1) Introduction

1.1 Background

In today's digital landscape, users are inundated with vast amounts of information, making it challenging to find precise and relevant answers. Traditional search engines often require users to sift through numerous documents, which can be time-consuming. Question-answering (QA) systems address this issue by providing direct, contextually accurate responses to user queries, streamlining the information retrieval process. The evolution of these systems, driven by advancements in Natural Language Processing (NLP) and deep learning, has significantly enhanced their ability to understand and generate human-like responses.

1.2 Dataset Overview

The Quora Question Answer Dataset is a rich collection of user-generated questions and answers from the Quora platform, encompassing a wide array of topics and query types. This dataset features diverse questions, from straightforward factual inquiries to intricate opinion-based ones.

1.3 Objectives

The primary objectives of this project are to thoroughly explore and preprocess the Quora dataset to ensure its suitability for model training, evaluate and fine-tune various NLP models such as BERT, T5, and GPT to identify the most effective one for the QA task, and assess model performance using metrics like ROUGE, BLEU, and F1-score. Additionally, the project aims to derive actionable insights from the analysis and provide recommendations for enhancing QA systems based on these findings.

1.4 Significance

Developing an advanced QA system has broad implications across various domains, including customer support, where it can reduce the need for human intervention by providing accurate responses; educational tools, where it can serve as an intelligent tutoring system offering precise answers; and information retrieval, where it can enhance the relevance and accuracy of search

engines and virtual assistants. This project seeks to advance QA technology and its practical applications by addressing the complexities of natural language understanding.

1.5 Scope and Limitations

While this project aims to create a state-of-the-art QA model, it is essential to recognize certain limitations, including potential constraints related to the quality and variety of answers in the Quora dataset, the inherent limitations of current NLP models in handling complex or ambiguous queries, and the computational resources required for training advanced models. Despite these challenges, the project will strive to develop a robust and effective QA system, pushing the boundaries of current technology.

2) Literature Survey

2.1 Overview of Question-Answering Systems

Question-answering systems have evolved significantly over the years:

- Early Approaches: These systems used rule-based methods and keyword matching techniques to identify relevant information.
- Modern Approaches: The advent of deep learning brought advancements such as neural networks and embeddings, enabling more nuanced understanding and generation of responses.
- Recent Advances: The introduction of pre-trained models like BERT, T5, and GPT has revolutionized the field, allowing models to perform various NLP tasks with higher accuracy.

2.2 Pre-trained Language Models

2.2.1 BERT (Bidirectional Encoder Representations from Transformers)

- Architecture: BERT employs a transformer-based architecture that reads text bidirectionally, enhancing its understanding of context.

- Applications: Effective in tasks requiring context understanding, such as question-answering and sentiment analysis.

2.2.2 T5 (Text-To-Text Transfer Transformer)

- Architecture: T5 frames every NLP task as a text-to-text problem, which simplifies the process of training models on multiple tasks.
- Applications: Suitable for generating responses and handling complex language tasks.

2.2.3 GPT (Generative Pre-trained Transformer)

- Architecture: GPT uses a transformer-based architecture focused on text generation with unidirectional context.
- Applications: Known for generating coherent and contextually relevant text, making it suitable for conversational agents and QA systems.

2.3 Evaluation Metrics

2.3.1 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

- Purpose: Measures the overlap of n-grams between the generated and reference texts.
- Metrics: Includes ROUGE-N, ROUGE-L, and ROUGE-W.

2.3.2 BLEU (Bilingual Evaluation Understudy Score)

- Purpose: Evaluates the precision of n-grams and assesses the quality of text generation.
- Metrics: Considers precision with a brevity penalty.

2.3.3 F1-Score

- Purpose: Combines precision and recall to provide a balanced measure of model performance.
- Application: Useful for evaluating QA systems where both precision and recall are important.

