**REPORT\_PART3: LAB2**

**Pair Programming Group 19 - Divya Neelamegam, Harika Boyina**

**Model Architecture of RAG and Experiments**

**Architectural Overview:**

**Sentence Transformers**: Employed for generating embeddings from text chunks. The `sentence-transformers/all-MiniLM-L6-v2` model was specifically used due to its efficiency in creating meaningful sentence-level embeddings.

**Llama Large Language Model (LLM)**: This model was configured with specific parameters to handle the substantial computational load and optimize performance. The use of `bitsandbytes` for quantization significantly reduced memory requirements while maintaining computational accuracy.

**Text Extraction and Processing**: Utilizing `PyPDF2` for reading PDF content and `langchain` for text chunking.

**Embedding Generation and Indexing**: Sentence embeddings were created and indexed using `FAISS` to facilitate efficient similarity searches.

**Query Handling and Response Generation**: The models responded to predefined queries by retrieving relevant text chunks and generating answers using the LLM.

**Experiments and Modifications:**

Several experiments were conducted, focusing on:

**Chunk Size Adjustment:** The text was initially chunked into sizes of 1000 characters with an overlap of 200. Adjustments to this parameter were explored to evaluate the impact on the model's ability to generate correct answers.

**Stopping Criteria**: Custom stopping criteria were implemented to prevent the model from generating overly verbose or irrelevant text. This involved defining specific stop tokens and configuring the generation pipeline to halt when these tokens were produced.

**Design Process and Challenges**

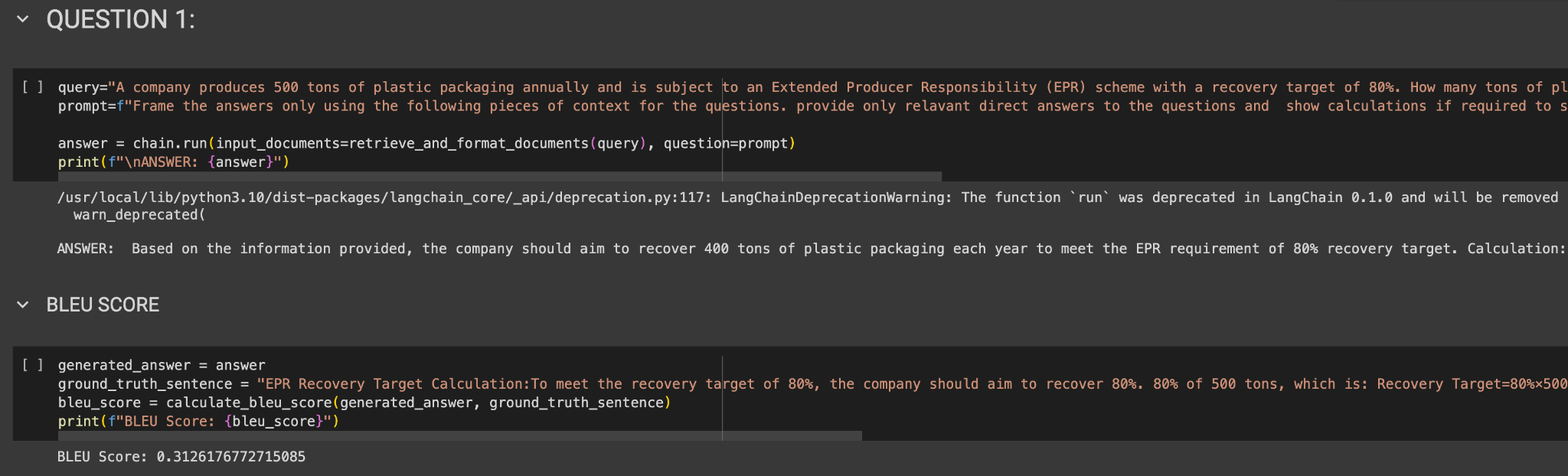
**Response Accuracy**: Particularly when chunk sizes were reduced, the models struggled to generate correct answers. The chunk size is increased to mitigate this.

