Millstone Meeting-11-01-2024

Meeting Minutes:

Progress and Existing Problems:

1. Documentation of Meetings: It's necessary to record every meeting in documents for future reference.

2. Understanding Decision-Making: Team members must be aware of the reasons behind decisions made.

3. Open Search Implementation: The current script uses Open Search without a Python package. The suggestion is to modify variable names and other elements accordingly.

4. Query Construction in Open Search: The team currently breaks down queries into tokens and constructs Open Search queries with "match" and "should" clauses. The suggestion is to consider using "match phrase" with a "slop" parameter for better efficiency.

5. Discussion on Mapping and Preprocessing: The group needs to discuss document model mapping and whether preprocessing steps are needed separately in the script. There's a suggestion to use built-in analyzers in Open Search for this purpose.

Tutor’s Suggestions:

1. Retrieval Augmented Generation: Refer to a specific lecture (Slide 8/12 from last Monday) covering chunking and vectorization strategies in semantic search.

2. Semantic Search Approach: Concerns were raised about the current method of transforming complete abstracts into embeddings, which might ignore the end of long abstracts due to token length limitations.

3. Chunking and Vectorization Strategies: The team should think about breaking down long abstracts (chunking) and how to represent these chunks as embeddings (vectorization).

4.Query Enrichment in Open Search: The tutor suggests enhancing Open Search with query enrichment or other techniques.

5.Storage of Embeddings: Consideration is needed on where to store embeddings, given the complexity of creating embeddings for both queries and abstracts.

6. Hybrid Search Methods: Explore hybrid search methods, focusing on the use of various fields, including embedding fields, in Open Search.

7. Use of Chat GPT API for Evaluation: Generate possible questions from a set of abstracts to create a ground truth dataset. This dataset can be used to evaluate the retrieval pipeline's performance.

8. Generating Questions from Abstracts: Select a subset of abstracts to generate potential questions, establishing a ground truth for evaluation.

9. Sentence Transformer for Embeddings: Avoid excessive preprocessing of abstracts, like removing stopwords, to prevent loss of contextual information.

10.Modern Methods for Data Management: Utilize more efficient methods for storing and managing embeddings, such as vector store databases or tools like Pinecone, instead of traditional CSV and numpy files.

Additional Points:
1. Query Processing Efficiency: Emphasize efficient query processing to improve efficiency and reduce latency, especially for large queries.

2. Key Phrase Matching: Ensure that key phrases are matched correctly as whole phrases for accurate results.

3. Handling Entity Queries: Optimize the system to process lengthy and complex entity queries efficiently.

4. Efficient Processing for Large Queries: Consider methods to handle large queries with multiple sub-questions effectively, without significant delays.

These points highlight the need for effective data management, efficient query processing, and the use of modern tools and techniques in semantic search and retrieval systems.