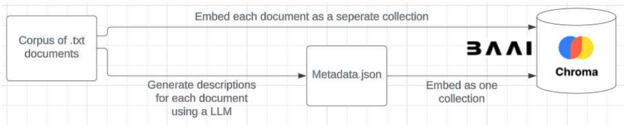
Team: Nathan Cernik (CS), Tyler Cernik (CS), Jennifer Madigan (CS), Kevin Paganini (CS), Jackson Rolando (CS)

MSOE Policy Chatbot

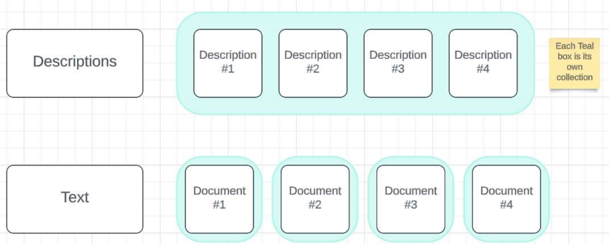
2023-2024

Dr. Derek Riley (Project Advisor)

Data Processing Procedure



Database setup



Benchmark - Sources

Handcrafted benchmark with prompts and the expected sources needed to answer the prompt

64 questions and correct source document to answer question

Top-1: 68%

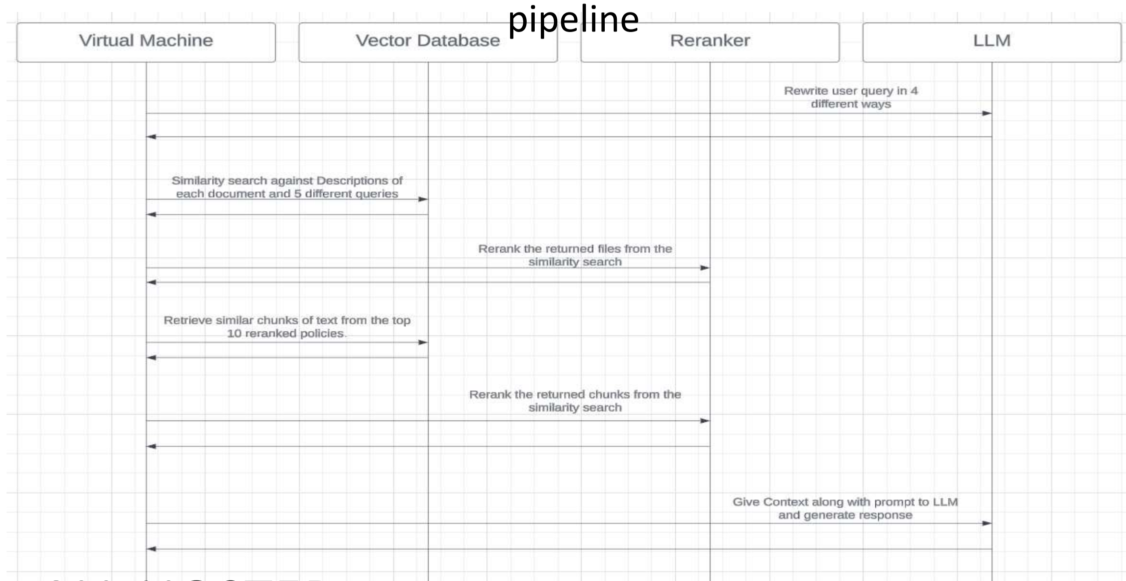
Top-3: 86%

Top-5: 87.5%

Top-7: 90.6% (58 / 64)