3) Methodology

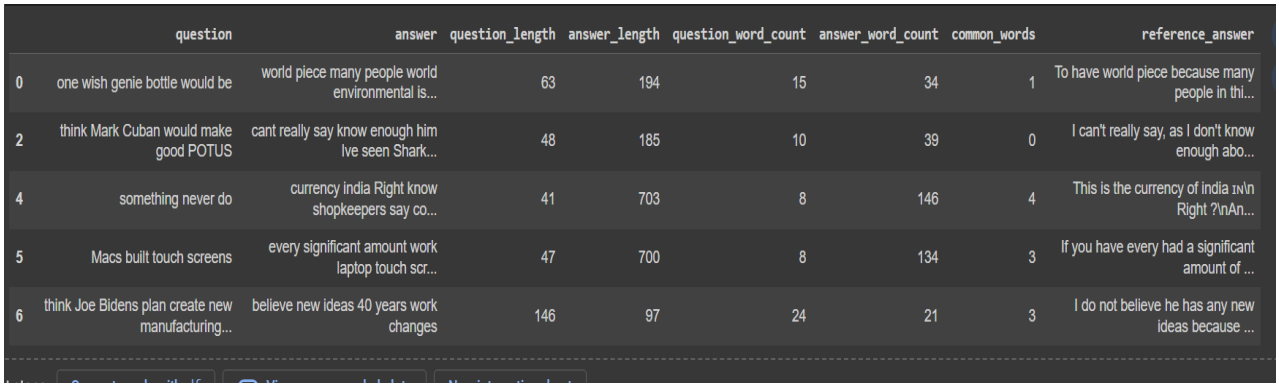
3.1 Data Exploration, Cleaning, and Preprocessing

3.1.1 Data Exploration

- Dataset Overview:
    - DataSet contains 56402 rows of data consisting of question and an answer field, scraped from Quora.
    - It contains 0 Null entries, however duplicate questions can be found.

```
question    0
answer      0
dtype: int64
```

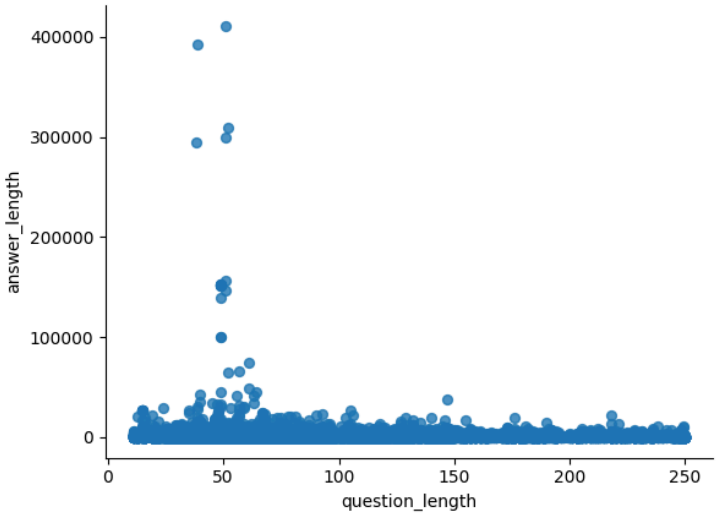
    - Contains heavy usage of stopwords, emoji's and special characters.
    - Extracted features such as length, frequency, common words and word count from the given data for each column.

