ClearFeed Query Evaluation Techniques

Technique 1: BM25 (Okapi BM25)

BM25 is a probabilistic information retrieval model. It ranks a set of documents based on the query terms appearing in each document.

Key Components:

1. **Libraries**: rank_bm25, nltk
2. **Steps**:
    - Tokenize document texts and queries.
    - Calculate BM25 scores for the query against all documents.
    - Retrieve the top 5 documents based on scores.
3. **Output**: Top 5 ranked URLs and an OpenAI-generated response based on the fetched data.

Technique 2: TF-IDF with Cosine Similarity

TF-IDF combined with cosine similarity measures the relevance of documents to the query based on term frequency.

Key Components:

1. **Libraries**: scikit-learn (TfidfVectorizer, cosine_similarity)
2. **Steps**:
    - Compute TF-IDF embeddings for all documents.
    - Calculate cosine similarity of the query vector with document vectors.
    - Retrieve the top 5 URLs based on similarity scores.
3. **Output**: Top 5 ranked URLs and an OpenAI response.

Technique 3: Sentence Transformer with FAISS

This approach leverages Sentence Transformers for semantic embeddings and FAISS for efficient similarity search.

Key Components:

1. **Libraries**: sentence-transformers, faiss
2. **Steps**:
    - Generate document embeddings using a pre-trained Sentence Transformer model.
    - Use FAISS to index embeddings and find the nearest neighbours for the query.

    - Retrieve top 5 documents and URLs.
3. **Output**: Top 5 URLs and an OpenAI response.

Technique 4: Google Search API Integration

This method queries Google Search for URLs relevant to the query within a specific domain.

Key Components:

1. **Libraries**: requests, bs4 (BeautifulSoup)
2. **Steps**:
    - Perform a Google Search limited to the target domain using site: syntax.
    - Parse search results to extract relevant URLs.
    - Fetch data from the URLs and query OpenAI for a response.
3. **Output**: Relevant URLs and an OpenAI-generated answer.

Flask Integration

The Flask application ties these techniques together and serves as the backend API for query evaluation.

API Endpoints:

- /api/query: Accepts a POST request with query and model_number to select a technique.

Example Workflow:

1. User sends a query and selects a model number (1-4).
2. The backend evaluates the query using the specified technique.
3. Returns top 5 URLs and an AI-generated answer.

Name : Priyanshu

Email : chugh.priyanshu2013@gmail.com

Mobile : 9306348136