Project 22: Retrieval-Augmented Recipe Search and Information Retrieval By Mehrdad

Goal: Build a **recipe retrieval and question-answering system** using information retrieval methods and Retrieval-Augmented Generation (RAG). The system should allow users to type a query (e.g., *"low-fat chicken pasta under 500 calories"*) and retrieve matching recipes based on structured nutritional attributes and textual fields.

Consider HUMMUS dataset

Dataset Columns:

- recipe_id, title, duration, directions, recipe_url, tags, servingsPerRecipe, servingSize [g], calories [cal], caloriesFromFat [cal], totalFat [g], saturatedFat [g], cholesterol [mg], sodium [mg], totalCarbohydrate [g], dietaryFiber [g], sugars [g], protein [g], direction_size, ingredients_sizes, ingredients, image_url, who_score_normalized, health_category

Specifications

1. **Exploratory Data Analysis (EDA):**
    - Analyze distributions: calories, protein, fat, sodium, cooking duration.
    - Visualize recipe health categories (e.g., healthy, very_healthy, unhealthy).
    - Compute correlations (e.g., protein vs. calories).
2. **Text Preprocessing of Queries and Recipes:**
    - Normalize recipe titles, direction, tags, and ingredients (lowercasing, tokenization, stopword removal).
    - Preprocess user queries in the same way for consistency.
    - Compare statistics: average ingredient length, number of tags per recipe.
3. **Indexing for Search:**
    - Build an inverted index using recipe title, tags, ingredients, and directions.
    - Implement basic keyword search with TF-IDF or BM25.
    - Example query: *"gluten-free pasta with tomato"* → retrieve recipes with relevant tags + title.
4. **Semantic Embeddings for Retrieval:**
    - Use BERT/Doc2Vec embeddings for semantic similarity between queries and recipes.
    - Compare keyword-based (BM25) vs. semantic search.
    - Example: Query *"low-carb chicken meal"* should match recipes tagged as "keto" or "low-carb" even if not exact matches.
5. **Hybrid Search:**
    - Combine keyword-based (BM25) and embedding-based retrieval with a weighted score.
    - Evaluate whether hybrid search improves retrieval over single methods.
6. **Structured Attribute Filtering:**
    - Allow structured filters based on numeric fields.
    - Example queries:
        - *"recipes under 400 calories"*
        - *"high-protein (>20g) vegetarian dish"*
    - Implement SQL-style filtering combined with IR.
7. **Query Understanding & Expansion:**
    - Implement query expansion with synonyms (e.g., *"aubergine"* → *"eggplant"*).
    - Handle vague queries like *"healthy dessert"* by expanding to tags + nutrition thresholds.
8. **RAG (Retrieval-Augmented Generation):**
    - Integrate a pre-trained LLM (e.g., GPT, BERT-QA, LLaMA) with the retrieval pipeline.
    - System pipeline:

1. Retrieve top-k candidate recipes.
2. Pass them as context to the LLM.
3. Generate a natural language answer explaining the recommendation.

- Example Query: *"Suggest a quick vegetarian dinner under 20 minutes."*
  - Retrieval: recipes with duration < 20 and tag vegetarian.
  - LLM: *"Here are some options: Quick Veggie Stir Fry (15 min, 250 calories), Tomato Basil Pasta (18 min, 400 calories). Both are vegetarian and under 20 minutes."*

9. **Evaluation of Retrieval Models:**
   - Create a set of test queries with known relevant recipes.
   - Metrics: Precision@k, Recall@k, MRR (Mean Reciprocal Rank), NDCG.
   - Compare keyword search vs. embeddings vs. hybrid vs. RAG-based answers.
10. **Explainability and User Experience:**
    - Generate explanations for retrieved results (e.g., "This recipe matches because it has <400 calories and contains chicken.").
    - Include highlighted snippets from directions/ingredients in the results page.
11. **Visualization & Interface (Optional but Encouraged):**
    - Build a simple web interface (Streamlit/Flask/React).
    - User enters query → system returns ranked list with:
      - Recipe title
      - Image (image_url)
      - Calories/protein/fat summary
      - Explanation (from RAG or rule-based).
12. **Extensions (Optional Advanced Task):**
    - Personalization: incorporate user preferences (e.g., low-salt diet, disliked ingredients).
    - Multi-turn retrieval: allow conversational queries like "Show me vegan desserts. Now only under 300 calories."
    - Compare RAG with pure retrieval-based QA (BM25 + extractive QA model).