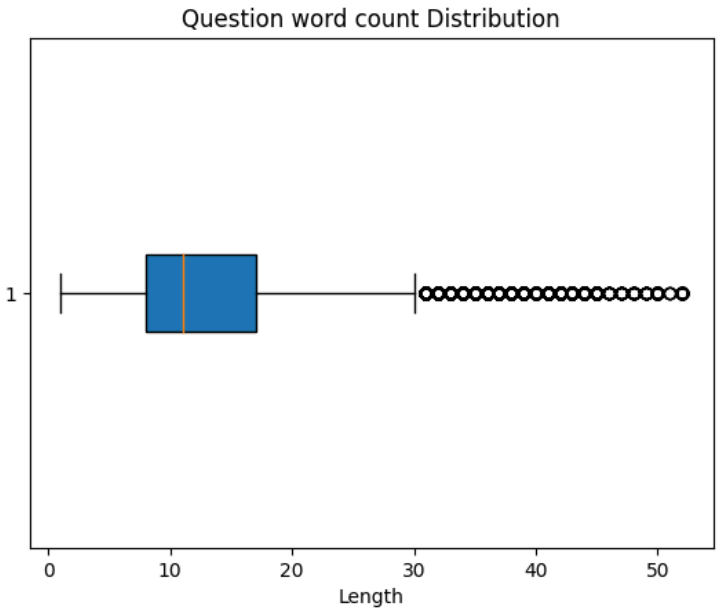


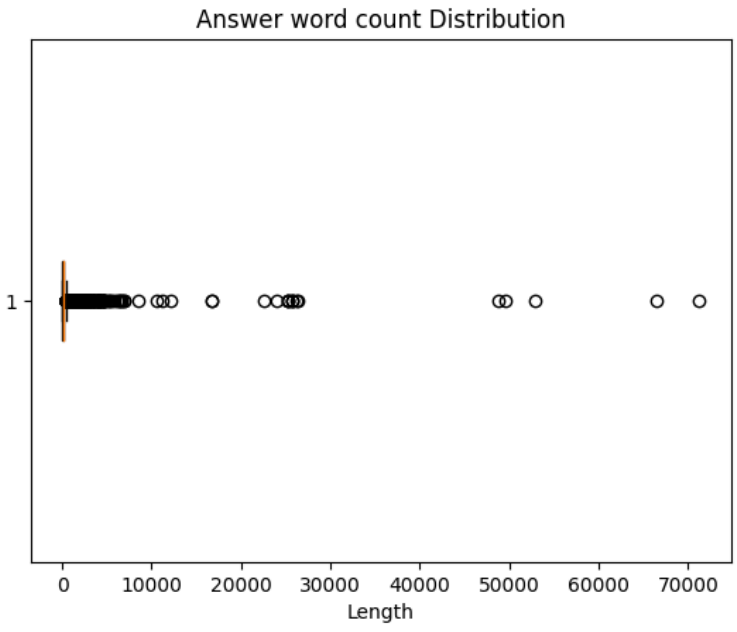
	question	answer	question_length	answer_length	question_word_count	answer_word_count	common_words	reference_answer
0	one wish genie bottle would be	world piece many people world environmental is...	63	194	15	34	1	To have world piece because many people in thi...
2	think Mark Cuban would make good POTUS	cant really say know enough him Ive seen Shark...	48	185	10	39	0	I can't really say, as I don't know enough abo...
4	something never do	currency india Right know shopkeepers say co...	41	703	8	146	4	This is the currency of india ₹\n Right ?\nAn...
5	Macs built touch screens	every significant amount work laptop touch scr...	47	700	8	134	3	If you have every had a significant amount of ...
6	think Joe Bidens plan create new manufacturing...	believe new ideas 40 years work changes	146	97	24	21	3	I do not believe he has any new ideas because ...

- Content Analysis:
    - Question word Count has an average value of 14.112 while answer has an average of 146.969. The average of the dataset is well within Berts capability.
    - The max word count per question in the dataset is 52, while the max answer length is 71232.

