STOCK NEWS QUESTION AND ANSWERING RAG BOT

Retrieval-Augmented Generation

Stock News RAG System

- Retrieves relevant stock news based on user queries
- Augments LLM context with news metadata
- Generates personalized answers with explanations

Dataset

- **Source:** Stock news articles embedded in JSON format
- **Content:** News title, link, ticker, full article, company

Architecture Overview

Mermaid Graph:

```
A[User Query] → B[Query Processing]
B → C[Embedding Generation]
C → D[Vector Search]
D → E[Relevant News Retrieval]
E → F[Context Augmentation]
F → G[LLM Generation]
G → H[Structured Response]
```

Core Components

- **Data Pipeline:** Loading and preprocessing news metadata
- **Embedding System:** Converting text to vector representations
- **Vector Database:** Efficient similarity search with ChromaDB
- **Retrieval Engine:** Multiple strategies for finding relevant content
- **Generation System:** Multiple LLMs for response creation

Implementation Details

Contents

```
├── README.md # This documentation └── Stock_News_QnA.ipynb # Complete implementation notebook
```

Technology Stack

- **Embeddings:** SentenceTransformers (all-mpnet-base-v2)
- **Vector Database:** ChromaDB for similarity search
- **Language Model:** Gemini, Cohere, Llama and Mistral for response generation
- **Data Processing:** Pandas for metadata manipulation
- **Structured Output:** Pydantic models for response formatting

RAG Workflow Implementation

- **Data Loading and Preprocessing**
  Load and preprocess news data
- **Embedding Creation**
  Generate vector embeddings for news and its metadata
- **Vector Database Setup**
  ChromaDB collection for similarity search

- **Retrieval Strategies**

Basic Retrieval

- Simple semantic similarity search
- Query expansion for improved recall
- Relevance scoring and ranking

HyDE (Hypothetical Document Embedding)

- Generate hypothetical news descriptions
- Enhanced semantic matching
- Better retrieval for abstract queries

Query Decomposition

- Break complex queries into sub-queries
- Comprehensive result aggregation

Key Features

Multi-Strategy Retrieval

- **Semantic Search:** Understanding query intent and context
- **Hybrid Approaches:** Combining multiple retrieval methods
- **Query Enhancement:** Expanding and refining user queries

Retrieval Methods Comparison

Method	Strengths	Use Cases	Performance
Basic Retrieval	Simple, fast, reliable	Direct news searches	High precision

Method	Strengths	Use Cases	Performance
HyDE	Better abstract queries	Inspiring stories	Medium precision, high recall
Query Decomposition	Complex multi-part queries	Detailed requirement matching	High coverage

Technical Implementation

Embedding Pipeline

- **Chunking Strategy:** Optimize text segments for embedding
- **Vector Generation:** Create high-quality embeddings
- **Storage Optimization:** Efficient vector database management

Retrieval Optimization

- **Similarity Thresholds:** Balanced precision-recall trade-offs
- **Result Ranking:** Multi-factor relevance scoring
- **Diversity Enhancement:** Avoid redundant information
- **Context Window Management:** Optimal information selection

Response Quality

- **Fact Checking:** Verify news information accuracy
- **Coherence Validation:** Ensure logical explanations
- **Personalization:** Adapt responses to user preferences
- **Safety Filtering:** Remove inappropriate content

Performance Metrics

Retrieval Quality

- **Relevance Score:** How well retrieved news matches queries
- **Diversity Index:** Variety in news types
- **Coverage Rate:** Percentage of database effectively searchable
- **Response Time:** Query processing and generation speed

User Experience

- **Answer Accuracy:** User satisfaction with suggestions
- **Explanation Quality:** Clarity and helpfulness of reasoning
- **System Responsiveness:** End-to-end response times
- **Result Consistency:** Stable performance across query types

Next Steps

- **Experiment with Data:** Try different datasets, like quarterly and yearly financial reports
- **Optimize Performance:** Fine-tune retrieval and generation parameters
- **Scale the System:** Implement production-ready optimizations
- **Add Features:** Incorporate user feedback and personalization