Tulane Academic Advisor with RAG

Gavin Galusha
Tulane University / CA
ggalusha@tulane.edu

Gabe Epstein
Tulane University / CO
gepstein1@tulane.edu

1 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.

2 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.

2 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.

Data Preparation

To turn our text file data into an easily retrievable format, we first needed to create embeddings for the documents. During data preprocessing, we used the Spacy library to clean the text in each document. This included normalizing the text to all lowercase, removing punctuation and stop words, and applying Lemmatization. In this process, we also used Named Entity Recognition to determine if each entity in a document was the name of a program or class, in which case the entire phrase name would be added to the processed text output, rather than a stripped version.

This pre-processing is an essential step to generate richer embeddings for the documents, aiding in effective retrieval.

Chunking

To further improve retrieval, we employed a chunking technique to split up each document into smaller parts. We found success with relatively large chunks, around 5000 characters, with 100 character overlaps. This improved the number of documents retrieved from around -13 to 20-30 for the programs data. No chunking was needed for the course data as documents are already sufficiently small.

Vectorization

We experimented with dense embeddings created by models like BERT and the Openai model text-embeddings-3-small , but concluded that they provided no significant advantage over simple TF-ITF vectorization.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Retrieval

The actual retrieval method used was a combination of cosine similarity and keyword priority. the retrieval takes an input query (a sentence) and vectorizes it into the same space as the document embeddings using TF-ITF again. From here we calculate a score for each document based on the cosine similarity of the vectors.

$$similarity(A,B) = cos(\theta) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \sum_{i=1}^{n} B_i^2}}.$$

We add this to the keyword score, which is calculated by taking the set of all word in the query, and searching for those words in the title of each document. This final score is then added to a list called matches with the associated document.

The next step in retrieval is to return the top k to form the context for the LLM. In our case, the LLM we are using has a context window of around 16,000 tokens, so in order to always fill that context window, we employed a dynamic top k approach.

Dynamic Top K

To optimize the amount of documents retrieved with each query, we used tiktoken to count the number of tokens appended to the context window. This way, we can start at the top of the matches list, getting the documents with the highest combined scores, and iteratively retrieve more documents until the maximum token limit is reached. This step proved essential, as it is effective on databases of documents with greatly varying lengths, and no top k variable is needed to be hardcoded.

Experiments

Top_k

Our experiments led us to many conclusions that would aid our project. The first was an attempt to combine both the courses and programs data in a top_k format. This quickly turn problematic, as the top_k documents may easily go far over or far under the context window, resulting in an error. This was a product of documents with too much varying length. After implementing the dynamic top_k strategy to deal with this issue, we still found confounding results.

Data Separation

Providing a slew of courses data with programs data seemed to be a confounding context for the LLM, as the response output accuracy took a sharp fall. uestions the system had easily answered like What are the major requirements for x major were answered, but with a jumble of misinformation not consistent with the actual website specifications. This led to the idea of separating the question answering into two modes, class expert, and programs expert. These two modes isolate the data that can be retrieved in the process, proving to faster and more accurate retrievals on respective tasks.

Files:

Another critical in the experimenting process was to save the pre-processed data to a file for easy use. The pre-processing function could take over a minute to run, and saving this information to be easily loaded in on app launch is much more efficient.

Web Scraping:

Initially we tried obtaining our course data from the schedule of classes page, using selenium to iteratively scrape every page from every subject. This proved challenging, and we pivoted to simply grabbing the corse descriptions instead.

Related Work:

Regarding related work, we explored a number of papers and projects that share similar goals or methods with our exploration.

An important work presented at NeurIPS 2020 by Lewis et al., titled Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, significantly enhances seq2seq models by integrating them with a retrieval-augmented generation (RAG) system (Lewis et al., 2020). This system innovatively combines parametric memory—a pre-trained seq2seq model—with non-parametric memory—a dense vector index of Wikipedia—to improve performance on various NLP tasks. The paper aims to address the limitations of pre-trained NLP models, which typically cannot access external information, thus restricting their ability to expand or revise their inherent knowledge. Similar to our approach, the RAG system utilizes a pre-trained retriever alongside a seq2seq model, employing Maximum Inner Product Search to identify top-K documents. These documents are then used, marginalized over seq2seq predictions, to generate more accurate and contextually relevant responses. The core concept underpinning our model mirrors this approach, providing a comprehensive insight into the mechanics of retrieval-augmented generation systems. The paper describes two main RAG configurations: RAG-Sequence and RAG-Token models. The RAG-Sequence model uses the same set of retrieved documents to generate the entire sequence, ensuring consistency across the generated content. In contrast, the RAG-Token model enhances flexibility by allowing each token in the output sequence to be conditioned on different retrieved documents, treating each token's context as a separate latent variable.