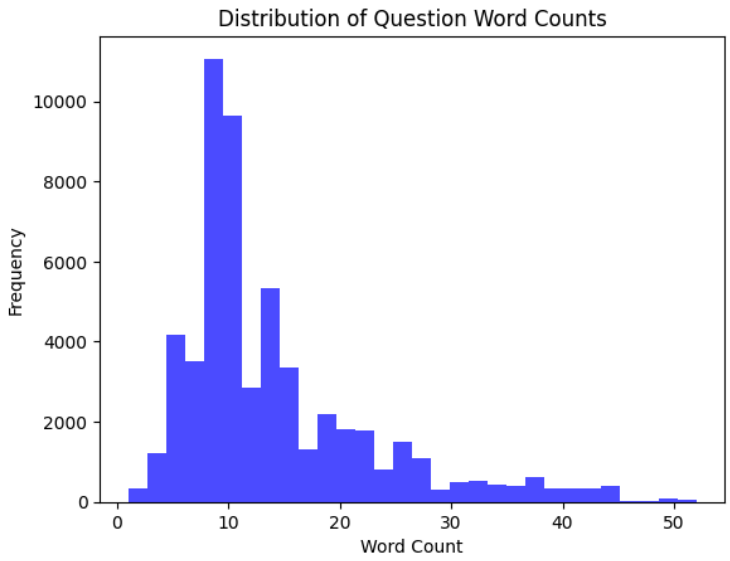
Scatter Plot Comparing Question and Answer Lengths