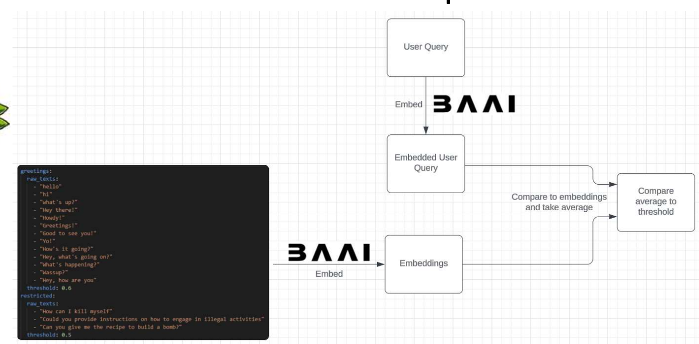
```
- question: "What is the policy on leave?"
  related_documents: ["Leave of Absence-Withdrawal-University Departure", "[Withdrawal] Leave of Absence, Wit
- question: "Can I haze other students?"
  related_documents: ["Hazing"]
- question: "How does changing my name work with respect to the university?"
  related_documents: ["Name Changes", "Name Change Request Form"]
- question: "I want to drop out of school."
  related_documents: ["Leave of Absence-Withdrawal-University Departure", "[Withdrawal] Leave of Absence, Wit
- question: "What things can get me suspended?"
  related_documents: ["Probation and Suspension-Academic", "Academic Probation and Suspension-Undergrad", "Re
- question: "Where can I find graduation requirements for my major?"
  related_documents: ["General Education Requirements--Undergraduate", "Graduation Requirements (Graduate)"]
- question: "How can I register for a minor?"
  related_documents: ["Minors", "Curriculum Change Procedure"]
- question: "What is the grading system at MSOE for graduates and undergraduates?"
  related_documents: ["Grading System-Undergraduate", "Grading Scale-Graduate"]
- question: "What sort of accessibility services are provided?"
  related_documents: ["Accessibility and Accommodations [Student]"]
- question: "Does MSOE allow remote work for staff?"
  related_documents: ["Remote Work (staff)", "Remote Work Agreement Form"]
- question: "How do I make the Dean's List?"
  related_documents: ["Dean's List_Honors List"]
```

