CS 410 Project Proposal

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.

Team Name: CourseBuddyAI
Team Members:
Avinash Badeo (abaldeo2@illinois.edu) - Team Captain
Zach Pohl (zcpohl2@illinois.edu)
Colton Bailey (coltonb4@illinois.edu)
Ehsan Sarfaraz (ehsans3@illinois.edu)
Kacper Dural (kdural2@illinois.edu)

2. What topic have you chosen? Why is it a problem? How does it relate to the theme and to the class?

For our project we are proposing to build a chrome extension to extend the Coursera Online Learning Platform and enable students to upload lecture transcripts and slides (pdf/ppt) to get summaries. Those files will also be indexed in a vector database and then used that to build a Q&A chatbot interface using the Retrieval Augmentation Generation (RAG) technique in combination with a large language model (i.e., ChatGPT/LLAMA2). We feel such a tool is needed because it can help enhance student learning when they have questions on lecture topics. This may also reduce the burden of TAs having to answer the same question repeatedly on Campuswire. This project can be considered related to the theme of the class since we are using a state-of-the-art LLM to perform NLP tasks such as text summarization and Question-Answering using information retrieval augmentation. Although a large language model may already be trained to answer trivia questions such as “What is precision”, different courses may have different terminology or explanation for the same concepts, so it’s important for it to be able to answer the students question guided by the course context. This will also help reduce the chance of hallucinations. Although these topics are not yet taught in the class, they are considered cutting edge for text information retrieval and are actively being explored by many researchers and AI startups.

3. Briefly describe any datasets, algorithms or techniques you plan to use

The dataset we plan to use will be all the lecture transcripts & slides available for download under the CS410 course videos on Coursera. We will take these documents and pass them through an embedding model to generate dense vectors that can be indexed in a vector database. When a question is asked by the user, the system will generate the embeddings for the query then perform cosine similarity search to find relevant pieces of the lecture content that can help the LLM accurately answer the question. We will take the question and retrieve relevant context and pass it the LLM prompt then store the response in a cache/database. This will help improve the system performance if other students have

similar questions. Due to the limitation of similarity search (i.e. short keyword queries), we also plan to explore the potential of doing hybrid search, meaning we will perform both semantic similarity search on the summarized documents and BM25 search on the full text documents, then take the normalized weighted sum of two in order to find the most relevant pieces of context for the question.

4. How will you demonstrate that your approach will work as expected?

We will demonstrate that the approach has worked as expected by using an LLM to generate a dataset of questions and relevant context, then evaluating common metrics such as MRR & NCDG. This way we can compare the results of doing hybrid search vs semantic similarity search alone.

For our project demo, we plan to show a user asking a question using the chatbot and getting back responses along with citations to the sources of the course content it used to answer the question. Note, due to the costs of LLM API we may cache a pre-chosen list of questions ahead of time and use them for our demo.

5. Which programming language do you plan to use?

For our project we will use Python for the backend and Javascript for the frontend (chrome extension). We chose Python because of the rich ecosystem of libraries that can be leveraged for integrating with an LLM such as Langchain/LlamaIndex.

6. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

Below are the main tasks for the project. In total it is about 125 hours of effort, which split across 5 members and eight weeks equates to about 3 hours per week.

Task #	Description	Estimated Time Cost
1	Define project scope and objectives.	1 week
2	Research and select appropriate development tools and frameworks.	1 week
3	Setting up the project infrastructure, including the development environment and project repository	1 week
4	Design the user interface of the Chrome extension, including file upload, summarization, and chatbot triggers.	1 week

5	Implement the file upload functionality to allow users to upload lecture transcripts and slides.	1 week
6	Implement text preprocessing to clean the uploaded text data (removing HTML tags, special characters, URLs).	1 week
7	Implement text summarization using LLM	1 week
8	Implementing the text embedding to generate dense vectors from the uploaded documents	2 weeks
9	Set up a vector database and implement indexing for documents	1 week
10	Implement a Q&A chatbot interface using the Retrieval Augmentation Generation (RAG) technique.	3 weeks
11	Integrate a large language model (e.g., ChatGPT or LLAMA2) into the chatbot for answering questions.	1 week
12	Create a cache/database system to store responses for similar questions to improve system performance.	1 week
13	Investigate and implement hybrid search combining semantic similarity and BM25 search for improved context retrieval.	2 week
14	Generate a dataset of questions and relevant context to evaluate the system's performance using metrics like MRR & NDCG.	2 week
15	Develop a mechanism to provide citations to the sources of course content used in chatbot responses.	1 week
16	Conduct testing and debugging to ensure the system works as expected.	1 week
17	Document the project, including user guides and technical documentation.	1 week
18	Prepare a project presentation demo.	1 week
19	Conduct final testing, performance optimization, and quality assurance.	1 week
20	Submit the project and prepare for presentation to the class.	1 week