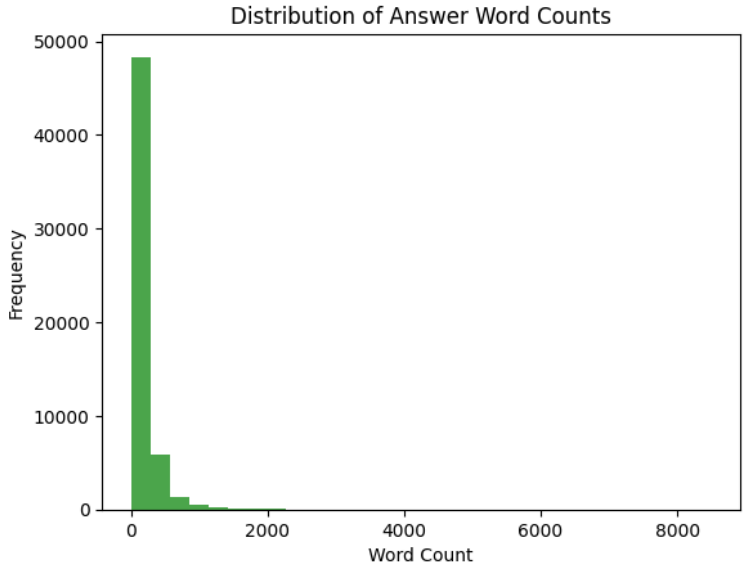




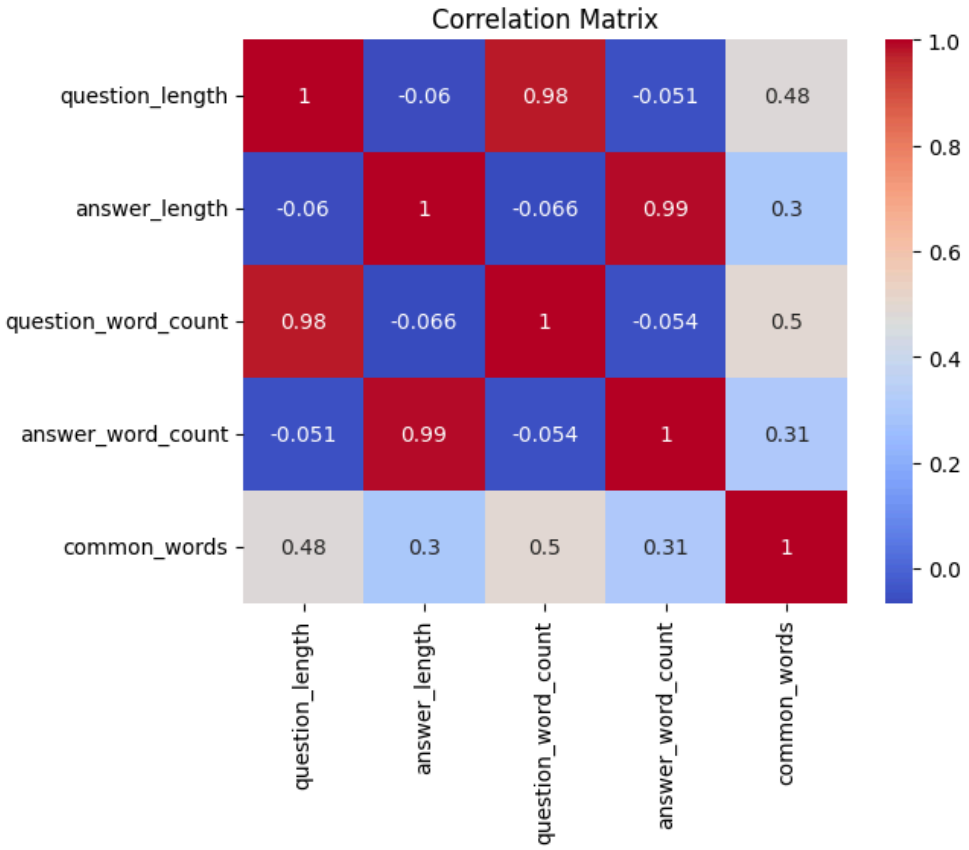


- As seen in the Boxplot for these features as well this is an extreme outlier value and most of the data has an answer word count well below 10,000 words.
- Created a histogram between frequency and word count features for both question and Answer columns. Most of the Questions have a word count around 10 while answers have their peak at around 50.

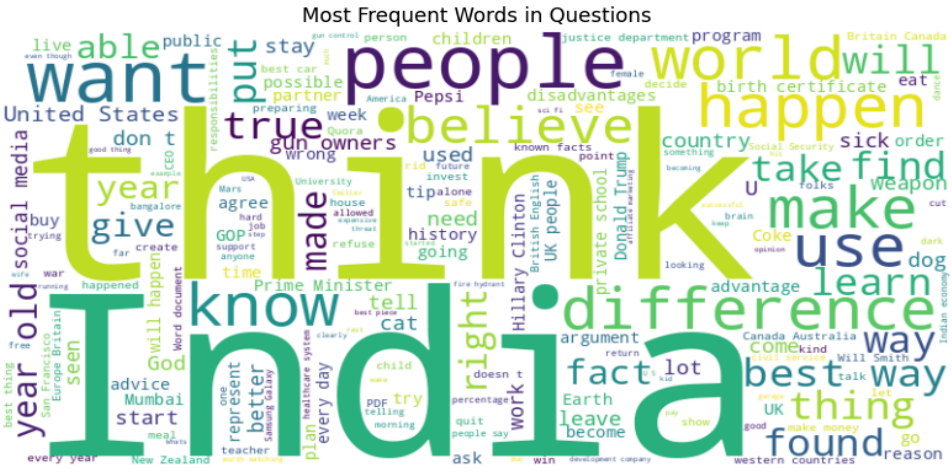




- Created a correlation Heatmap using seaborn between all the extracted features for Question and Answer Column.



- Their question Length and Answer Length seems to be negatively as well as poorly correlated indicating that longer questions need not have longer answers, the same trend is observed for word counts.
- A very High Correlation between word count and length feature as expected.
- Longer questions are somewhat likely to share more words with their corresponding answers.
- The relatively low correlation indicates that the length of the answer has a weaker relationship with the number of common words. This might mean that longer answers do not necessarily contain more words that overlap with the questions.
- Created a wordcloud to visualize the most frequent words in the Questions and Answers column.



3.1.2 Data Cleaning

- Removed all duplicate rows and explicitly converted all row values to string.
- Removed all rows with empty string/ string with only spaces for either question or answer.
- Removed stopwords from both question and answers
- Removed Emojis and special characters
- Removed Outlier answer rows with very huge word counts.

