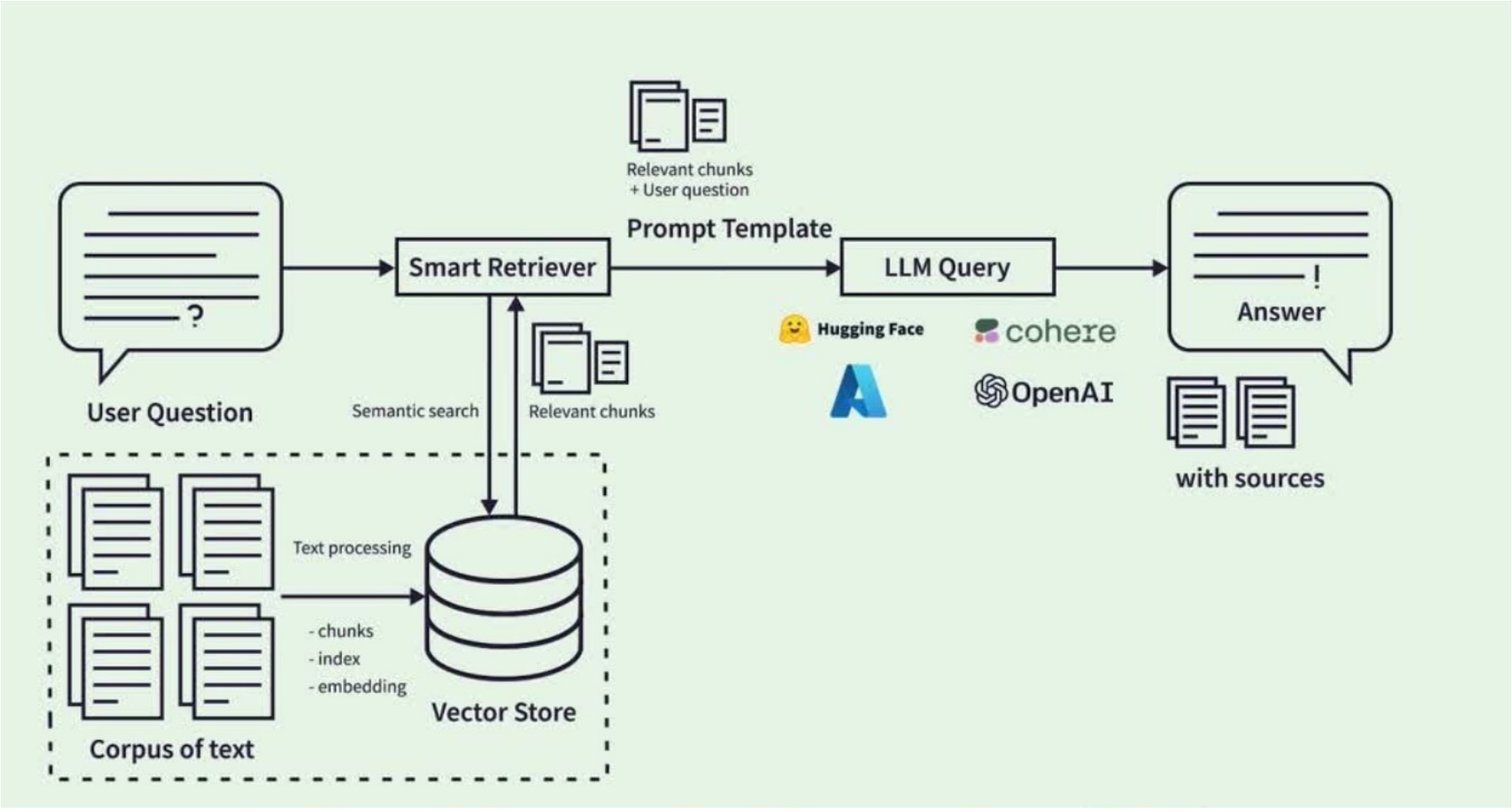
QUESTION ANSWERING CHATBOT USING RAG

SMAI Project

BY:
RHITESH KUMAR SINGH, 2023202013

Goal: To build an abstractive QA chatbot using RAG.

