 3.5. This model is not trained on up-to-date course information from the 2023-24 academic year; however, it does have some general knowledge of Stanford courses before its last update in January 2022. We expect our baseline to be able to perform tasks relating to course names or descriptions which remain relatively constant from year to year; however, it will likely perform very poorly on tasks related to scheduling where information is dynamic.

For quantitative testing metrics, we defined several prompts to which the chatbot can either answer correctly or incorrectly. These prompts tested the chatbots in a variety of categories supported by the data: course title, course description, instructor, prerequisites, quarters offered, time/day, location, units, course topics, UG requirements, prerequisites, cross-listings, times repeatable for credit, and grading basis. For each of the fourteen categories, we made five queries covering a wide range of departments for a total of seventy standardized questions. We recorded amount of questions answered correctly, incorrectly, and unanswered in each category to make more nuanced analyses, along with overall accuracy.

For qualitative testing, we had twenty Stanford students complete user testing by evaluating and blindly comparing the two chatbots through a series of in depth questions meant to draw attention to the different intended areas of knowledge. After experimenting with each bot, our survey asked users to rate each chatbot on a scale of one to five for each of the following criteria [17]: speed, fluency, coherence/consistency, correctness, commonsense, style, grammatically, and redundancy; as well as compare performance on the prompt categories above. Human evaluation is the gold standard of chatbot evaluations and so we wanted to place a focus on this area in our analysis.

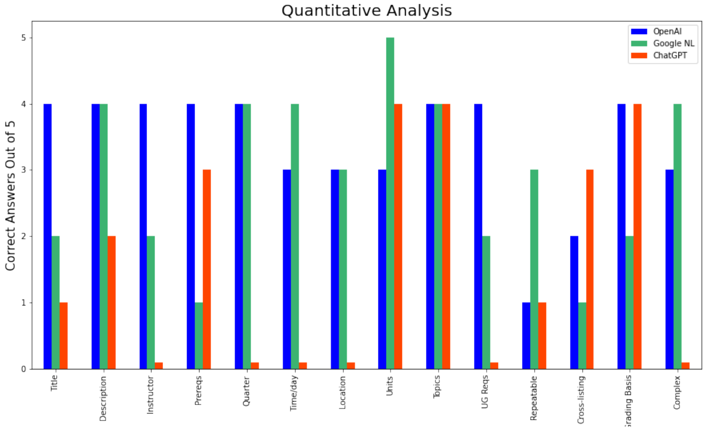
5.3 Experimental details

For the OpenAI chatbot, training on tens of thousands of courses took approximately 40 minutes. Responses are generated using `openai.Completion.create` with the model `gpt-3.5-turbo-instruct`. The temperature parameter is set to 0, meaning answers are deterministic and focused. Frequency and presence penalties are also set to 0, leading to a balanced preference for unseen words.

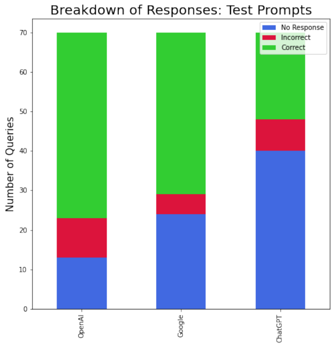
For the Google Natural Language API chatbot, training on tens of thousands of courses took approximately 20 minutes. Responses are generated using generators sourced from `Vertex AI Conversation`. The temperature parameter is set to 0. Frequency and presence penalties are also set to 0.

5.4 Results

As anticipated, both models showed significant performance improvements over ChatGPT. The OpenAI API chatbot answered 47 out of 70 questions correctly, or 67.1%. The Google API chatbot answered 41 out of 70 questions correctly, or 58.6%. ChatGPT answered 22 out of 70 questions correctly, or 31.4%. ChatGPT performed well on descriptive tasks, as we hypothesized. Differences in our model and the baseline were most obvious for scheduling tasks (quarter, time, day, location, instructor).



Directly comparing the two models, we see that OpenAI outperformed Google in 6 of 14 categories, while Google outperformed OpenAI in 4 of 14 categories. They performed the same in the remaining 4 categories.



