Emory Infobot: A LLM-powered Virtual Assistant for University Students

Darren Ni, Aaron Zaiman, Yutong Hu, Jinho D. Choi
Emory University, Emory NLP Lab
Atlanta, GA, USA
{darren.ni, aaron.zaiman, yutong.hu, jinho.choi}@emory.edu

Introduction

Universities often store important student-related information across multiple web domains making it difficult for students to navigate information. This dispersion of information can sometimes create confusion when sources have conflicting information and ultimately result in a poor user-experience. However, the modern era of LLMs are revolutionizing the way we interact with and process information. Conversational AI applications such as ChatGPT are pioneering more natural interactions with digital information, moving beyond traditional search methods. These new conversational agents offer a more intuitive and accessible way to find answers, so we wanted to extend the same motivation to the university student experience. Thus, we introduce Emory Infobot, a conversational QA system tailored for students at Emory University designed to improve the student experience.

Data Collection

Web Crawling

We implement a custom web crawler to scrape textual data from university-related websites. To attain a search tree specific to Emory, we base our search algorithm from ECS, an Emory-specific directory containing a diverse set of web links ranging from Academic Support to Campus Facilities.

- BFS of ECS Link Tree
- Parse HTML → MD
- Filtering Irrelevant Pages

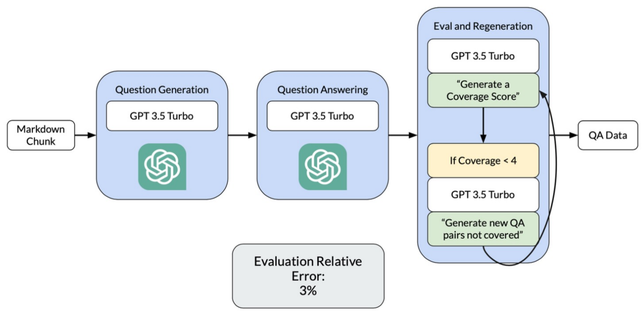
Content from 3.5K+ websites spanning 50+ departments

Synthetic Data Generation and Data Quality Evaluation

We generate a synthetic dataset of question answer pair dialogues between students and a chatbot to replicate components of a real conversation we expect our infobot to answer. To do this, we employ a two-pass data generation pipeline.

- Text Splitting (sliding window)
- Feed markdown chunk GPT-3.5-Turbo with a question generation task
- Pass generated questions with the original content chunk back to GPT-3.5-Turbo with a question answering task

We get: 180K+ generated QA pairs



Data Quality Evaluation

To ensure the quality of these dataset, we con- structed an evaluation pipeline which utilizes GPT as an evaluator. We task GPT to rate how well the generated QA covers the content from which it was derived from on a scale of 1-5. We then cross evaluate the GPT scores against human evaluated scores across 100 chunks. The results yielded a 4.65, indicating GPT as a comparable evaluator to a human. Additionally, we task GPT to regenerate QA pairs if the coverage score falls below 4. Our results yielded a 3% post-evaluation error.

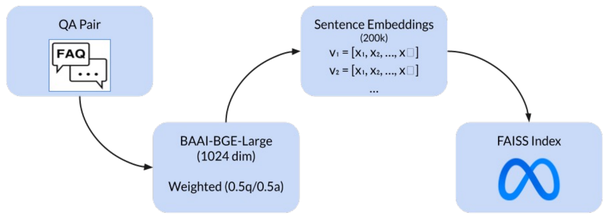
Model

Retrieval Augmented Generation (RAG) Framework

We choose to use a RAG system as a lightweight alternative to fine-tuning a LLM (Gao et al., 2023). Recent works also show that the RAG architecture performs better than parametric-only-S2S models and task-specific retrieve-and-extract architectures (Lewis et al., 2020). We implement our RAG system to perform semantic search over our synthetic dataset and utilize careful prompt templates to generate tailored responses to user queries, giving us a robust, interpretable model.

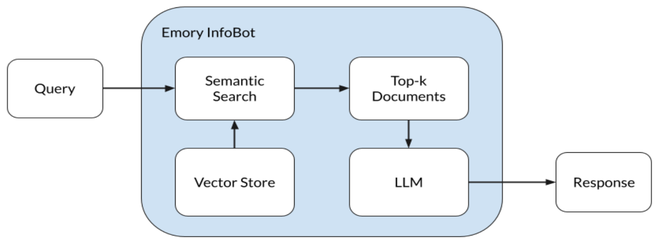
Embeddings

We embed each QA pair in our dataset using the BAAI/bge-large-e} model with a vector normalization step in between. Since user query embeddings could remseble either parts of our dataset questions more than answers or vice versa, we attach equal weights to both the question-and-answer component of each QA pair to create our vector store. These weights were through human evaluation.



Search Engine

We construct a search engine to perform a cosine similarity search between user query and our QA dataset, which is accelerated with FAISS, a GPU-optimized similarity search library. We construct a FlatL2 index with our full QA dataset, which is loaded onto RAM upon instantiation. To determine the optimal number of documents to return to answer each query, we introduce and fine-tune a similarity threshold.



Prompt Templates

We prompt engineer a template designed to summarize retrieved documents prioritizing accuracy of information and minimizing hallucinations. At inference time, retrieved documents are synthesized into the template and inputted to GPT and the output is displayed as the system response.

Safeguards

To ensure the reliability and relevance of the responses generated by our system, we implement two safeguards. First, we compare the cosine similarity score between our system-generated response and each retrieved document to provide a general heuristic for us to determine whether the generated response is sufficiently relevant. Second, we provide the source links of every document used to generate each system response to encourage model interpretability.

Evaluation

Validation Set

To evaluate the Infobot's performance and accuracy, we generate and test on a hybrid dataset of human-generated queries and GPT-generated queries specific to Emory University. The hybrid dataset consists of questions from real students and hypothetical questions generated from GPT, whose topic scope ranges from financial aid to academic policies to athletics

Results

We tasked human annotators to rate the generated responses of the validation set from 1-5. The results yielded an average score of 4.63, indicating high user satisfaction and accuracy of information.

Limitations and Future Work

The current iteration of the Infobot is robust and capable of answering most student queries, but we would like to put more efforts into making the Infobot more conversational. To do this, we plan to utilize a Dialogue State Generation (DSG) approach, where extracted utterance-level context can be used to generate significantly more tailored responses. Further, we plan to add more safeguard mechanisms to address the ethical challenges associated with deploying AI-based assistants (Piñeiro- Martín et al., 2023).

Works Cited

1. Lu, Y., Shen, M., Wang, H., Wang, X., van Rechem, C., & Wei, W. (2023). Machine Learning for Synthetic Data Generation: A Review. arXiv. https://arxiv.org/abs/2302.04062
2. Piñeiro-Martín, A., García-Mateo, C., Docío-Fernández, L., & López-Pérez, M. del C. (2023). Ethical Challenges in the Development of Virtual Assistants Powered by Large Language Models. Electronics, 12(14), 3170. https://doi.org/10.3390/electronics12143170
3. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv. https://arxiv.org/abs/2312.10997
4. 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. arXiv. https://arxiv.org/abs/2005.11401