Architecture Diagram



RAG Pipeline

I. Indexing: ingesting data from a source and indexing it.

1. **Document Loaders**: Load different types of documents like text file, pdf file etc.

2. **Text Splitters**: Breaks large documents into smaller chunks to fit in a model's finite context window.

3. **Text Embedding Models**: Converts chunks into embeddings using different methods like cohere, huggingface and OpenAI embeddings.

4. **Vector Stores**: To store and index the split embeddings, so they can be later searched over. Used Faiss and Chromadb.

RAG Pipeline

II. Retrieval and generation:

1. **Retrievers**: Given a user input, relevant splits are retrieved from storage using a retriever. It accepts a string query and returns a list of documents as output.

Examples:

- Vector store, Contextual Compression retriever, Multi-query retriever, Parent Document retriever, Ensemble retrievers.
- RAGatouille (uses ColBERT), Cohere and Cross Encoder rerankers.

2. **Generation**: A RAG chain produces an answer using a prompt that includes the answer and the retrieved data.

Basic RAG Pipeline

- Input document: Bert research paper, 16pgs
- Embedding model: miniLM (open source)
- Vector store: FAISS
- LLM: OpenAI gpt-3.5-turbo, temperature = 0
- Retriever: vector store backed retriever – Faiss
- Created a prompt and a rag chain and generated results.

```
from langchain.prompts import ChatPromptTemplate
template = """You are an assistant for question-answering tasks.
Use the following pieces of retrieved context to answer the question.
If you don't know the answer, just say that you don't know.
Use three sentences maximum and keep the answer concise.
Question: {question}
Context: {context}
Answer:
"""
prompt = ChatPromptTemplate.from_template(template)
print(prompt)
```

```
query = "What was the best model performance achieved?"
rag_chain.invoke(query)
```

```
'The best model performance achieved was by the BERT based model with additional layers for SQuAD, which had an F
1 score of 73.77 on the development set and 72.95 on the test set. This model outperformed other models tested, w
ith the median size model QANet coming in second. The performance of the models was determined by factors such as
model size, depth, and training time.'
```

Datasets for Evaluation

1. **To evaluate embeddings**: STS benchmark dataset, the Semantic Textual Similarity dataset. It contains pair of sentences and a human annotated score that gave the similarity between them.

2. **To evaluate the retrievers and their respective RAG pipelines**: used a subset of dataset from the paper, “ ARAGOG: Advanced RAG Output Grading”, April 2024. From the original set of 107 samples, only 8 samples were used for evaluating. This was the gold dataset, which had questions and ground truths that was generated using GPT-4 in the original paper. Only one input document was used, owing to limited compute.

Embedding Models

- OpenAI Embeddings, default ada model, 1536 dim
- Cohere Embeddings, embed-english-light-v3.0, 384 dim
- HuggingFace Embeddings, BAAI/bge-base-en-v1.5, 768 dim
- Sentence Transformers MiniLM Embeddings, all-MiniLM-L6-v2, 384 dim
- Matroyshka Embeddings, tomaarsen/mpnet-base-nli-matryoshka, 768 dim

Embedding Models Evaluation

	sentence1	sentence2	score	cos_sim_MiniLM	MiniLM_Score	cos_sim_Matryoshka	Matryoshka_Score
0	A train is moving.	A man is doing yoga.	0.0	0.072262	0.361308	0.132224	0.661121
1	A group of men play soccer on the beach.	A group of boys are playing soccer on the beach.	3.6	0.788625	3.943125	0.826425	4.132123
2	One woman is measuring another woman's ankle.	A woman measures another woman's ankle.	5.0	0.946494	4.732468	0.971200	4.856000

- 5 pairs of sentences were selected from the STS dataset, and cosine similarity was computed. It was transformed to [0,5] range in accordance to the STS dataset. Similarly scores were calculated for all other embedding models.
- **Evaluation metrics:**
- Pearson correlation measures the linear correlation between two variables.
- Spearman correlation measures the strength and direction of association between the ranked values of two variables. It assesses how well the relationship between two variables can be described using a monotonic function.
- Mean Squared Error is a measure of the average squared difference between the actual values and the predicted values.