Another interesting point of comparison between our models is the distinction between the bots providing no response versus an incorrect response. Providing no response and instead giving a prompt for the user to rephrase their question is preferential to providing a hallucinated incorrect answer because we do not want our chatbot to spread misinformation. Of the 23 not correct responses by the OpenAI chatbot, 10/23 or 43% were incorrect answers compared to 5/29 or 17% for Google API and 8/48 or 17% for the ChatGPT baseline. So, while OpenAI API seemed to outperform Google NL API on this evaluation, the "wrong" answers by the Google chatbot were more desirable for our use case.

Our quantitative results surprised us. The chatbot performed surprisingly well on complex queries that combined aspects of scheduling and description into one prompt. However, it sometimes struggled to answer simple queries on questions as straightforward as asking the number of units of a course.

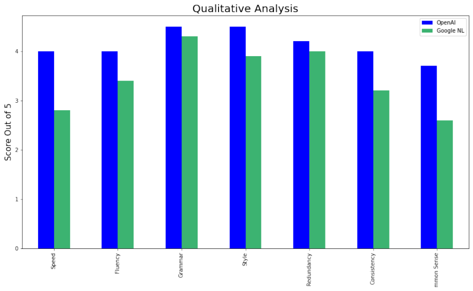
6 Analysis

We found difficulty in predicting when our models would fail and when they would succeed. Similar to the ChatGPT baseline, both performed better on descriptive tasks that allowed for more freedom in interpretation and response generation.

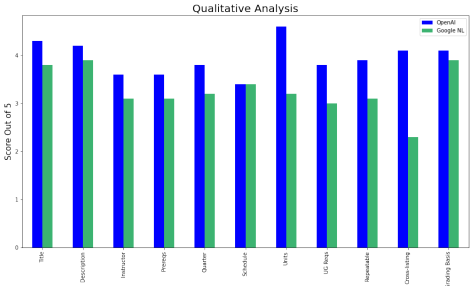
The chatbots often failed to provide responses for a question about one course, but would correctly answer the same question about another course. One explanation for this could be the chunking method, as some information about a class could be left out of the context for the specific task.

Additionally, the OpenAI API chatbot had a tendency to hallucinate responses, feeding the user false information. While preventing hallucinations is often an advantage of a RAG system, some of this may be explained by the plethora of mentions of popular classes like CS106B or MATH51 which are prerequisites for many courses and thus would be included in many course descriptions outside of their own.

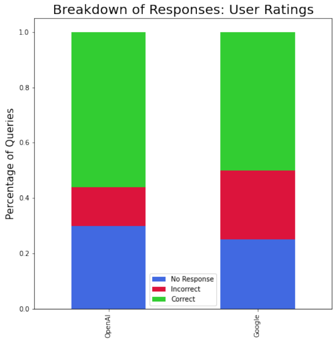
The qualitative results from user testing were promising and somewhat more decisive. In the criteria outlined in the methods section, the OpenAI chatbot outperformed the Google NL chatbot in every category, but not always by a wide margin. Moreover, our user evaluations had a large variance, so to get a better assessment, more testing with a larger sample would be recommended.



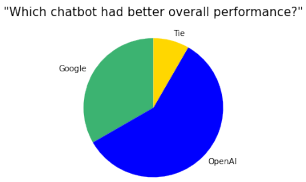
The OpenAI API chatbot also outperformed the Google API chatbot in every prompt category according to user evaluations; however, margins were slim and further testing should be done to further evaluate.

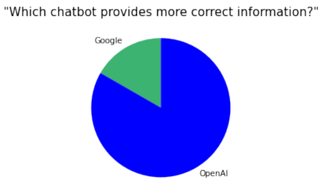


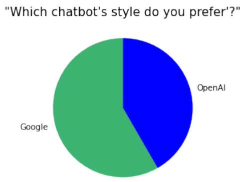
When comparing responses reported by users, the breakdown contrasts that of our quantitative results. Users claimed that about 32% of the questions which were not correctly answered for OpenAI API produced false infromation compared to 50% of Google NLP API. The reported rates of correct answers were lower, but still fairly consistent with our previous results (about 56% for OpenAI, 50% for Google). This leaves plenty of room for improvement.



Users preferred the OpenAI chatbot in terms of overall performance and accuracy, as supported by other responses. However, a large number preferred the style of the Google chatbot, with their reasoning being expressed well by one user, "[Google chatbot] seemed more concise, but [OpenAI chatbot] was notably more accurate."