While this approach offers greater granularity and adaptability, it is more computationally intensive due to the need for repeated retrieval operations for each token in the sequence. More recently, the 2024 paper by Lakatos et al., titled 'Investigating the Performance of Retrieval-Augmented Generation and Fine-tuning for the Development of AI-driven Knowledge-based Systems' examines the efficacy of Retrieval-Augmented Generation (RAG) versus Fine-tuning (FN) methods in optimizing generative large language models (G-LLMs) for specific domains (Lakatos et al., 2024). Utilizing a variety of NLP performance metrics such as ROUGE, BLEU, METEOR, and cosine similarity across models including GPT-J-6B, OPT-6.7B, and LlaMA, the research assesses the effectiveness of these methodologies. The authors' findings reveal that RAG typically outperforms FN, providing more reliable retrieval with less risk of hallucination and eliminating the need for continuous retraining. Interestingly, combining RAG with FN does not consistently enhance performance and can sometimes reduce it. Similarly, our project leverages RAG to improve system functionality, aiming to deliver precise and relevant outputs to our user's queries. While our project specifically addresses academic advising queries at Tulane using dynamically scraped web data, the paper explores broader applications of G-LLMs across various domains using structured datasets. The paper's systematic evaluation of RAG and FN across diverse models and datasets reinforces the approach utilized in our project, providing a foundational backing for our methodological choices.

On the topic of creating academic advising enhancements with AI, notable paper from the Higher Colleges of Technology and The British University in Dubai in 2022, titled "AI-based Academic Advising Framework: A Knowledge Management Perspective," addresses challenges in traditional academic advising from a knowledge management perspective (Bilquise & Shaalan, 2022). The authors propose an integrated AI-based framework aimed at enhancing the advising process in higher education institutes. Their study identifies inefficiencies in knowledge management due to high student-to-advisor ratios and inadequate personalization in advising. They suggest a comprehensive AI-based system in three parts: a rule-based expert system for study plans that automates the course selection and planning process, a machine learning model for identifying at-risk students to allow timely interventions, and a conversational AI chatbot for digital assistance. The latter aligns closely with our project, which also employs conversational AI to interact with users. While their paper aims to address a broad range of academic advising tasks across various aspects of student and institutional needs, our focus is more specific. We are developing a Retrieval Augmented Generation system tailored specifically to answering Tulane-specific academic queries by dynamically scraping web data for real-time querying and response generation. Our system utilizes a dynamic top-k retrieval mechanism to handle variations in document sizes and content, which ensures the use of the most up-to-date information from Tulane's frequently updated web sources. In their proposed framework, a student inputs a query in natural language through a user interface. This query is processed on the backend using an NLP engine, which converts the text to structured data, extracts the intent, and generates an appropriate response. Similarly, our project uses advanced NLP techniques to understand and respond to user inquiries effectively, though our application specifically leverages the continuously updated content from the Tulane website.

The paper 'A Natural Language Conversational System for Online Academic Advising' by Latorre-Navarro and Harris outlines the development of an academic advising system utilizing NLP to automate responses to frequently asked academic questions at the University of Florida (Latorre-Navarro & Harris, 2012). Designed to free human advisors from repetitive tasks, the system embeds knowledge of teaching schedules, degree requirements, course prerequisites, and administrative procedures into a pattern-matching dialogue management system accessible via a web browser. Operating on a rule-based dialogue management using ChatScript, this system aims to provide instantaneous academic advice without requiring student training. Similar to our project at Tulane, it leverages NLP to enhance the advising experience; however, while our project employs a Retrieval Augmented Generation system that dynamically. pulls and processes data in real-time, the system described by Latorre-Navarro and Harris relies on a more static, pattern-matching approach based on a predefined knowledge base. Their project, developed over a decade ago, anticipated many of the advancements in academic advising systems by introducing automation and immediate response capabilities. Despite its innovative approach for the time, it lacks the adaptive and real-time updating capabilities inherent in more contemporary systems like ours at Tulane, which utilize cutting-edge technology to adjust to ongoing changes in academic data and student needs. The evolution from their system to ours illustrates significant progress in the field of educational technology, showcasing how advances in AI and machine learning have transformed the possibilities for academic advising systems.