Embedding models Evaluation Results

	Model	Pearson Correlation	Spearman Correlation	Mean Squared Error
0	MiniLM	0.992813	1.0	0.072809
1	Matryoshka	0.988837	1.0	0.187659
2	Cohere	0.991153	1.0	0.146696
3	HF_BAAI	0.992190	1.0	1.547897
4	OpenAI	0.987199	1.0	4.487634

- All models exhibit strong correlations with the human-annotated scores, suggesting that they capture semantic similarity effectively.
- The high Spearman correlation coefficients indicate that the models rank the sentence pairs in the same order as humans, demonstrating robustness in capturing relative similarities.
- MiniLM performs the best overall, showing the highest Pearson correlation and the lowest MSE.
- Cohere and HF_BAAI also perform well, with high correlation coefficients and relatively low MSE values.
- OpenAI lags behind the other models, with the highest MSE, indicating less precise prediction of similarity scores.

Embedding models Evaluation using Ragas

```
results #openai, cohere, miniLM
```

```
[{'context_precision': 1.0000, 'context_recall': 1.0000, 'context_relevancy': 0.0198},
 {'context_precision': 1.0000, 'context_recall': 1.0000, 'context_relevancy': 0.0196},
 {'context_precision': 1.0000, 'context_recall': 1.0000, 'context_relevancy': 0.0158}]
```

- **Context Precision and Recall:** All three models, OpenAI, Cohere, and miniLM, have achieved perfect scores of 1.0000 in both context precision and recall.
- This indicates that they are exceptionally effective in selecting highly relevant context for the questions and are also successful in capturing all necessary information from the context that appears in the ground truth answers.
- **Relevancy:** The slightly higher relevancy scores for OpenAI and Cohere compared to miniLM could imply a marginal advantage in handling broader or more complex contextual relevancies.

Advanced RAG

1. Vector store-backed retriever: Basic retriever, works on MMR.

2. Parent document retriever: Improving a retriever by embedding our documents into small chunks, and then retrieve a significant amount of additional context that "surrounds" the found context.

3. Prompt Tuning - Multi Query retriever**:** It automates the process of prompt tuning by using an LLM to generate multiple queries from different perspectives for a given user input query.

4. Ensemble BM25 retriever: Given a user query, it retrieves documents from the dense vector retrieval and the sparse BM25 vector retrieval and using Reciprocal Rank Fusion, fuses and weights them into a single retrieved list.

5. Contextual Compression: Instead of immediately returning retrieved documents as-is, we compress them using the context of the given query, so that only the relevant information is returned.

