RAG USING TF-IDF AND BART/FLAN-T5

Problem Statement

This assignment focused on developing a Retrieval-Augmented Generation (RAG) system, combining document retrieval and generative models for question answering. Key steps included implementing document retrieval using TF-IDF and generating answers with pre-trained language models.

For answer generation, both base and pre-fine-tuned versions of BART and FLAN-T5 on the SQuAD v2 dataset to enhance precision and contextual relevance. A performance comparison was conducted across all these models.

System Overview

- **Text Corpus and Question Creation**

  The knowledge base for this project was constructed using documents covering five key technologies: Machine Learning, Artificial Intelligence, Social Networking, Web Technology, and Cybersecurity. Each technology is represented by a set of 8-9 questions designed to rigorously evaluate the system's capability to retrieve and generate accurate answers. This comprehensive question set ensures thorough system testing.

- **Data Preprocessing**

  The data was cleaned to remove extra spaces and converted to lowercase for consistency. Afterward, the text was tokenized into chunks of M(set to 3 for our case) sentences each, with an overlap of N(set to 1 for our case) sentences between chunks. Additionally, a corpus and corresponding metadata were created for indexing purposes.

- **Document Retrieval Using TF-IDF**

  A TF-IDF model was created to convert the documents into a vector store, and this vectorizer was also used to embed the queries or questions. A retriever was then built using cosine similarity to identify and return the top K tokens, with each token consisting of multiple sentences.

- **Answer Generation Using Pre-trained Models**

  The system uses the retriever to gather relevant context based on the incoming question. This context, along with the question, is used to create a complete prompt for the generative models. The question and the retrieved context are then input into models like BART or FLAN-T5 to generate a response. This approach allows the model to produce accurate answers that are directly related to the question.

- **Evaluation**

  The test data, which includes questions and their corresponding expected answers, was utilized in this evaluation process. The Exact Match (EM) and F1 Score metrics were applied to compare the generated answers against the expected responses, providing insights into the accuracy and comprehensiveness of the system's performance.

Challenges Encountered

- Model Limitations: Pre-trained models like BART may not perform optimally for specific question-answering tasks since they were not originally trained for this purpose. To improve performance, fine-tuned versions of models on QA datasets, such as Squad v2, were utilized, enhancing the models' ability to provide accurate and contextually relevant answers.
- Computational Efficiency: The reliance on TF-IDF for retrieval, particularly with large datasets, required significant computational resources, impacting the system's real-time efficiency. Exploring more computationally efficient retrieval methods, like dense embeddings, could help mitigate this issue and improve system performance.

Suggestion for improvements:

- Use Better Retrieval Methods: TF-IDF generates sparse embeddings that grow in size with more data, which can be inefficient. Transitioning to dense embeddings from models like Sentence-BERT is more effective for managing larger datasets and improving the accuracy of data retrieval.
- Fine-Tune Models for Specific Tasks: Enhance the precision and relevance of generated answers by fine-tuning models specifically for your tasks. This involves training them further on domain-specific datasets, allowing the models to better understand and respond to the particular nuances of the task.
- Adopt Technologies Like Langchain: Utilize frameworks like Langchain, which offer a standard interface for experimenting with different models. This allows for easy swapping of models and efficient tracking of results, facilitating experimentation and optimization of model performance.