OPTIMISING RAG

Efficient Embedding Techniques with Approximate Nearest Neighbors (ANN)

In the current RAG architecture, embeddings for documents and queries are computed using a pre-trained sentence transformer (like all-MiniLM-L6-v2). This embedding process, especially when the dataset is large, can become computationally expensive. Optimizing the retrieval step by using Approximate Nearest Neighbor (ANN) search can significantly improve efficiency in a RAG pipeline.

Technique:

Use ANN-based search algorithms, such as FAISS (Facebook AI Similarity Search) or HNSW (Hierarchical Navigable Small World), to perform fast similarity search on document embeddings. These algorithms reduce the search time compared to exhaustive search, especially when the dataset grows large.

Why it's beneficial:

Scalability: ANN algorithms can handle very large collections of documents by indexing the embeddings, reducing the search complexity to sub-linear time.

Speed: ANN-based retrieval can be done much faster, improving the overall efficiency of the RAG model during real-time query processing.

Cost Reduction: Since ANN allows approximate rather than exact matches, the cost of inference can be reduced while still achieving high-quality results.

Example Workflow:

1. Embed Documents using the **SentenceTransformer** model.

2. Index the Embeddings using FAISS or HNSW.

3. Retrieve the Most Relevant Documents based on the query embedding using ANN search, rather than exact similarity matching.

4. Feed the Retrieved Documents into the BERT model for question answering.

Adaptive Document Retrieval with Query Expansion

In traditional RAG models, the retrieval step only considers the query as it is. However, for many complex or ambiguous queries, the retrieval process can be improved by using **Query Expansion**. Query Expansion involves modifying or enriching the original query with related terms, phrases, or synonyms to retrieve more relevant documents.

Technique:

Automatically expand the query using techniques like **Word2Vec**, **GloVe embeddings**, or a **Transformer-based model (such as BERT or T5)**. These models can generate semantically similar words or phrases, improving the chances of retrieving more relevant documents.

Why it's beneficial:

Improved Retrieval: By expanding the query, you can improve the diversity and relevance of the documents retrieved, especially for vague or under-specified queries.

Contextual Relevance: Query expansion based on contextual relationships between words helps retrieve documents that are contextually relevant to the original query, even if exact matches don’t exist.

Higher Accuracy: With a broader set of documents to choose from, the retrieval system is more likely to find the best context for answering the question.

Example Workflow:

1. **Input Query Expansion**: Use BERT or other models to expand the query by adding synonyms or related terms.

2. **Retrieve Documents**: Retrieve the top documents from the vector database (Chroma DB) using the expanded query.

3. **Answer Generation**: Feed the expanded documents into BERT for answering, allowing it to leverage the broader context.