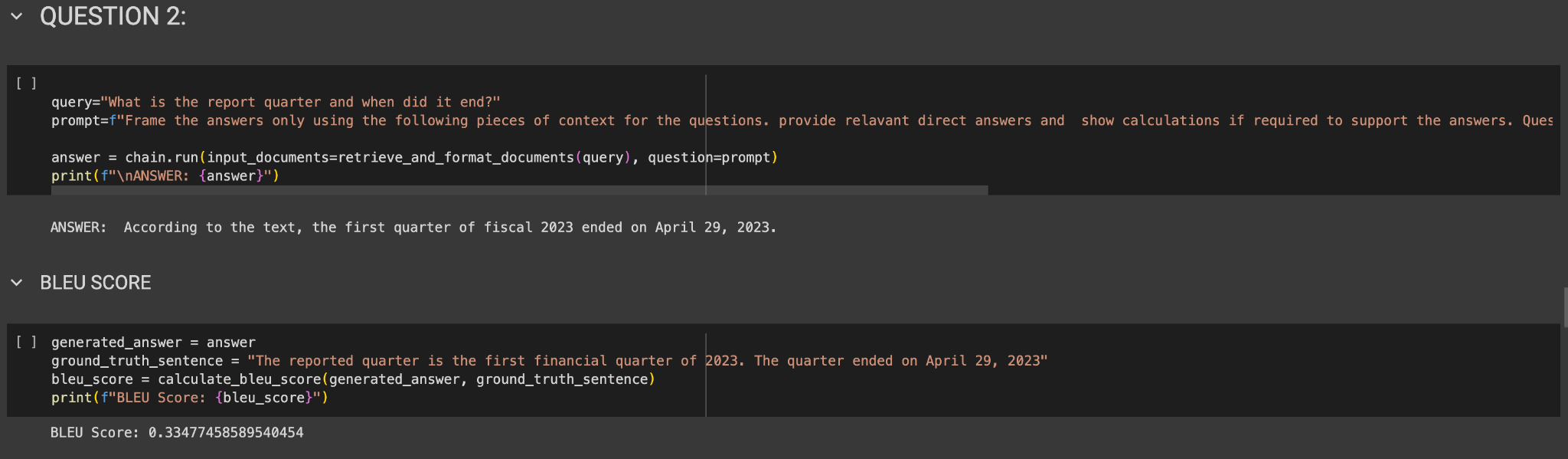
**Parameter Tuning:** Parameters such as chunk size, overlap, and the model's temperature during text generation were meticulously adjusted to strike an optimal balance between performance and resource usage. Additionally, the'return\_full\_text` parameter was fine-tuned and the stopping criteria were enabled to experiment to enhance the contextual accuracy of the generated responses.

**Result and Conclusion:**

The evaluation of the model's performance on two specific questions yielded BLEU scores of 0.3126 and 0.3348, respectively. These scores reflect the model's capacity to generate answers that are syntactically similar to the ground truth answers, although there is room for improvement in terms of semantic accuracy and detail.

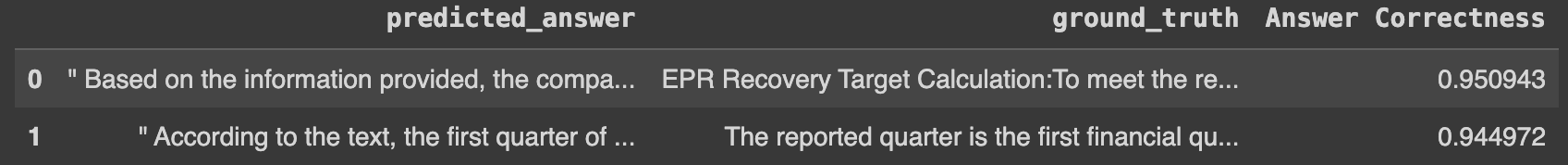
**i) BLEU Score**



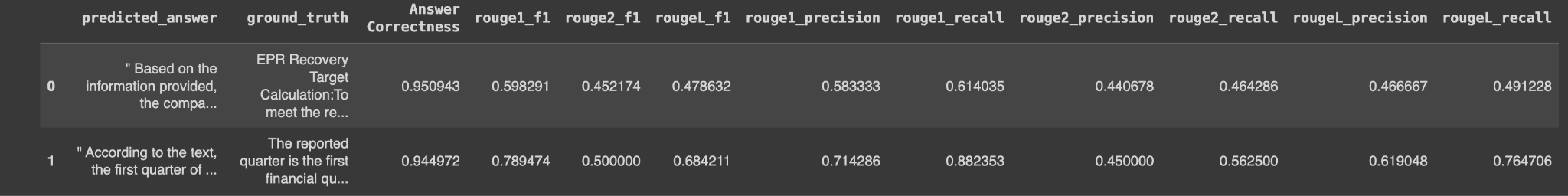


**ii) Answer Correctness:**

The high correctness scores, 0.9509 and 0.944972 respectively, indicate strong alignment between the generated responses and the ground truths, demonstrating the model's effectiveness in capturing and reproducing the factual content from the input data.



**iii) ROGUE Metric**



This capability confirms the utility of the effectiveness of RAG with the LLM model in processing and understanding complex textual information in a targeted manner.

**NOTE:** The generated answers for all the questions are saved as a separate file, **“part3\_RAG\_predicted\_answers.txt”** with each answer within “” and separated by ‘|’ for easy in-class competition evaluation for the remaining questions.