Advanced RAG

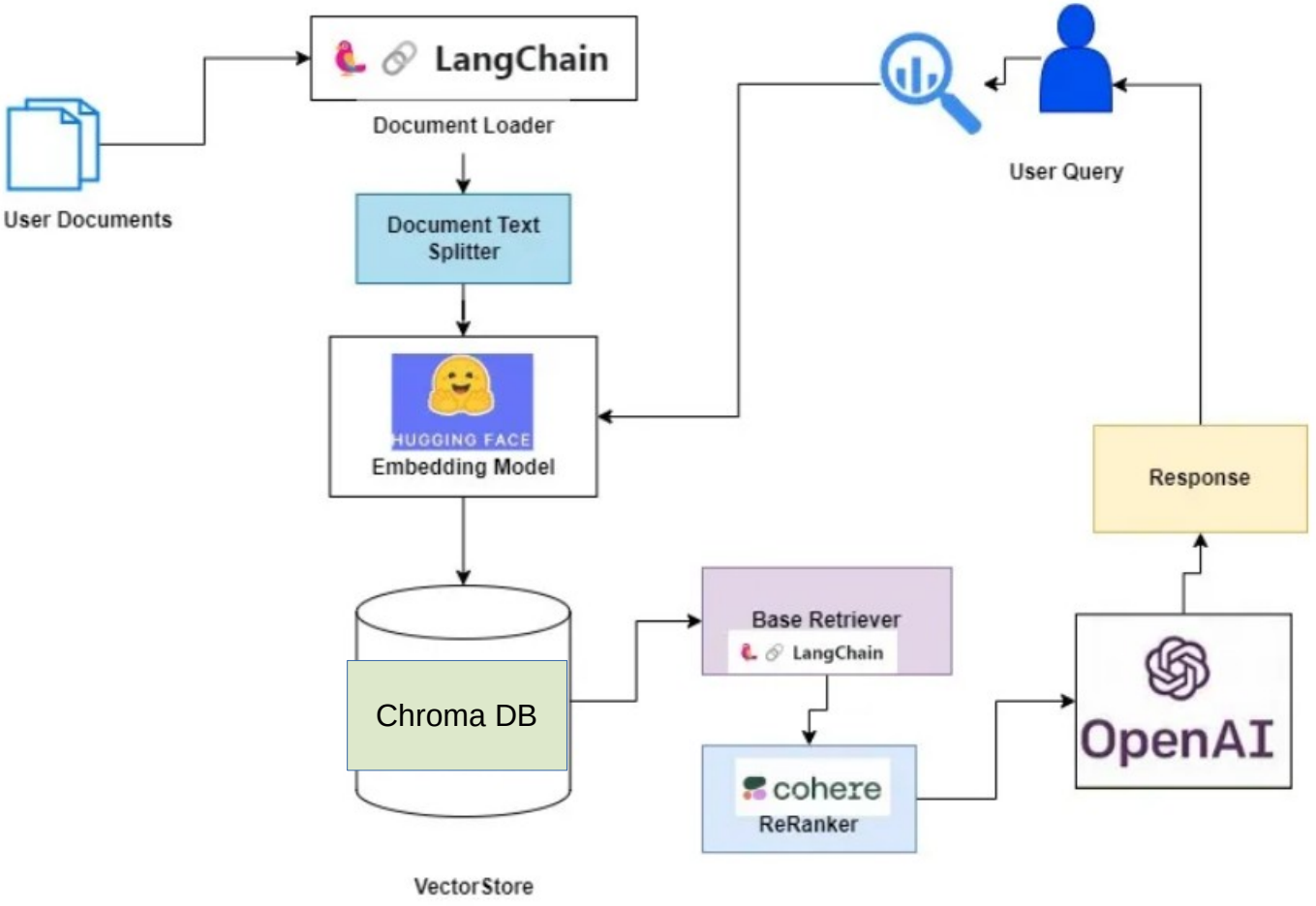
6. ColBERT retriever: ColBERT is a fast and accurate retrieval model, enabling scalable BERT-based search over large text collections in tens of milliseconds.

7. Cohere reranker: After the base retriever retrieves the top 20 relevant documents, there might be lots of irrelevant texts retrieved too so the reranker ranks it by keeping the most relevant documents at the top.

8. Multi Query + Cohere reranker: Kept Multi Query retriever as base retriever and perfomed reranking using cohere reranker.

9. Cross Encoder reranker: the retrieval system might retrieve documents that are not that relevant for the search query. Hence, we use a re-ranker based on a cross-encoder that scores the relevancy of all candidates for the given search query.

Reranker



Source: https://medium.aiplanet.com/advanced-rag-cohere-re-ranker-99acc941601c

Retrievers Evaluation Results

	name	context_precision	faithfulness	answer_relevancy	context_recall	context_relevancy	answer_correctness	answer_similarity
0	Chroma	0.9201	0.906200	0.813800	0.875000	0.031300	0.705200	0.895600
1	Parent Document	0.9167	1.000000	0.916900	0.901800	0.022600	0.760000	0.882000
2	Ensemble BM25	0.8807	1.000000	0.920300	0.875000	0.066000	0.683300	0.866000
3	MultiQuery	0.9165	0.943700	0.944500	0.750000	0.036800	0.698400	0.886300
4	Context Compressionr	1.0000	0.933333	0.929567	0.833333	0.030646	0.656128	0.891121
5	Cross Encoder Reranker	1.0000	1.000000	0.938041	0.833333	0.030646	0.702988	0.888144