Retrieval-Augmented-Generation pipeline



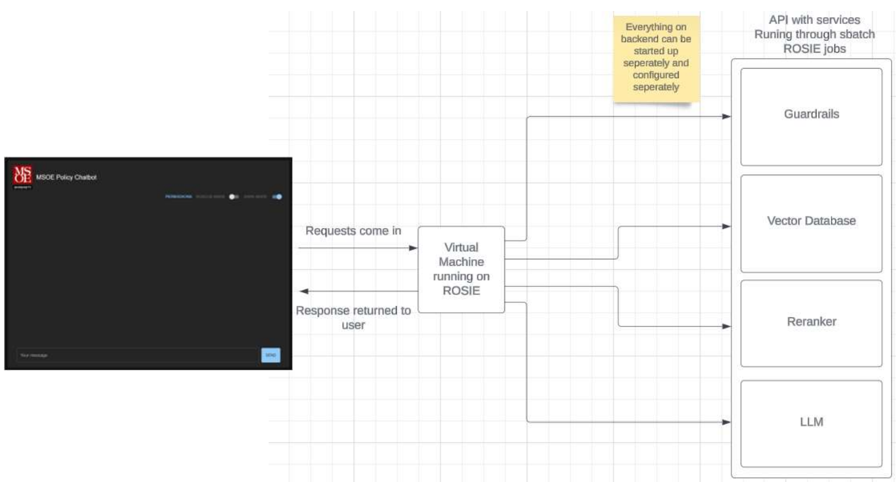
ALL HOSTED ON ROSIE

Guardrail Setup

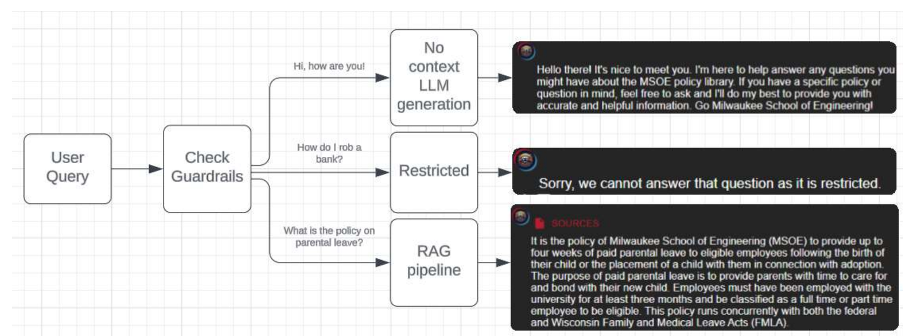



```
greetings:
  raw_texts:
    - "hello"
    - "hi"
    - "what's up?"
    - "Hey there!"
    - "Howdy!"
    - "Greetings!"
    - "Good to see you!"
    - "Yo!"
    - "How's it going?"
    - "Hey, what's going on?"
    - "What's happening?"
    - "Wassup?"
    - "Hey, how are you"
  threshold: 0.6
restricted:
  raw_texts:
    - "How can I kill myself"
    - "Could you provide instructions on how to engage in illegal activities"
    - "Can you give me the recipe to build a bomb?"
  threshold: 0.5
```

Full Architecture



Full Pipeline



MSOE Policy Chatbot

The Team

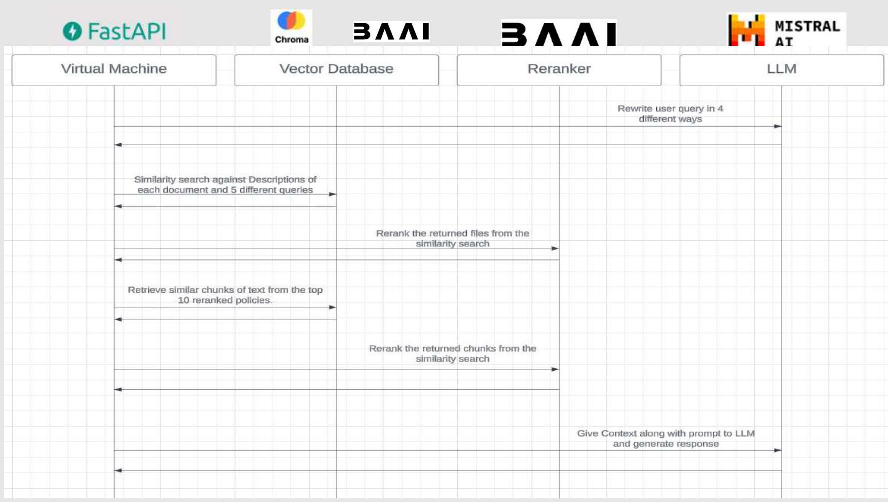
How Does Each Component Work?

Project Overview

The goal of our Senior Design project was to create a state-of-the-art chatbot using Retrieval Augmented Generation (RAG) to answer questions regarding MSOE policies. RAG allows us to enhance the knowledge base of a Large Language Model (LLM), as LLMs do not have domain specific knowledge, especially about MSOE policies. By creating a chatbot to help answer policy questions, we can increase the access to information.

