

11-711 Assignment 2 : Retrieval Augmented Generation System

Rajeev Veeraraghavan *

rveerara@andrew.cmu.edu

Abstract

This report summarizes the work done to build a stand alone Retrieval augmented generation system to answers questions on CMU (Carnegie Mellon University). The system uses publicly available data to build a knowledge base, and LLama2-7b model to generate answers to questions.

1 Introduction

2 Data Creation

2.1 Raw Data Collection

2.1.1 Which Data was Collected ?

All collected raw data were from the publicly available sources mentioned on the assignment's github repository (Neubig, 2024). All the documents listed on the page were used because they broadly covered the topics that would be asked in the question set.

2.1.2 Data Extraction

LTI Faculty The LTI faculty page was scraped using BeautifulSoup to get the names of the faculty members and their research interests. The scraped text was stored in raw fashion with one file per faculty member. These files were transformed into a single JSON file and also into a descriptive text file where each sentence contained exactly one faculty member's information.

LTI Faculty Research Semantic Scholar was used to get the research papers (and abstract, metadata) of the faculty members. The data was parsed into a JSON file which had a list where each member

contained information about one paper.

Academics @ LTI The old LTI website landing page was scraped using BeautifulSoup to get gists of the courses offered by LTI. The scraped text was stored in raw fashion with one file per course, and fed directly to the embedder. The program handbooks were downloaded as pdfs and were then parsed using PyPdf to generate text files. The PDFs were directly used in one set of embeddings while the text files were used in another set of other embeddings.

CMU Courses The CMU course schedule catalog was scraped using BeautifulSoup to get the schedules of all courses. The scraped data was converted into a JSON file and into a combined text file where each sentence contained a worded form of one course's information. The academic calendars for 2022-23 and 2023-24 were downloaded as PDFs and parsed into text files. Both formats were fed into different sets of embedders.

History, Athletics and Events The relevant pages were scraped with BeautifulSoup, and PDFs were downloaded where available. As the documents here had different inherent structures, most of the data was manually copied into text files and parsed with separate scripts to convert the raw data into sentence form.

2.2 Annotations

2.2.1 Type and Quantity of Annotations

A total of 115 test annotation pairs were created manually - 31 for the history and events category, 15 for the LTI program handbook category, 30 for the CMU courses category, 20 for the LTI faculty category, and 19 for the LTI faculty research category. This split covered the different types of data collected, and the different types of questions that could be asked on them. No train annotations were created, since there was no fine tuning or few shot prompting being performed. The annotations

*This work was done as part of the 11-711 course at Carnegie Mellon University. The assignment started off with a group of 3 members, but after 21st February, I had to complete the assignment individually. Permission to work alone was taken from the course instructor. Permission to use material that the group had worked on together, prior to splitting up, was also taken from the course instructor. Shared work has been attributed to contributors.

were used to select the best models and embeddings. All the reference answers were kept down to 5 words with multiple variations being accepted (eg - 11711; 11-711; 11711 course; 11-711 course; 11711 course at CMU; 11-711 course at CMU etc). Reference answers were kept as short as possible with the goal of selecting a model that gave concise answers (to maintain the F1 score), while maintaining recall.

What sort of annotation interface did you use?

How did you estimate the quality of your annotations? (IAA)

2.3

To produce a PDF file, pdf \LaTeX is strongly recommended (over original \LaTeX plus dvips+ps2pdf or dvipdf). Xe \LaTeX also produces PDF files, and is especially suitable for text in non-Latin scripts.

3 Model Details

4 Results

5 Analysis

Acknowledgements

References

Graham Neubig. 2024. [nlp-from-scratch-assignment-spring2024](#). github repository.