Project Report: Automated Question Answering System

Introduction

This project is about building an automated question answering system using NLP and machine learning. The system will take in large text datasets, process them, and generate answers to user queries. Main components of the project are data extraction, preprocessing, embedding generation, indexing and query handling. We will be using models from Hugging Face's transformer library and FAISS (Facebook AI Similarity Search) for efficient similarity search and retrieval.

Data Extraction

First step is data extraction. The project uses a collection of milestone papers and lecture notes in text format as the main data sources. The files are loaded from specified directories. The loading process is to traverse through the directories and read the content of all text files. This step makes sure all the relevant documents are available for further processing.

Data Preprocessing

After data is extracted, it goes through preprocessing to prepare it for embedding generation. Preprocessing involves several steps to make the data ready for analysis:

1. **Text Cleaning**: Removing boilerplate content such as HTML markup and other non-textual elements.
2. **Chunking**: Splitting the text into manageable chunks. This is because the transformer models have a limit on the input size they can process at once.
3. **Tokenization**: Converting text into tokens that the model can understand.

These steps help in managing large documents and making sure the text fits within the model's input size constraints.

Embedding Generation

Embeddings are numerical vector representations of text that capture the semantic meaning of the content. For this project we are using the meta-llama/Meta-Llama-3-8B-Instruct model from Hugging Face to generate these embeddings. The model converts text chunks into vectors which can then be used for similarity search and retrieval. This step is important as it converts textual data into a format that can be processed by machine learning algorithms.

Index Building

To enable fast and efficient similarity search we use FAISS to build an index from the generated embeddings. FAISS is a library developed by Facebook AI that is optimized for searching through large sets of vectors. By indexing the embeddings the system can quickly retrieve the most relevant document chunks for a query. This indexing step is important for handling large datasets and making the system responsive.

Query Handling

When a user inputs a query the system first computes the embedding of the query using the same model used for the documents. Then the query embedding is compared against the document embeddings stored in the FAISS index. The index returns the most similar document chunks based on the query embedding. These chunks are the most relevant to the user’s query and are used in the subsequent steps to generate the answer.

Answer Generation

The retrieved document chunks are processed to generate a comprehensive answer. Using the Hugging Face model, we construct a prompt that instructs the model to write a paragraph answering the query using the provided text. This involves:

1. **Preprocessing Prompt:** Creating a prompt that combines the query and the relevant document chunks to guide the model in generating a response.
2. **Analysis Prompt:** Further analyzing the generated answer to ensure it is detailed and contextually appropriate.

The model uses these prompts to generate a coherent and contextually relevant answer to the user's query.

Deployment

The entire system is deployed using Streamlit, a popular framework for creating web applications with Python. Streamlit allows for easy interaction with machine learning models and provides a user-friendly interface for querying and receiving answers. The deployment process involves setting up the Streamlit application to load the model, preprocess the data, handle queries, and display the results.

Conclusion

This project successfully demonstrates the integration of advanced NLP and machine learning techniques to build a robust question-answering system. By leveraging pre-trained models from Hugging Face, efficient text processing techniques, and FAISS for similarity search, the system is capable of providing insightful answers based on large datasets. The deployment with Streamlit makes the system accessible and interactive, providing users with a seamless experience for querying and receiving responses. This project showcases the potential of NLP in transforming how we interact with and derive insights from textual data.