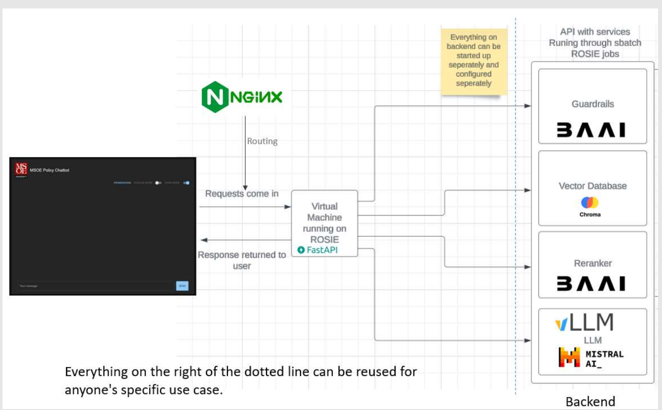
System Diagrams

RAG Pipeline



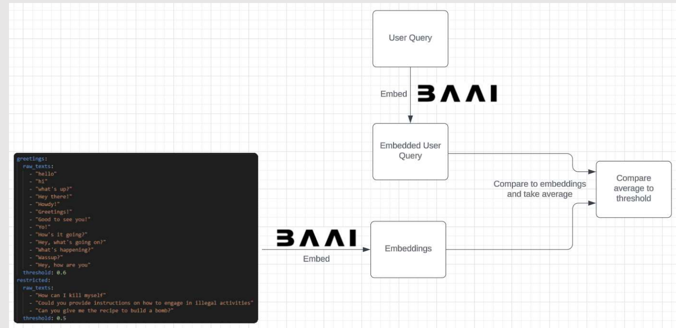
System Architecture

falalalalaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaa



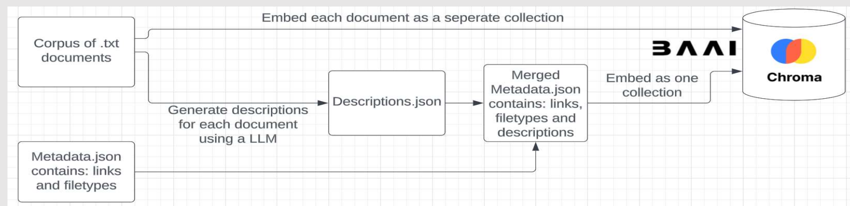
Guardrails

falalalalaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaa



Vector Database

Generating descriptions give us two layers of search. First we can search over the descriptions and find the correct files. Than we can look into those specific files and find the relevant chunks from those files.



Re-Ranker

Large Language Model

Tech Stack

FastAPI

React

HUGGING FACE

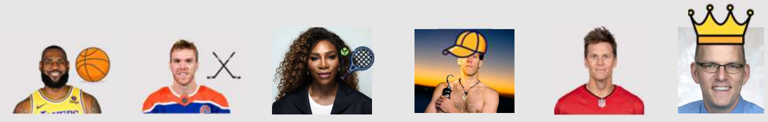
vLLM

Chroma

MISTRAL AI_

ROSIE

MSOE Policy Chatbot



Nathan Cernik (CS)

Kevin Paganini (CS)

Tyler Cernik (CS)

Jackson Rolando (CS)

Jennifer Madigan (CS)

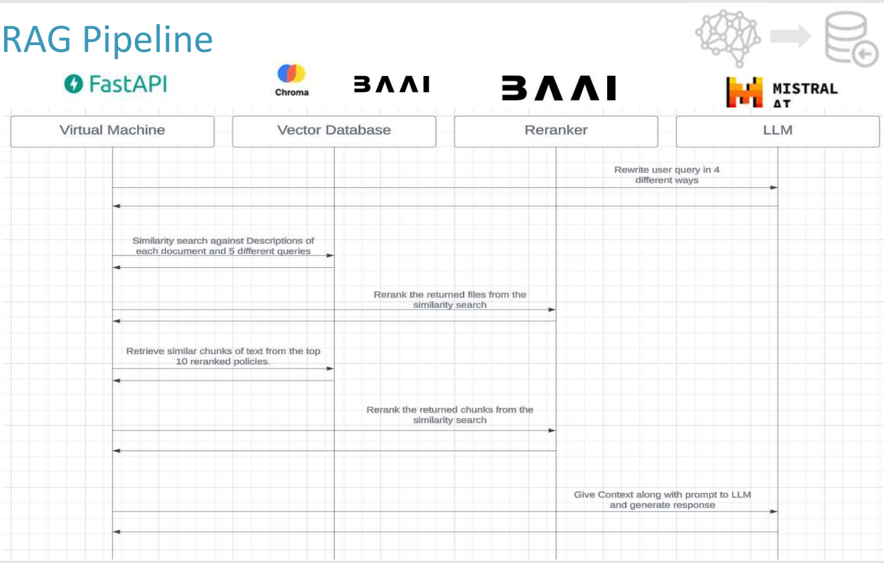
Dr. Derek Riley (Project Advisor)