3.1.3 Data Preprocessing

- Tokenized Answer col into sentences and truncated it to 512 Tokens which is the maximal Length of texts allowed in Bert.
- Loaded the SentenceTransformer model with pre-trained GloVe embeddings for generating word embeddings, utilizing the specified device.
- Converted the questions in the DataFrame df into embeddings using the loaded model, showing a progress bar during the process, and ensures the embeddings are of type 'float32'.
- Initialized a FAISS index for efficient similarity search using inner product (dot product) with the dimensionality matching the embeddings.
- Added the knowledge base embeddings to the index for later retrieval and search operations.

3.2 Model Selection

3.2.1 BERT (For Information Extraction)

- I chose BERT over T5 or GPT for a question-answering system because BERT's bidirectional approach captures context from both directions, leading to a deeper understanding of nuanced questions and answers.
- Its architecture has demonstrated exceptional performance on question-answering benchmarks like SQuAD, making it particularly effective for comprehension tasks.
- T5 and GPT are powerful generative models but BERT's focus on understanding rather than generating text provides a strong edge for accurate information extraction in QA scenarios.

3.2.2 GPT-2 (For Natural Language Generation with Answer)

- GPT excels at generating coherent and contextually appropriate text. Its autoregressive nature allows it to produce fluent, human-like responses that are well-suited for transforming BERT's extracted answers into natural-sounding language.
- This combination leverages BERT's precise understanding with GPT's generative capabilities to create clear, engaging, and natural language responses.

3.3 Implementation

3.3.1 Training

- **Model Initialization:** A SentenceTransformer model, specifically using pre-trained GloVe embeddings, is initialized. The model is configured to run on a GPU if available; otherwise, it defaults to the CPU. This model is used to convert textual questions into vector embeddings, which capture the semantic meaning of the text.
- **Create Knowledge Base Embeddings:** The questions from the knowledge base, stored in a DataFrame, are encoded into vector embeddings using the SentenceTransformer model. These embeddings represent the questions in a high-dimensional space where semantically similar questions are located closer together. The embeddings are stored as float 32 for consistency and efficiency.
- **FAISS Index Setup:** A FAISS (Facebook AI Similarity Search) index is initialized to facilitate fast similarity searches. This index is configured to use inner product (dot product) for similarity measurement. The precomputed question embeddings are added to the FAISS index, enabling efficient retrieval of similar vectors.
- **Load and Prepare QA Model:** A pre-trained DistilBERT model fine-tuned on the SQuAD dataset is loaded, along with its tokenizer. A pipeline is created for generating answers based on questions and context.
- **Vector Search:** A function is defined to perform similarity searches on the FAISS index. Given a query vector, the function retrieves the top k most similar vectors, returning both the distances and indices of these vectors.
- **Get Relevant Documents:** For a given question, the system converts the question into an embedding, searches the FAISS index to find the most similar questions in the knowledge base, and retrieves the corresponding documents from the DataFrame. This process identifies the most relevant context for answering the question.
- **Generate Answer for User Question:** A question in string format is input and a generate_answer function is called which uses the Bert model inside pipeline to return the most apt answer, this answer along with the question is then fed to GPT-2 with experimentally derived hyperparameters that converts it to natural language. The answer

is then truncated to the last full stop to avoid unfinished sentences from being in the output.

- **Frontend and Backend:** Created a System using Python and Streamlit to effectively use this QA system with an incorporated feedback in place which updates the training dataset in accordance to feedback received i.e adding more question answer pairs in dataset as per performance eval by a stakeholder.