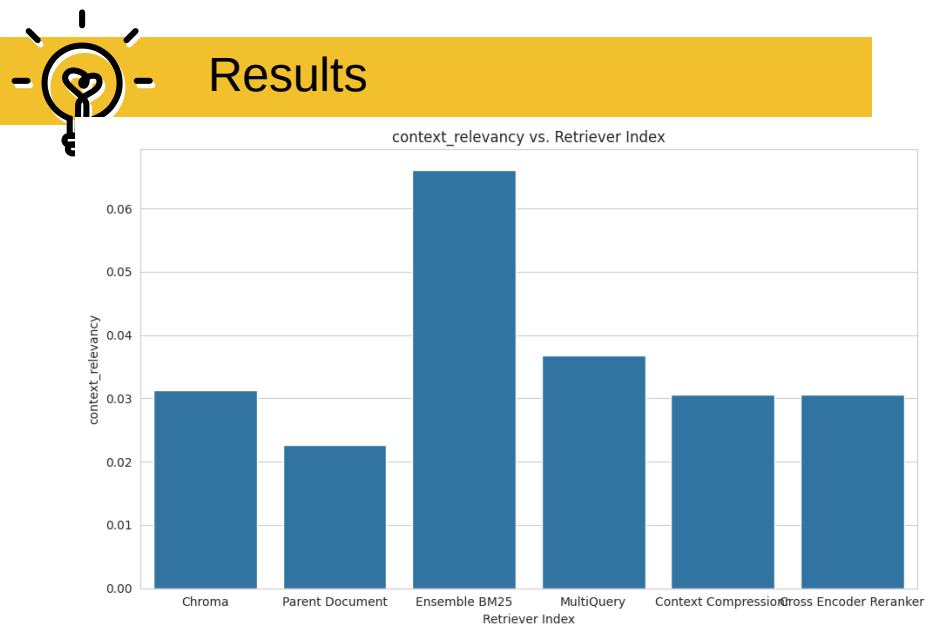
- Context Precision: Both Context Compression Retriever and Cross Encoder Reranker score a perfect approximate value of 1.0, indicating that they excel at selecting highly relevant context for the questions.
- Context Recall: Parent Document Retriever scores the highest in context recall at approximately 0.902, which indicates it is particularly good at including context that contains information present in the ground truth answers.
- Context Relevancy: It implies that the context being retrieved, while comprehensive (high context recall) is not always relevant to the query. Ensemble BM25 scores the highest.

