AcademAI

Q/A Chatbot for Ashoka Academic Policies

Kahaan Shah & Suyog Joshi

Motivation

- We wanted to make doubt solving easier for OAA related doubts
- Wanted to do so in a transparent way
- Wanted to see if LLMs could help improve upon simple vectorization and retrieval algorithms

Data Collection

- We collected two types of documents:
  - Policy documents from AMS
  - Major wise handbooks
- We had handbooks for:
  - Chemistry, IR, CS, English, EVS, SOA, Mathematics, PPE and Visual Arts
  - AU Student Handbook 2023

Tools

- We used Llama API (LLaMA 2) since we don’t have the computing resources to run or fine-tune such a large LLM.
- We vectorised using HuggingFace Embeddings, and created a vector store using Chroma, an open source vector database library.
- We use the *maximum marginal relevance* retrieval algorithm.

Document Splitting and Storage

- Pre-Processing: We split each of the documents into chunks of 1100 characters + 300 characters of overlap between each chunk.
- These chunks were vectorised and stored in a vector database along with metadata about their document of origin and page number within the document.
- The database was used to retrieve documents relevant to the question.

Question-Answering Without Chat History

- The user’s question is passed to the vector database retriever, which gets the most relevant document chunks using the MMR algorithm.
- These chunks along with the question are passed to the LLM, and it is prompted to answer the question taking the context chunks into account.
- There is an issue with this - the LLM does not see the chat history.
  - The user has to be specific when asking follow up questions.

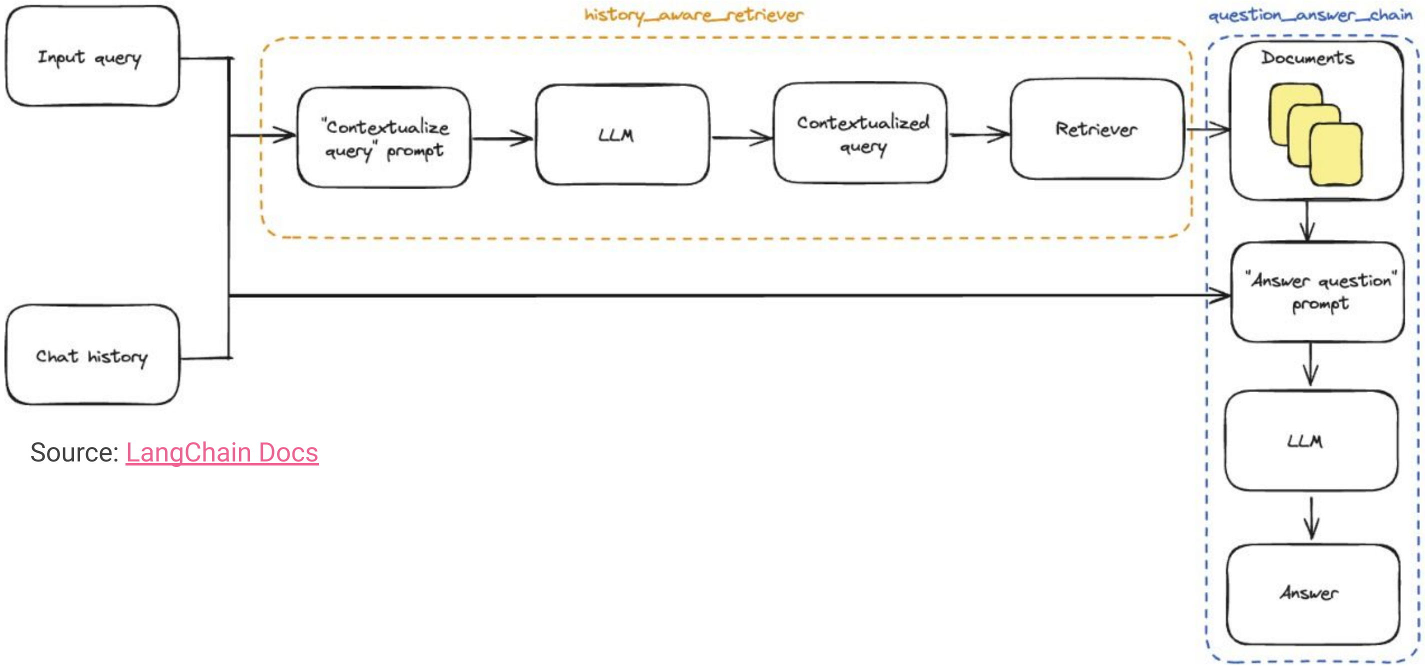
History-aware Retriever

- We wanted to be able to converse with the LLM, rather than just ask it direct questions every time.
  - Need a way for the LLM to get context from previous messages.
- For this, we implemented a history-aware retriever.
- The past conversation history is stored
  - Human questions and their corresponding answers.
- When a new question is asked:
  - The chat history is sent to the API along with the new question.
  - It creates a new standalone question which can be understood without the chat history.
  - This is passed to the retriever which fetches appropriate context docs.

