Revised Midterm exercise

Implementing Hierarchical indices in RAG

Personnel

- **Team Members:**
  - Aviv Itzhak Barel Balulu- CEO at Indexify.AI
  - Lihu Zur- CTO at Indexify.AI
  - Ali Abu-Liel- Senior ML Engineer at Indexify.AI
  - Basel Amin- Senior Data Scientist at Indexify.AI
- **Client Representatives:**
  - Avi Israeli- experienced freelance financial advisor.
  - Amit Cohen- beginner trader.
  - Together Investments- A venture capital firm that focuses on facilitating investments in high-tech and startup ventures.
- **Academic Advisors:**
  - Dr. Ishai Rozenberg

Business Background

- **Client**: Users who are interested in financial advice (investors, individuals, etc.)
- **B2C Product-** directly serves individual consumers by assisting with financial tasks and providing personalized advice.
- **Client requirements:**

  A faithful, efficient financial bot which can provide both detailed and concise answers on need, on a variety of financial topics and from different points of view (company, young individual, rich individual, advisor, etc.)
- **Business Domain**: Financial services, specifically real-time news aggregation and analysis for investment decision-making.
- **Problem:**
  - The existing system struggles to efficiently handle large-scale financial data retrieval due to flat indexing in the vector database. By large-scale we mean: given that our system handles a big number of user requests, and the search is not efficient enough to support all of them.
  - Relevant documents are being missed while also irrelevant documents are being fetched, also financial bot returns irrelevant responses and hallucinates.

- **Business KPIs**: Our solution will focus on business KPIs: churn, satisfaction, conversion rates and operational efficiency:
    - **Churn**: Inefficient data retrieval and irrelevant responses frustrate customers, increasing the likelihood they'll abandon the service, which raises churn rates. Improving accuracy reduces churn by fostering trust and satisfaction.
    - **Customer Satisfaction**: Incorrect or missing information lowers customer satisfaction. Enhancing the bot's ability to deliver relevant answers boosts satisfaction by providing a more reliable experience.
    - **Conversion Rate**: Irrelevant responses harm user trust, lowering conversion rates. By ensuring the bot gives accurate financial advice, users are more likely to engage and act on recommendations.
    - **Operational Efficiency**: Irrelevant responses lead to more support tickets and manual fixes, draining resources. Improving the bot's accuracy reduces these costs and improves overall efficiency.

Scope

- **Data Science Solutions**:
    - Enhance the **financial assistant bot** by implementing **hierarchical indexing** in the vector database and improve the RAG by integrating with hierarchical indices in the feature pipeline and the inference pipeline
- **What We Will Do**:
    - Redesign the vector database to implementing **hierarchical indexing** utilizing three levels of encoding: A general sector, the subject, and the specific described operation. In addition, we will increase the existing top-k value so our model can benefit from the new hierarchy.
- **Consumption by Users**:
    - Example: Queries like "What are the latest financial trends in the healthcare sector?" or "Should I invest in Microsoft?" will retrieve results and with improved relevance due to the hierarchical structure. (Removed: faster)

Business Justification

Example: ***Query: "Should I invest in Microsoft?"***

Baseline System (Without Chunking and Summarization):

1. **Query Embedding:**
   - The query is embedded and searched against all documents in the database.
   - Example retrieved data: A 5,000-word financial report on Microsoft, including unrelated sections like company history, and legal disputes.
   - **Result:** The system retrieves the entire document, making it harder to extract relevant insights.

With Hierarchical Indexing:

1. **Hierarchical Indexing:**
   During preprocessing, the original documents were organized into 3 levels of hierarchical encoding:
   - **General Sector:** E.g., "Technology"
   - **Subject:** E.g., "Microsoft"
   - **Specific Operation:** E.g., "Release of a new product"
2. **Improved Retrieval:**
   - Classify the query into the same 3 levels of hierarchy, and fetch only documents with matching context
     - **Example 1 for retrieved context document:** "Microsoft announced a 10% increase in revenue…"
     - **Example 2 for retrieved context document:** "Microsoft just bought a new successful startup."
     - **Example 3 for retrieved context document:** "Microsoft in undergoing an antitrust scrutiny investigation in Europe…"

Generated Response Using Hierarchical Indexing:
"Microsoft's recent growth is driven by a 10% increase in revenue, primarily due to innovation and investments in new ideas. However, ongoing antitrust scrutiny in Europe could pose risks to its
operations. Carefully consider these factors before investing."

