**Task 1: Retrieval (Search) System Evaluation**

**Overview and Approach**

For Task 1, I created a pipeline to evaluate embedding models by using a synthetic dataset and multiple evaluation metrics. I generated a dataset related to Elon Musk’s biography using ChatGPT, which allowed me to curate realistic “ground truth” data for each query. I selected relevant document excerpts from web articles as documents for these queries, creating a small, manageable dataset ideal for evaluating models.

The pipeline employs three embedding models—Universal Sentence Encoder (USE), BERT, and Cohere Embed v3—to convert queries and documents into numerical representations. Among these, Cohere performed notably well, surpassing BERT in similarity scoring, as demonstrated by a bar plot comparing Precision@2, Recall@2, and Mean Reciprocal Rank (MRR) metrics.

**Pipeline Architecture**

**1. Input Data (Queries and Documents)**

* The dataset comprises queries and corresponding documents. Each query has a subset of documents that serve as the relevant ground truth answers. For instance:
  + *Query Text*: "Where was Elon Musk born?"
  + *Relevant Documents*: Documents 1 and 2, each containing information about Musk’s birthplace.

**2. Embedding Models**

* Three embedding models (USE, BERT, and Cohere) were selected to represent the query and document texts as vectors. This representation enables us to compare textual similarity in a vector space. Texts are first tokenized and processed through each model to produce numerical embeddings, capturing semantic meanings of the texts.

**3. Similarity Computation**

* The cosine similarity metric calculates the closeness between each query embedding and its corresponding document embeddings. For each query, this process yields a similarity score per document, allowing us to rank documents based on relevance.

**4. Document Ranking**

* Based on similarity scores, documents are ranked in descending order. This ordering signifies the model’s assessment of document relevance to each query.

**5. Evaluation Metrics**

* *Precision@k*: Measures the fraction of relevant documents within the top-k retrieved documents.
* *Recall@k*: Determines the proportion of relevant documents identified within the top-k results.
* *Mean Reciprocal Rank (MRR)*: Calculates the position of the first relevant document in the ranked list.

Each metric assesses the model’s effectiveness in retrieving relevant documents based on the ground truth.

**6. Results Collection and Comparative Analysis**

* Results from each query-model combination are stored in a DataFrame, tracking Precision@2, Recall@2, and MRR for each model.
* Comparative analysis involves calculating average metric scores per model and visualizing them using a bar chart, which highlights the performance of each model across the evaluation metrics.

**Challenges Faced**

One significant challenge was the lack of an available benchmark dataset tailored for Retrieval-Augmented Generation (RAG) models, which are designed for document retrieval tasks. I created a synthetic dataset with ChatGPT to establish a ground truth, but a rigorously curated dataset would be essential for broader use cases.

**Solutions and Future Improvements**

Creating the dataset enabled me to evaluate models effectively within the pipeline, though future iterations could benefit from datasets created specifically for RAG model evaluation. Incorporating additional evaluation metrics and exploring other embedding models (e.g., Jina Embeddings). Additionally, gathering larger datasets could enhance the generalizability and accuracy of the evaluation results.

**Conclusion**

The embedding evaluation pipeline successfully compared the performance of USE, BERT, and Cohere embeddings for document retrieval tasks. **BERT** emerged as the best model for ensuring that a relevant document appears as the top result, with an MRR score of 1.0. However, **Cohere** demonstrated the most reliable performance in terms of both precision and recall, making it the best model overall for consistently retrieving relevant documents. Meanwhile, **USE** underperformed compared to the other two models across all metrics, indicating that it may be less suitable for this retrieval task. These insights highlight the utility of a structured, metrics-based evaluation framework in identifying the strengths and weaknesses of each model for specific retrieval objectives.

**Task 2 - Text Generation and Evaluation System**

**Objective**

The goal of this task is to evaluate the effectiveness of three different language models—GPT-2, BART, and T5—in generating accurate responses to queries based on retrieved documents. Generated texts are compared against reference texts using the ROUGE, BLEU, and BERTScore metrics.

**Methodology**

1. **Data Preparation**:
   * Three queries related to Elon Musk were selected.
   * Two documents per query, containing relevant information, were used as input for the generation models.
   * Reference texts were manually created to serve as a standard for comparison.
2. **Model Execution**:
   * For each query, the top two documents were concatenated and provided as input prompts to GPT-2, BART, and T5 models.
   * Each model generated a text response based on the input prompts.
3. **Evaluation Metrics**:
   * **ROUGE**: Measures overlap of n-grams between generated text and reference text.
   * **BLEU**: Assesses precision of n-gram overlap in the generated text.
   * **BERTScore**: Evaluates the similarity between the generated and reference texts using contextual embeddings.
4. **Challenges**:
   * Limited time for task completion due to academic commitments.
   * Had to manually construct a dataset and reference texts due to a lack of pre-existing resources.
5. **Future Work**:
   * Expand the evaluation with additional models, such as Ollama's Mixtral, LLama, and others, to compare performance across a broader set of models.

**Results Summary**

The table below displays the evaluation metrics for each model (GPT-2, BART, T5) across the three queries. The metrics show ROUGE-1, ROUGE-L, BLEU, and BERTScore for each generated response:

| **Query** | **Model** | **ROUGE-1** | **ROUGE-L** | **BLEU** | **BERTScore** |
| --- | --- | --- | --- | --- | --- |
| Where was Elon Musk born? | BART | 0.425532 | 0.425532 | 0.244 | 0.919492 |
|  | GPT-2 | 0.232558 | 0.232558 | 0.099 | 0.885008 |
|  | T5 | 0.454545 | 0.454545 | 0.171 | 0.927223 |
| What was Musk's childhood like? | BART | 0.264151 | 0.188679 | 0.000 | 0.913298 |
|  | GPT-2 | 0.239130 | 0.152174 | 0.000 | 0.892665 |
|  | T5 | 0.333333 | 0.222222 | 0.000 | 0.922462 |
| How did Musk get interested in tech? | BART | 0.571429 | 0.321429 | 0.137 | 0.906031 |
|  | GPT-2 | 0.344086 | 0.301075 | 0.089 | 0.885932 |
|  | T5 | 0.530612 | 0.448980 | 0.157 | 0.894565 |

**Conclusion**

The results indicate that T5 performed best overall across the evaluation metrics, showing a consistent edge in ROUGE-1, ROUGE-L, and BERTScore metrics, suggesting a higher relevance and fluency in generated text responses. However, BLEU scores remained low for all models, highlighting potential limitations in n-gram precision.