"Which chatbot had better overall performance?"


"Which chatbot provides more correct information?"


"Which chatbot's style do you prefer'?"

7 Conclusion

Our work is a preliminary investigation into the utilization of existing AI models to support student learning through a course recommendation chatbot. We have found promising results from quantitative testing and user sentiment. After looking through the related literature, we were unable to find a chatbot that fulfills the same role as ours, but we did find many uses of AI chatbots in education with positive reception.

From our quantitative analysis, we found that both chatbots outperformed the current alternative, Chat GPT 3.5. We struggled to find consistency in our results with both chatbots trained on the same curated data, and the chatbot's success on similar questions varied. We were surprised to find that both chatbots responded with high accuracy to complex questions; however, the chatbots struggled with some simple questions.

Our qualitative user testing results varied from the stricter quantitative testing, as expected. Overall, users reported more correct information produced by Open AI and a majority of users reported better overall performance by OpenAI. However, when asked about which chatbot's style they preferred Google.

While the OpenAI API chatbot has a higher rate of accuracy than the Google API chatbot, the OpenAI chatbot hallucinates, at a much higher rate. We conclude that we would recommend the Google API chatbot. Although it does not provide correct answers as often, it rarely provides false information. This aligns with our ethical concerns since we hope to avoid the spread of misinformation. Additionally, it is more difficult to screen for hallucinations as opposed to improving the accuracy of the chatbot. Thus, pursuing further improvements to the Google API is a more promising route. Overall, our findings support further developing the Google API chatbot by improving accuracy to reliably aid and assist students in the course selection process.

8 Ethics Statement

The main ethical concern with our project, as with any chatbot, is the spreading of misinformation. With generative AI, there is always a risk of hallucinating false information and providing it to the user as true. Should a student follow flawed advice, it could have negative consequences, affecting their future education and career. To mitigate this risk, the chatbot can advise students to confirm results with the official course catalog before enrollment. To further mitigate this concern, a tedious yet effective method is to hard code much of the data from the course catalog into the framework of the chatbot as is done by many professional companies. Additionally, the training data should be regularly updated as changes are made to the course catalog.

An additional ethical challenge regards protecting privacy and data security. Students using the chatbot provide personal information about their preferences and class history which they may not want to be made public. This can be mitigated by deleting all user data when the chatbot completes a session with a user. An alternative mitigation approach is to include security measures which would allow the users of the chatbot to save their personal history to have a customized chatbot without the threat of their information being publicized or misused. A couple of security measures that should be added would be to securely encrypt user data, require users to sign in to have access to their personalized chatbot and information, and the chabot should store the minimum amount of information needed to be personalized to the user. Furhter, we would recommend addressing the OWASP Top 10 LLM compromises and using a standardized data secutiry protocol such as GDPR (General Data Protection Regulation)[18].

References

[1] Stanford University. Explore courses. https://explorecourses.stanford.edu/.

[2] Christian Weber Hasan Abu-Rasheed, Mohamad Hussam Abdulsalam and Madjid Fathi. Supporting student decisions on learning recommendations: An llm-based chatbot with knowledge graph contextualization for conversational explainability and mentoring, Janruary 2024.

[3] Suraya Alias Ag Asri Ag Ibrahim Mohd Fairuz Iskandar Othman Tan Chia Wei, Mohd Hanafi Ahmad Hijazi. Intelligent course recommender chatbot using natural language processing, 2022.

[4] Andy Viano. How universities can use ai chatbots to connect with students and drive success. https://edtechmagazine.com/higher/article/2023/02/how-universities-can-use-ai-chatbots-connect-students-and-drive-success, 2023.

[5] Soniya Antony and R. Ramnath. A phenomenological exploration of students' perceptions of ai chatbots in higher education. https://files.eric.ed.gov/fulltext/EJ1403709.pdf#:~:text=URL%3A%20https%3A%2F%2Ffiles.eric.ed.gov%2Ffulltext%2FEJ1403709.pdf%0AVisible%3A%200%25%20, 2023.

[6] Maya Grigolia Lela Machaidze Lasha Labadze. Role of ai chatbots in education: systematic literature review. https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-023-00426-1, 2023.