Architecture

- **Data Flow:**
  - **Ingestion**: Financial news streamed in real-time, pre-processed, and embedded.
  - **Indexing**: Hierarchical indices in the vector DB categorize embeddings by sector, subject, and operation.
  - **Inference**: LangChain uses the hierarchical index to improve query accuracy and reduce latency.
- **Tools:**
  - **Vector Database**: Qdrant for storing and retrieving embeddings.
  - **LangChain**: For managing the inference pipeline with Retrieval-Augmented Generation (RAG).
  - **ChatGPT**: For classifying the documents and creating the hierarchy.
  - **Azure**: Hosts the pipelines with scalable and serverless infrastructure.

Workflow Details:

Training Pipeline:

Loading the pre-trained Falcon 7B model, and performs fine-tuning on it, using training and testing Questions and Answers samples. Each sample include about me and context sections, in addition to the question and the ground-truth answer. This helps to model to learn how to answer the financial questions for different users and different contexts.

Streaming Pipeline:

The process begins with fetching financial news data from Alpaca, followed by cleaning and transforming the text into a machine-readable format. Each document will go through an hierarchical flow of classification into a 3-level hierarchy using ChatGPT API. The final accumulated hierarchy is saved for the inference to use later. The documents with their classifications are re-encoded into embeddings and stored in a vector database for efficient retrieval.

Inference Pipeline:

The process starts with classifying the user query into the same 3-level hierarchy as in the streaming pipeline, using the given options in the accumulated hierarchy, and then fetching only the relevant documents from Qdrant according to this classification. Then, embedding the query and performing a semantic search on the retrieved documents to find the most similar ones to the query. Finally the Falcon 7B receives the original user query and the context from the chosen documents, and generates a response.

Quantifiable Metrics

We chose to evaluate the impact of system improvements by focusing on key performance indicators.

1. **Answer Similarity:**
   - **Definition:** measures how closely the generated response matches a reference answer.
   - **Goal:** Increase Answer Similarity in approximately 10%
   - **KPI Correlation:** improves the relevance of responses, which directly boosts **customer satisfaction** and **conversion rates** by providing more accurate and consistent information. It also reduces **churn** by fostering user trust through reliable answers.
2. **Faithfulness:**
   - **Definition:** The degree to which the system's generated responses are directly supported by the retrieved documents.
   - **Goal:** Increase faithfulness in approximately 10%
   - **KPI Correlation:** ensures that the bot delivers accurate and truthful responses, enhancing **customer satisfaction** and **conversion rates**, while also increasing **operational efficiency** by reducing the need for manual corrections or support intervention.

Impact on Metrics

we anticipate significant improvements in the system's overall performance and reliability through targeted enhancements. These changes are expected to refine its ability to deliver accurate and relevant outputs while addressing critical user needs.

1. **Answer Similarity:**

- **Improvement:** The retrieved context will be significantly more related to the query.
- **Effect:** Increasing the chance to generate a better answer, which would match the (good) ground-truth answer

2. **Faithfulness:**

- **Improvement:** The retrieved context is ensured to be contextually-correct, prioritizing highly-related data.
- **Effect:** ensuring higher accuracy in responses and significantly reducing hallucinations.

Plan

Timeline: 6 weeks

Phases:

1. **Phase 1: Analysis and Design (1 week)**
    - Evaluate limitations of the current system, Design hierarchical indexing strategy.
2. **Phase 2: Hierarchical Indexing Implementation (2 weeks)**
    - Implementing the Hierarchical Indexing approach.
    - Integrate hierarchical indices into the vector database (Qdrant).
3. **Phase 3: Pipeline Integration (1 week)**
    - Update the streaming pipeline, inference pipeline to leverage hierarchical indices for embedding new financial data, and query routing.
4. **Phase 4: Testing and Optimization (1 week)**
    - Conduct rigorous testing for accuracy and benchmark the enhanced system against the baseline.
5. **Phase 5: Final Evaluation and Submission (1 week)**
    - Gather results, finalize documentation, and prepare for project submission.

Communication

- **Weekly Team Meetings**: Discuss progress, challenges, and next steps.
- **Advisor Feedback Sessions**: Initial feedback on the midterm exercise after reviewing the idea and final feedback after delivering the code and presenting the improved version in class.
- **Documentation**: Maintain a shared repository for code, test results, and report.

Deliverables

- An enhanced financial assistant bot with hierarchical indexing.
- Documentation of improvements and performance benchmarks.
- Final presentation to peers on the MLOPS course at Reichman University.