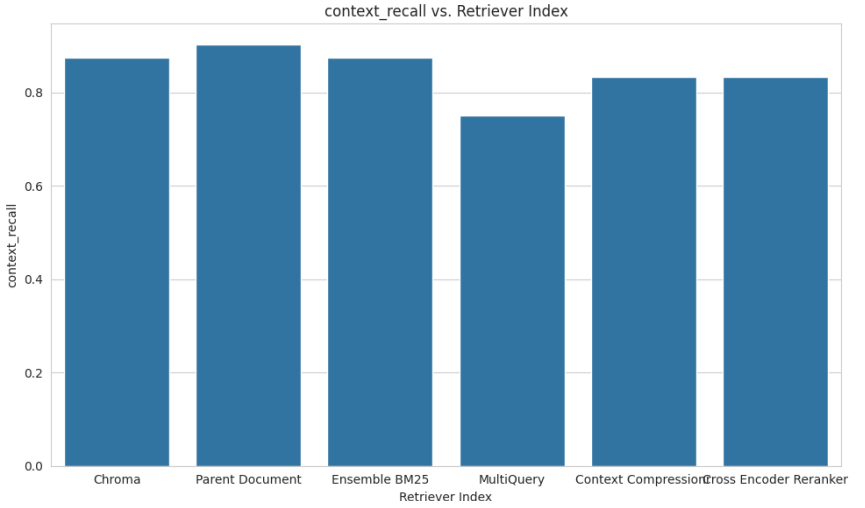
Generation Evaluation Results

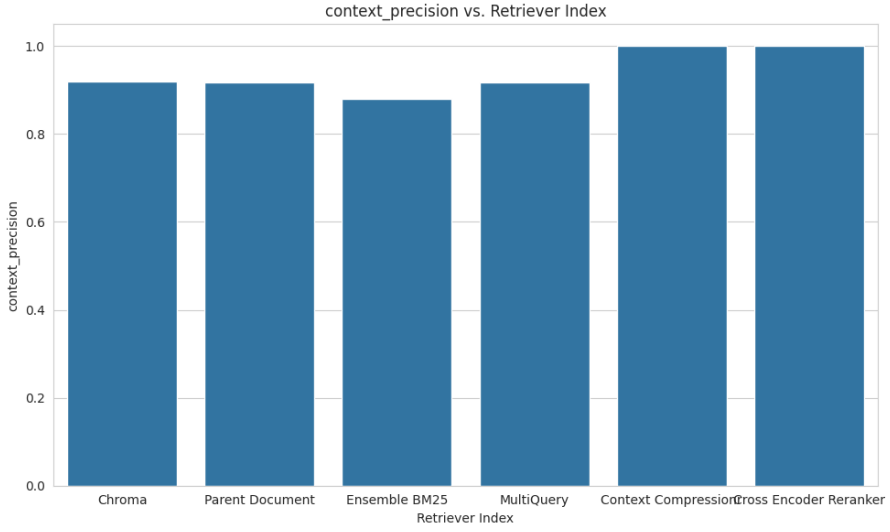
	name	context_precision	faithfulness	answer_relevancy	context_recall	context_relevancy	answer_correctness	answer_similarity
0	Chroma	0.9201	0.906200	0.813800	0.875000	0.031300	0.705200	0.895600
1	Parent Document	0.9167	1.000000	0.916900	0.901800	0.022600	0.760000	0.882000
2	Ensemble BM25	0.8807	1.000000	0.920300	0.875000	0.066000	0.683300	0.866000
3	MultiQuery	0.9165	0.943700	0.944500	0.750000	0.036800	0.698400	0.886300
4	Context Compressionr	1.0000	0.933333	0.929567	0.833333	0.030646	0.656128	0.891121
5	Cross Encoder Reranker	1.0000	1.000000	0.938041	0.833333	0.030646	0.702988	0.888144

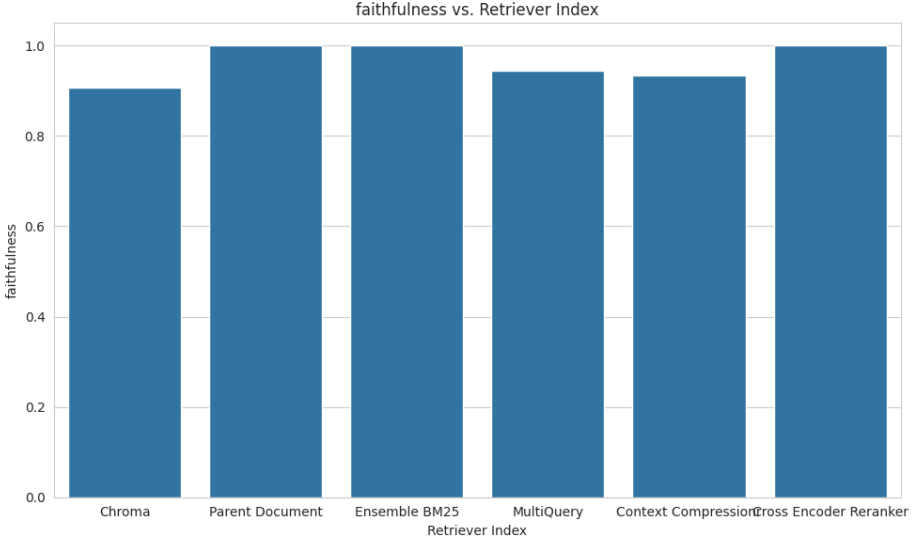
- Answer relevancy: MultiQuery Retriever leads in answer relevancy with a score of about 0.944, indicating that it provides the most relevant and complete answers in relation to the questions.
- Faithfulness: Parent Document Retriever, Ensemble BM25 Retriever, and Cross Encoder Retriever score a perfect 1.0 in faithfulness, indicating that the contexts provided by these models are completely factual and support the answers derived from them.
- Answer Correctness: Parent Document Retriever leads with a score of approximately 0.76, suggesting it is most effective at providing factually correct and semantically aligned answers with the ground truth.