The paper titled 'Use of a ChatBot-Based Advising System for the Higher-Education System,' presented at the International Conference on Advances in Technology and Computing (ICATC 2023) by Radhakrishnan and Dias, explores the implementation of a chatbot-based advising system within the higher education context of Sri Lanka (Radhakrishnan & Dias, 2023). This system aims to alleviate the workload on academic advisors by employing a Large Language Model (LLM), vector databases, and semantic search technologies to automate responses to common student inquiries. These technologies enable the chatbot to utilize a custom-built knowledge base, sourced from static institutional documents such as handbooks and course outlines, ensuring that the responses are both accurate and relevant.

In contrast, our project at Tulane uses a sophisticated pipeline that begins with web scraping using BeautifulSoup and Regex to dynamically collect Tulane-specific information. This information is then processed and vectorized using Spacy and TF-IDF to enhance text quality and relevance. Our retrieval system employs cosine similarity and dynamic token counting to achieve a nuanced alignment between query and document vectors, significantly enhancing the precision of the responses. These responses are then generated through our Retrieval Augmented Generation system, which integrates the GPT-3.5-turbo model via an API call.

Unlike the ICATC system, which features a dual-interface design that requires advisors to actively manage the knowledge base and includes conversational memory—a feature we plan to integrate—our system focuses on dynamic document retrieval. This approach adjusts the scope of retrieved documents based on the complexity and richness of the queries, enhancing response relevance without necessitating continuous updates to the knowledge base. The ICATC prototype has shown high response accuracy and operational efficiency, providing a valuable model for the application of AI in academic advising.

While this approach offers greater granularity and adaptability, it is more computationally intensive due to the need for repeated retrieval operations for each token in the sequence. More recently, the 2024 paper by Lakatos et al., titled 'Investigating the Performance of Retrieval-Augmented Generation and Fine-tuning for the Development of AI-driven Knowledge-based Systems' examines the efficacy of Retrieval-Augmented Generation (RAG) versus Fine-tuning (FN) methods in optimizing generative large language models (G-LLMs) for specific domains (Lakatos et al., 2024). Utilizing a variety of NLP performance metrics such as ROUGE, BLEU, METEOR, and cosine similarity across models including GPT-J-6B, OPT-6.7B, and LlaMA, the research assesses the effectiveness of these methodologies. The authors' findings reveal that RAG typically outperforms FN, providing more reliable retrieval with less risk of hallucination and eliminating the need for continuous retraining. Interestingly, combining RAG with FN does not consistently enhance performance and can sometimes reduce it. Similarly, our project leverages RAG to improve system functionality, aiming to deliver precise and relevant outputs to our user's queries. While our project specifically addresses academic advising queries at Tulane using dynamically scraped web data, the paper explores broader applications of G-LLMs across various domains using structured datasets. The paper's systematic evaluation of RAG and FN across diverse models and datasets reinforces the approach utilized in our project, providing a foundational backing for our methodological choices.