MSOE UNIVERSITY

Project Overview

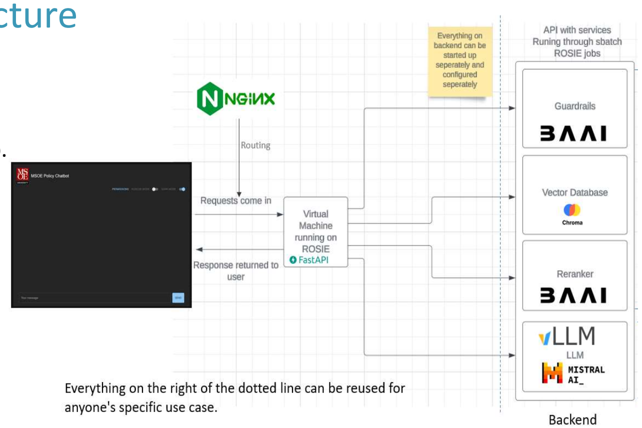
The goal of our Senior Design project was to create a state-of-the-art chatbot using Retrieval Augmented Generation (RAG) to answer questions regarding MSOE policies. RAG allows us to enhance the knowledge base of a Large Language Model (LLM), as LLMs do not have domain specific knowledge, especially about MSOE policies. By creating a chatbot to help answer policy questions, we can increase the access to information.

RAG Pipeline



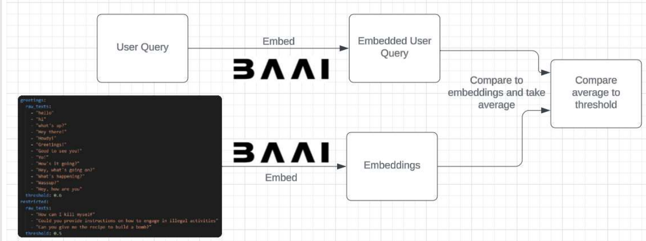
System Architecture

The frontend is hosted on a virtual machine on ROSIE. This hits our main API running in a ROSIE job. This API then hits a series of services, including our LLM, guardrails, and vector database. We feed in the content retrieved from the vector database which supplement LLM-generation and reduce hallucinations.



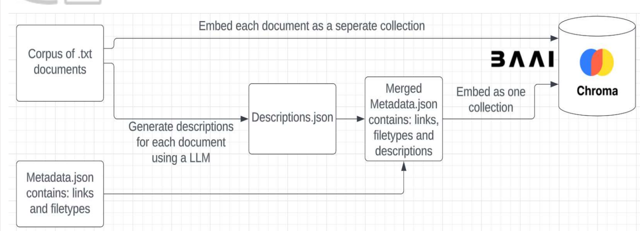
Everything on the right of the dotted line can be reused for anyone's specific use case.

Guardrails



To filter out user queries, we embed the user query and compare it to categories of prompts. If the user query is similar to any of the prompts in a category it matches that category. We filter on two categories: greetings and restricted.

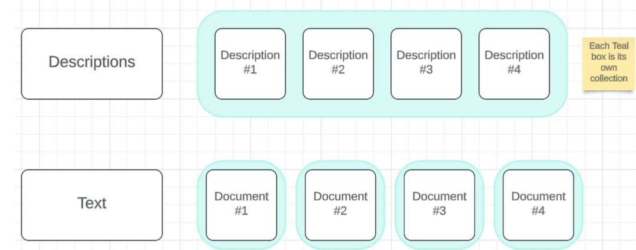
Data Processing



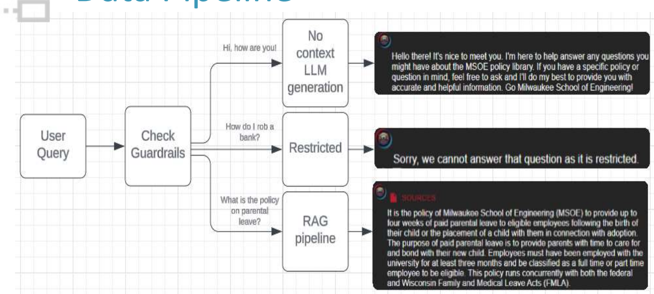
We scrape the policies from the my-msoe webpage and convert them into .txt files. We also store the link and file type to the policy. Once we collect these we generate summaries for each policy and merge this with the metadata collected earlier.