Results

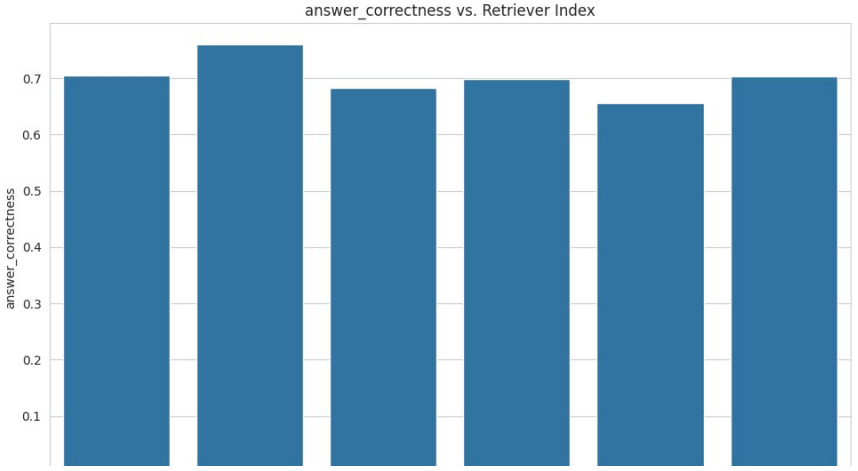

context_relevancy vs. Retriever Index
context_relevancy
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Chroma
Parent Document
Ensemble BM25
MultiQuery
Context Compression
Cross Encoder Reranker
Retriever Index

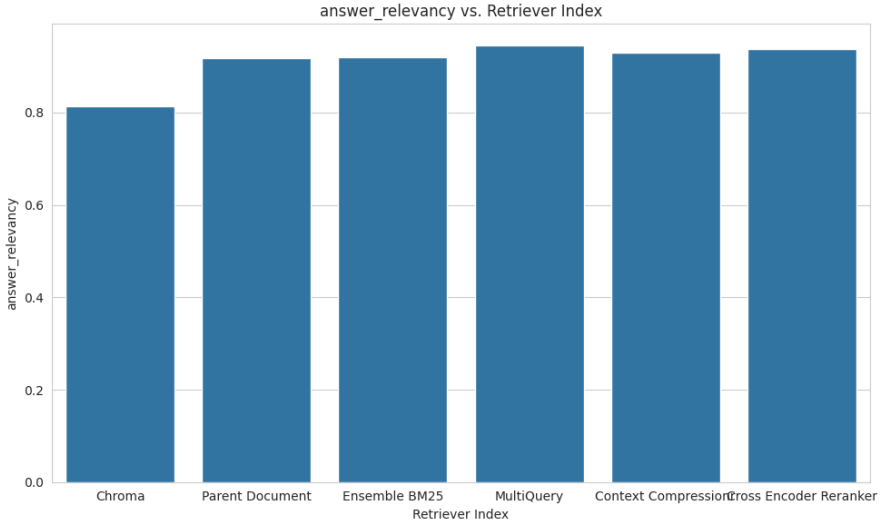

context_recall vs. Retriever Index
context_recall
0.0
0.2
0.4
0.6
0.8
Chroma
Parent Document
Ensemble BM25
MultiQuery
Context Compression
Cross Encoder Reranker
Retriever Index