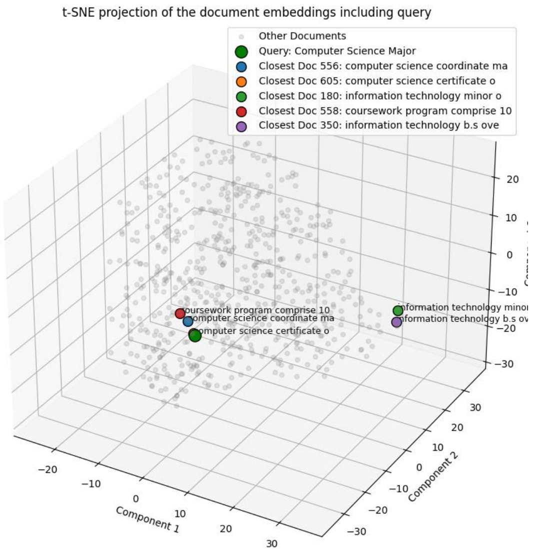
On the topic of creating academic advising enhancements with AI, notable paper from the Higher Colleges of Technology and The British University in Dubai in 2022, titled "AI-based Academic Advising Framework: A Knowledge Management Perspective," addresses challenges in traditional academic advising from a knowledge management perspective (Bilquise & Shaalan, 2022). The authors propose an integrated AI-based framework aimed at enhancing the advising process in higher education institutes. Their study identifies inefficiencies in knowledge management due to high student-to-advisor ratios and inadequate personalization in advising. They suggest a comprehensive AI-based system in three parts: a rule-based expert system for study plans that automates the course selection and planning process, a machine learning model for identifying at-risk students to allow timely interventions, and a conversational AI chatbot for digital assistance. The latter aligns closely with our project, which also employs conversational AI to interact with users. While their paper aims to address a broad range of academic advising tasks across various aspects of student and institutional needs, our focus is more specific. We are developing a Retrieval Augmented Generation system tailored specifically to answering Tulane-specific academic queries by dynamically scraping web data for real-time querying and response generation. Our system utilizes a dynamic top-k retrieval mechanism to handle variations in document sizes and content, which ensures the use of the most up-to-date information from Tulane's frequently updated web sources. In their proposed framework, a student inputs a query in natural language through a user interface. This query is processed on the backend using an NLP engine, which converts the text to structured data, extracts the intent, and generates an appropriate response. Similarly, our project uses advanced NLP techniques to understand and respond to user inquiries effectively, though our application specifically leverages the continuously updated content from the Tulane website.

References

- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Proceedings of NeurIPS.
- Lakatos, R., Pollner, P., Hajdu, A., & Joo, T. (2024). Investigating the Performance of Retrieval-Augmented Generation and Fine-tuning for the Development of AI-driven Knowledge-based Systems.
- Bilquise, G., & Shaalan, K. (2022). AI-based Academic Advising Framework: A Knowledge Management Perspective.
- Latorre-Navarro, E. M., & Harris, J. G. (2012). A Natural Language Conversational System for Online Academic Advising.
- Radhakrishnan, H. K., & Dias, N. G. J. (2023). Use of a ChatBot-Based Advising System for the Higher-Education System. Proceedings of ICATC.

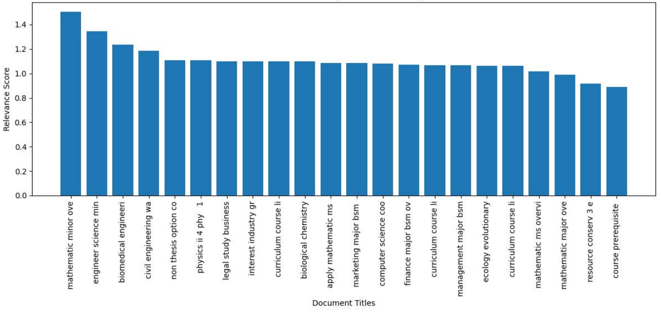
Results

The resulting functionality of the retrieve method is accurate at retrieving the relevant chunks of data. In this example we reduce dimensionality to 3d space, and as an example show the most similar chunked documents to the query: "Computer Science Major"



Our retrieval function average time to obtain documents hovered around 0.32 seconds on over 100 training instances. This time could be reduced greatly by specifically taking out the keyword importance in our function, but from a user experience perspective anything less then a second is negligible.

Furthermore, our retrieval score calculations are working effectively, as seen by in this graph where the math Minor query appropriately scored the math minor chunk with the greatest score.


Relevance Scores for Query: "Math minor requirements"
Relevance Score
mathematic minor ove
engineer science min
biomedical engineeri
civil engineering wa
non thesis option co
physics ii 4 phy   1
legal study business
interest industry gr
curriculum course li
biological chemistry
apply mathematic ms
marketing major bsm
computer science coo
finance major bsm ov
curriculum course li
management major bsm
ecology evolutionary
curriculum course li
mathematic ms overvi
mathematic major ove
resource conserv 3 e
course prerequisite
Document Titles