Question-Answer Chain with History

- The context documents from the reformulated question, along with the chat history and the original question is passed to the LLM.
- It is prompted to answer the question given the context and the chat history.
- The answer is presented to the user, and the chat history is updated.
- The top-ranked (similarity) source document name is also presented for reference.

Final Chain Flow



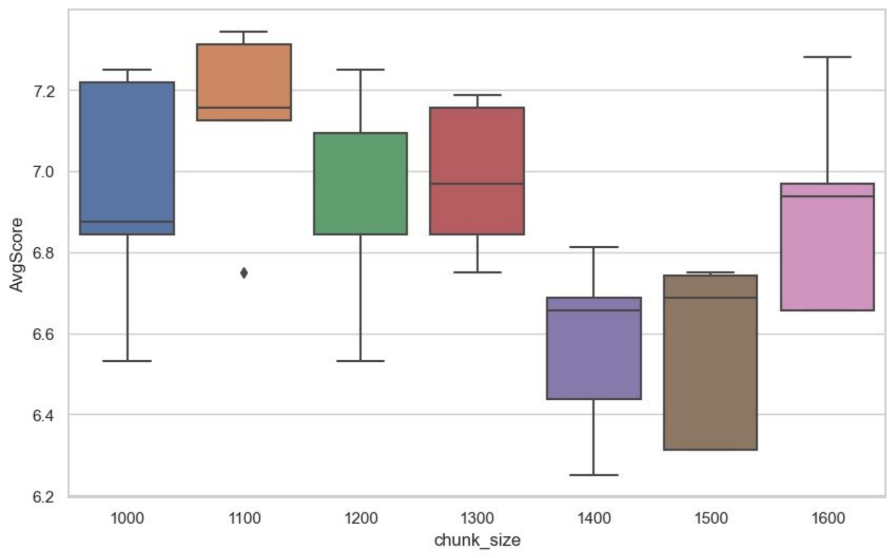
Source: LangChain Docs

Performance on FAQs

- We tested the model on 32 FAQs from the OAA’s FAQ document
- We handpicked the FAQs
- Using wordspace embeddings was not viable since sentences with similar meaning could be embedded far apart
- We asked Llama to rate the responses from 1-10 on the basis of how accurate the answer was compared to the actual answer
- The process was repeated for various document splits.
- This method has many shortcomings, but we wanted to evaluate in some manner given our constraints as well
- Eval took 5 hours with a stable connection

Evaluation Results

- The best performing chunk-size and overlap combination was 1100/300 (avg score 7.36).
- Overall 1100 was the best chunk size and scored 7+ consistently on all but one overlap
- In general, as chunk size increased, eval score decreased moderately (-0.41 correlation coef)
- Correlation coefficient for overlap was negligible (0.017)



Chunk size evals + analyses are in the notebook `bot_eval.ipynb`

Human Evaluation

- We asked a human to evaluate the top two best-scoring combinations of chunk size and overlap

Combination	LLM Eval Score	Human Eval Score
1100/300	7.36	6.23
1100/600	7.31	6.19

Anecdotal Observations

- Successfully looks at relevant parts of documents
- Successfully formulates answers based on context provided
- History-aware chain works and it can answer chat-context based questions
- Can provide the source for an answer
- Is very sensitive to the input prompt (“chem rep” vs “chemistry rep”)
- Often hallucinates or fills in the gaps with external knowledge

Further Work

- Waiting for NEP policies to fall into place
- Trying different retrieval algorithms (we didn’t have enough time)
- Attempting to find contradictions in policies using the pipeline
- Using local model rather than API so we can fine-tune

Learning Outcomes

- How to create a context based LLM chain using LangChain
- How to make the LLM remember history when we have limited access to the API
- Fine tuning prompts to improve performance from an LLM
- Realising what parts of documents are difficult to understand for the LLM
- Understanding the limitations of our pipeline
- Setting up simple UI with history using StreamLit

Try it yourself!