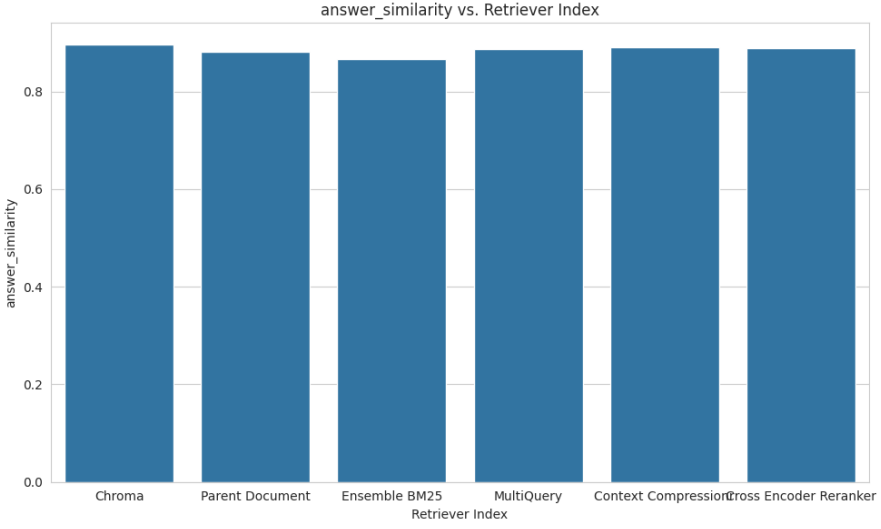

context_precision vs. Retriever Index
context_precision
0.0
0.2
0.4
0.6
0.8
1.0
Chroma
Parent Document
Ensemble BM25
MultiQuery
Context Compression
Cross Encoder Reranker
Retriever Index


faithfulness vs. Retriever Index
faithfulness
0.0
0.2
0.4
0.6
0.8
1.0
Chroma
Parent Document
Ensemble BM25
MultiQuery
Context Compression
Cross Encoder Reranker
Retriever Index

Results


answer_correctness vs. Retriever Index
answer_correctness
0.1
0.2
0.3
0.4
0.5
0.6
0.7


answer_relevancy vs. Retriever Index
answer_relevancy
0.0
0.2
0.4
0.6
0.8
Chroma
Parent Document
Ensemble BM25
MultiQuery
Context Compression
Cross Encoder Reranker
Retriever Index


answer_similarity vs. Retriever Index
answer_similarity
0.0
0.2
0.4
0.6
0.8
Chroma
Parent Document
Ensemble BM25
MultiQuery
Context Compression
Cross Encoder Reranker
Retriever Index

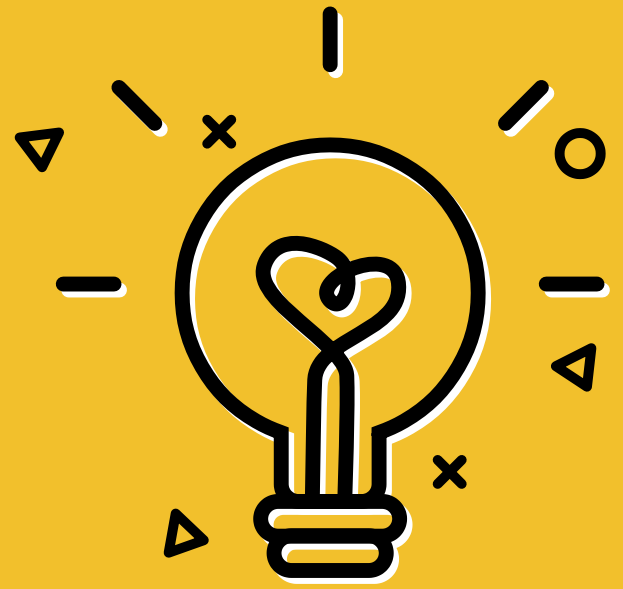
References

1. Relevance-guided Supervision for OpenQA with ColBERT,
https://arxiv.org/abs/2007.00814

2. Precise Zero-Shot Dense Retrieval without Relevance Labels,
https://arxiv.org/abs/2212.10496

3. ARAGOG: Advanced RAG Output Grading,
https://arxiv.org/pdf/2404.01037

4. RAGAS: Automated Evaluation of Retrieval Augmented Generation,
https://arxiv.org/pdf/2309.15217



THANK YOU