[7] OpenAI. How to build an ai that can answer questions about your website. https://platform.openai.com/docs/tutorials/web-qa-embeddings.

[8] Simón Fishman, Logan Kilpatrick, and John Tregoning. Web q&a with embeddings. https://github.com/openai/web-crawl-q-and-a-example/tree/main.

[9] Pratik Bhavsar. Mastering rag: Improve rag performance with 4 powerful rag metrics. https://www.rungalileo.io/blog/mastering-rag-improve-performance-with-4-powerful-metrics, 2024.

[10] Google Cloud Skills Boost. Vertes ai search and conversation architecture. https://www.cloudskillsboost.google/course_templates/892, 2023.

[11] Coding Money. How to build ai chatbot with custom knowledge base. https://www.youtube.com/watch?v=6hQF80_xMkQ.

[12] Google Cloud. What is vertex ai. https://www.youtube.com/watch?v=Rl9Gz1Yf6wM, 2024.

[13] Google Cloud. Automl. https://cloud.google.com/vertex-ai/docs/beginner/beginners-guide, 2023.

[14] IBM. What is explainable ai? https://www.ibm.com/topics/explainable-ai#:~:text=What%20is%20explainable%20AI%3F,expected%20impact%20and%20potential%20biases., 2021.

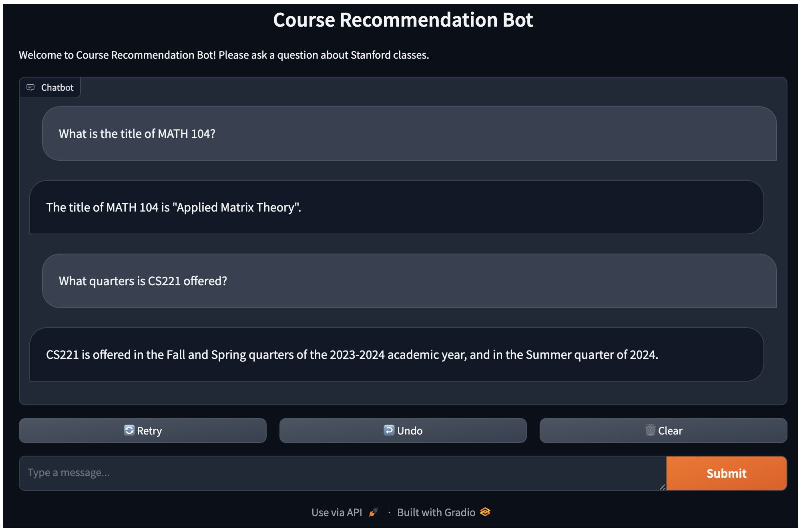
[15] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019.

[16] koverholt. Gradio app + dialogflow cx agent. https://github.com/koverholt/gradio-dialogflow-cx, 2024.

[17] Yann Dubois. Lecture 11: Benchmarking and evaluation, May 2024.

[18] Ravinder Verma. Building secure chatbots: Best practices that ensure privacy. https://www.nagarro.com/en/blog/building-secure-chatbots-best-practices, 2023.

A Appendix



Course Recommendation Bot

Welcome to Course Recommendation Bot! Please ask a question about Stanford classes.

What is the title of MATH 104?

The title of MATH 104 is "Applied Matrix Theory".

What quarters is CS221 offered?

CS221 is offered in the Fall and Spring quarters of the 2023-2024 academic year, and in the Summer quarter of 2024.

Test Group	OpenAI API	Google API	ChatGPT 3.5
Course title	4/5	2/5	1/5
Course description	4/5	4/5	2/5
Instructor	4/5	2/5	0/5
Prerequisites	4/5	1/5	3/5
Schedule (Quarter)	4/5	4/5	0/5
Schedule (Time and day)	3/5	4/5	0/5
Location	3/5	3/5	0/5
Units	3/5	5/5	4/5
Topics	4/5	4/5	4/5
UG Requirements	4/5	2/5	0/5
Number of times repeatable	1/5	3/5	1/5
Cross listings	2/5	1/5	3/5
Grading basis	4/5	2/5	4/5
Complex query	3/5	4/5	0/5
Total	47/70 (.671)	41/70 (.586)	22/70 (.314)