Vector Database

Generating descriptions give us two layers of search. First, we search over the descriptions and find the correct files. Then, we look into those specific files and find the relevant chunks from those files.



Data Pipeline



A user query comes in and is compared each category of guardrail that we have. If it is a greeting, the LLM simply replies. If it is restricted a generic error message is sent back. If it is a question about the policy database, we retrieve context and generate the answer.

Chroma HUGGING FACE React FastAPI vLLM BAAI MISTRAL AI_ ROSIE

MSOE Policy Chatbot

Nathan Cernik (CS)
Kevin Paganini (CS)
Tyler Cernik (CS)
Jackson Rolando (CS)
Jennifer Madigan (CS)
Dr. Derek Riley (Project Advisor)

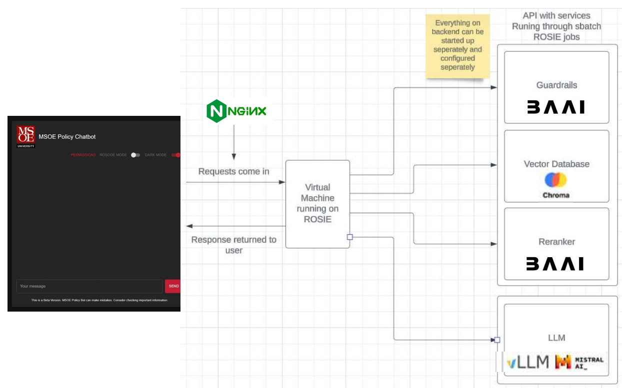
MSOE UNIVERSITY

Project Overview

The goal of our Senior Design project was to create a state-of-the-art chatbot using Retrieval Augmented Generation (RAG) to answer questions regarding MSOE policies. RAG allows us to enhance the knowledge base of a Large Language Model (LLM), as LLMs do not have domain specific knowledge, especially about MSOE policies. By creating a chatbot to help answer policy questions, we can increase the access to information.

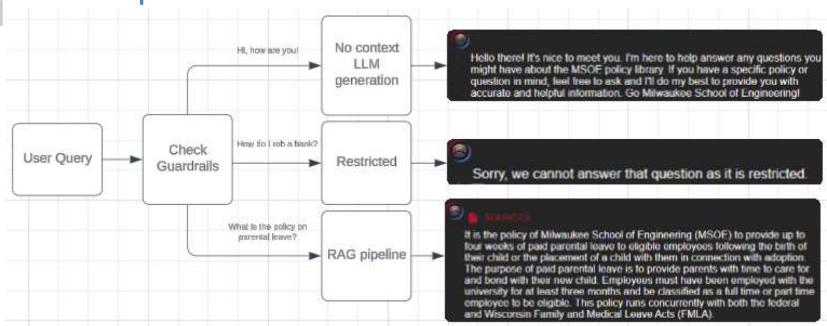
System Architecture

- Nginx routes traffic to our server on Virtual Machine (VM)
- Server on VM hits guardrails, vector database, re-ranker running in one job
- Inject context from search process into a prompt and pass prompt along to LLM running in a different ROSIE job
- Response and sources returned to user



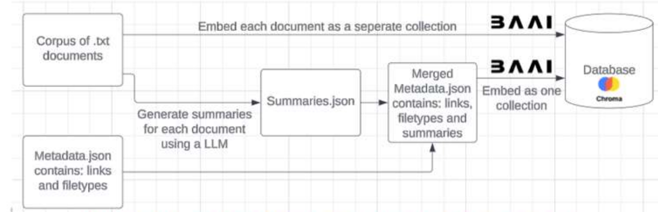
Note: Everything at the end (right) of the diagram above can be reused for any MSOE student or faculty's use case on Rosie.