3.3.2 Evaluation

- **Initialize Metrics:** Sets up ROUGE scorer for evaluating answers and initializes lists for BLEU and F1 scores.
- **Iterate Through DataFrame:** Extracts question and reference answer, retrieves relevant documents, generates an answer using the context.
- **Calculate Evaluation Metrics:** Computes ROUGE, BLEU, and F1 scores to assess the overlap and quality of generated answers.
- **Print Scores:** Outputs the question, reference answer, generated answer, and evaluation metrics.
- **Average Scores:** Calculates and prints the average BLEU and F1 scores.
- **Evaluate the Model:** Runs the evaluate_model function on DataFrame Head.(for faster execution time).

4) Results

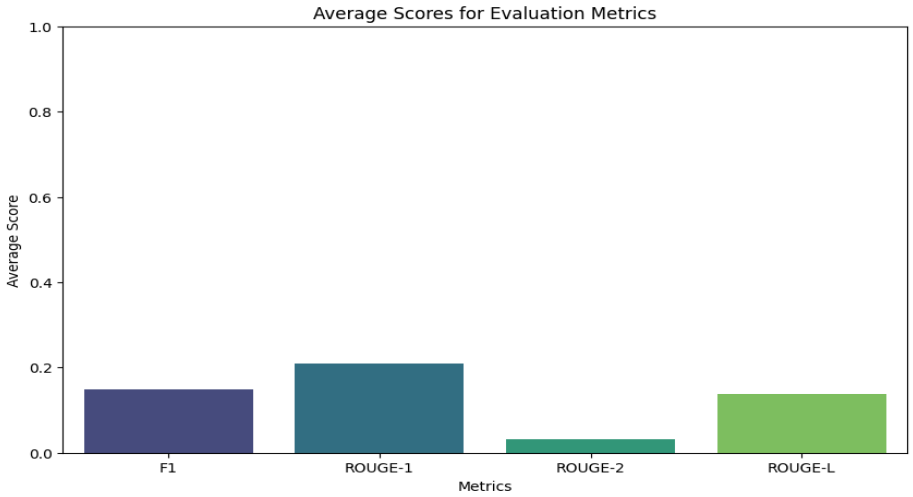
4.1 Model Performance

- **BLEU Score**:
  - **Average BLEU score: 3.63e-79**
  - Indicates very low n-gram overlap with reference answers.
- **F1 Score**:
  - **Average F1 score: 0.149**
  - Shows moderate token overlap, suggesting some relevant word capture.
- **ROUGE Scores**:
  - **Average ROUGE-1 score: 0.210**
  - Shows around 21% unigram overlap, indicating some lexical similarity.
  - **Average ROUGE-2 score: 0.032**
  - Indicates minimal bigram overlap.
  - **Average ROUGE-L score: 0.138**
  - Reflects limited sentence-level structural alignment.

Summary

- The model demonstrates moderate token overlap and some lexical similarity with reference answers.
- There is potential for improvement through enhanced training data, model fine-tuning, and advanced techniques.

4.2 Visualizations



4.3 Link to Github Repository

- GitHub Repo with Instructions to replicate all results as well as a working UI for the same : -

5) Conclusion

5.2 Key Findings

- GPT-2 has significantly better natural language generation while Bert is useful for getting accurate results from the given context.
- Lack of Accurate answers for reference while calculating scores.

5.3 Future Improvements

Data Quality and Quantity:

- We can Ensure that the training data is comprehensive and diverse, covering a wide range of questions and answers to improve the model's understanding and generation capabilities.
- Data Quality and correctness was also not guaranteed since the data was scraped from Quora.

Model Fine-Tuning:

- Fine-tune the model on a more specific dataset that is closely related to the task at hand. This can help the model learn the nuances of the expected answers.(the dataset contained a vague and generalized set of answers).

Adaptable Techniques:

- Consider using more advanced models or architectures, such as transformer-based models with larger parameter sizes, which may capture more context and generate better answers.

- Implement techniques like reinforcement learning with human feedback to iteratively improve the model's performance based on user feedback. (implemented this on the dataset to improve average data quality).