Tulane Academic Advisor With RAG

Gavin Galusha
Tulane University / CA
ggalusha@tulane.edu

Gabe Epstein
Tulane University / CO
gepstein1@tulane.edu

Abstract

We created a Retrieval Augmented Generation system with the goal of answering Tulane specific academic questions. By web-scraping Tulane's website, we were able to create a generative question answering system that was knowledgable on two main sectors: Tulane Programs, and Tulane Courses. We paired this retrieval system with a state of the art LLM (gpt-3.5-turbo) using it's chat completion feature to generate high quality responses to Tulane specific queries. After experimenting with a variety of retrieval methods, we landed on a chunking technique for the documents, followed by pre-processing, and dynamic top k retrieval, which proved to be a reliable way to obtain the relevant context for the LLM. To make our system easily usable, we implemented a web interface using flask, which allows users to toggle which data they want to interact with, and ask questions in an intuitive manner.

1 Introduction

Contacting academic advisors can be an arduous process. Advisors are often bombarded with easily answerable questions, they can be hard to get ahold of, or perhaps simply unqualified for their position. Our goal was to create a system that would streamline information from readily available public information online, directly to a student.

The emphasis on this project was to build a robust retrieval system, so that in the future as more and more data is available, the system improves relationally.

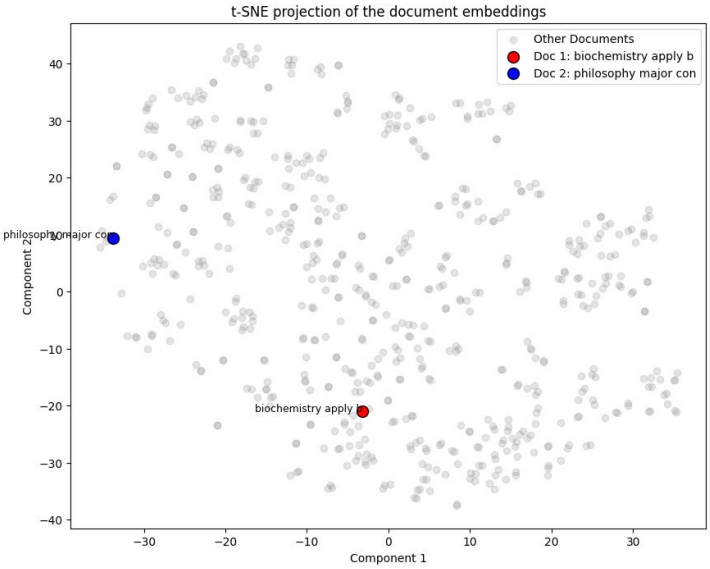
3 Approach

Web Scraping

Our first step in the overall process was to collect the relevant data. We used Beautiful Soup to parse the Tulane courses page, where we were able to extract relevant text data. After obtaining a dictionary of each program name as the keys, and the associated links as the values, we parsed the two main pages for each program, the requirements and home tabs found under each program. From here, we concatenated all the relevant text, and saved it to a directory where each program was divided into text files based on the topic. The result was a directory of over 400 files, with a hearty amount of text dedicated to each program.

A similar procedure was done for the Tulane courses page, only we had to employ further text manipulation to accurately extract each specific class offered. We used regex matching to create a new entry every time four capital letters were seen, indicative of class descriptions like CMPS and ACCN. We did further filtering out for courses that had no descriptions, or special courses like "Independent Study" which would offer no relevance in our retrieval system. The output of this were over a thousand very small class files, each with the course code, and whatever information about the class was given.

This web scraping process used no hard encoded values besides the links to the websites, so the web scraper can easily be run and obtain any new courses, programs, or just general information published on the Tulane website.



5 Data Preparation

There are two primary datasets that we have used for this project: https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset/code and https://www.kaggle.com/datasets/ashpalsingh1525/netflixLinks. The first dataset contains a variety of dataframes all linked together with keys like userId, movieId, and tagId. This one is essential for user-user collaborative filtering.

The second dataset contains specific movie attributes, like description, duration, cast, etc. This one is ideal for calculating movie-movie similarity. The first step was to join together the various dataframes, and string together a cohesive data frame with no missing values.

6 Conclusion

The evaluation of our recommendation system will rely on objective measurements derived from the system's performance on a held-out test dataset. We will use metrics such as precision, recall, F1 score, and Mean Average Precision (MAP) to quantify the system's accuracy in predicting user preferences based on their input and interaction history. Comparison with a baseline model, such as traditional collaborative filtering without real-time user input, will highlight improvements. We'll also utilize confusion matrices to analyze the model's performance in different recommendation scenarios, providing a detailed view of its strengths and areas for improvement. This method ensures an objective and comprehensive evaluation without the need for live deployment or user feedback.

7 Division of Labor

The work was divided between Gabe Epstein and Gavin Galusha on this project. Both members collaborated frequently, but Gavin focused more on the data gathering and pre-processing, while Gabe did more model evaluating and fitting