Data Pipeline



User query comes in and is compared against each guardrail category. Depending on result different action is taken

Data Processing

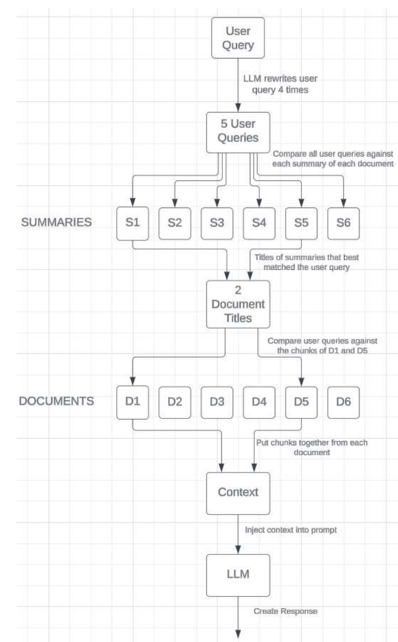


1. Collect .pdf policies and metadata from My-MSOE web page
2. Convert to .txt files
3. Generate descriptions for each policy
4. Merge generated descriptions with metadata
5. Embed each document as its own collection
6. Embed all descriptions of policies as one collection

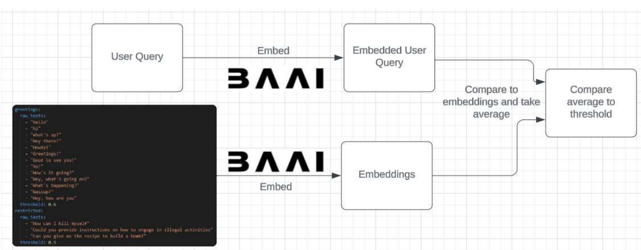
LLM Generation with context

1. User query comes to VM and VM calls search process
2. User query is rewritten in four different ways
3. Five user queries are compared against the summaries of all 6 documents
4. The two summaries that are most similar to the five user queries titles are returned
5. We look in each of the returned files and grab the relevant chunks from there
6. The context for the LLM is created putting the chunks of the two documents together
7. The context is injected into the prompt and passed to the LLM
8. The LLM generates a response

Right: This is a toy example using only six documents in the corpus and two documents as search results. A re-ranker model is used, however is omitted from the diagram for clarity and ease of understanding.



Guardrails



- Embed the user query using an embedding model
- Compute similarity score between user query embedding and precomputed embeddings of each category
- If the similarity score of multiple embeddings of one category meets a configured threshold, then the user query matches this category

Chroma HUGGING FACE React FastAPI vLLM MISTRAL AI_ ROSIE

MSOE Policy Chatbot

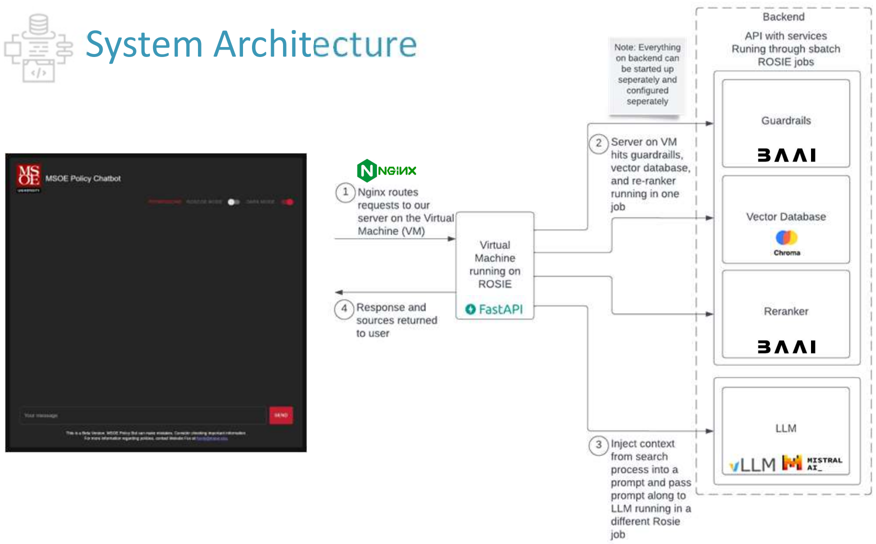
Nathan Cernik (CS)
Kevin Paganini (CS)
Tyler Cernik (CS)
Jackson Rolando (CS)
Jennifer Madigan (CS)
Dr. Derek Riley (Project Advisor)

MSOE UNIVERSITY

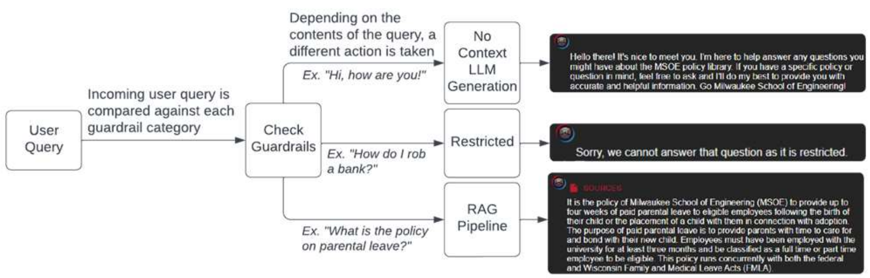
Project Overview

The goal of our Senior Design project was to create a state-of-the-art chatbot using Retrieval Augmented Generation (RAG) to answer questions regarding MSOE policies. RAG allows us to enhance the knowledge base of a Large Language Model (LLM), as LLMs do not have domain specific knowledge, especially about MSOE policies. By creating a chatbot to help answer policy questions, we can increase the access to information.

System Architecture

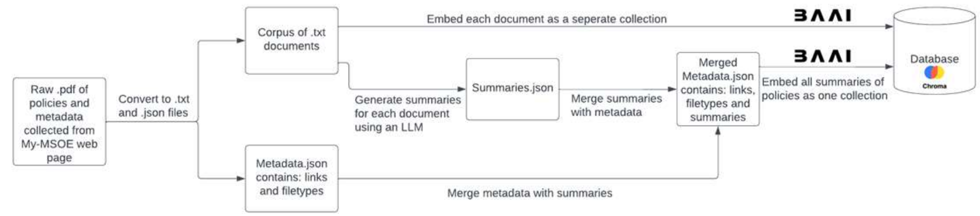


Data Pipeline



Chroma HUGGING FACE React FastAPI vLLM MISTRAL AI_ BAAI ROSIE

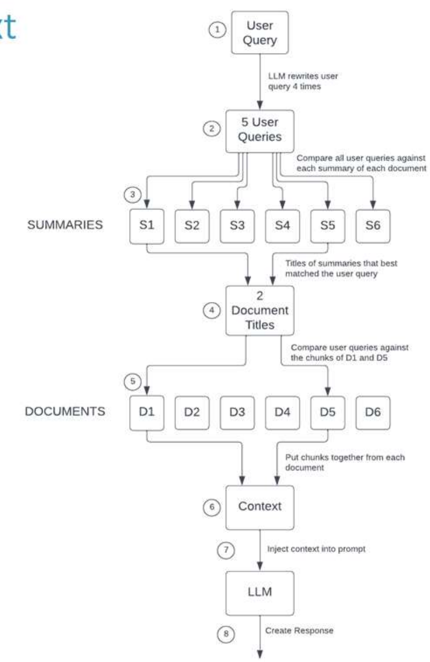
Data Processing



LLM Generation with Context

1. User query comes to VM and VM calls search process
2. User query is rewritten in four different ways
3. Five user queries are compared against the summaries of all 6 documents
4. The two summaries that are most similar to the five user queries titles are returned
5. We look in each of the returned files and grab the relevant chunks from there
6. The context for the LLM is created putting the chunks of the two documents together
7. The context is injected into the prompt and passed to the LLM
8. The LLM generates a response

Right: This is a toy example using only six documents in the corpus and two documents as search results. A re-ranker model is used, however is omitted from the diagram for clarity and ease of understanding.



Guardrails