Assignment Details:

Code Repository: https://github.com/ManishPant97/Text-Summarization-NER

Instructions to run the code: Available in readme.md in the code repo.

Dataset:https://www.kaggle.com/datasets/sayelabualigah/high-quality-financial-news-dataset-for-nlp-tasks?select=dataset.csv

Agentic System Concept: LLM based News Search Engine (Semantic meaning & Keyword based hybrid search)

Goal:

This document outlines an agentic concept for a sophisticated news search engine. The agent's core function is to take a user's natural language query and return the most relevant news articles by combining vector-based semantic search with traditional keyword matching.

Example:

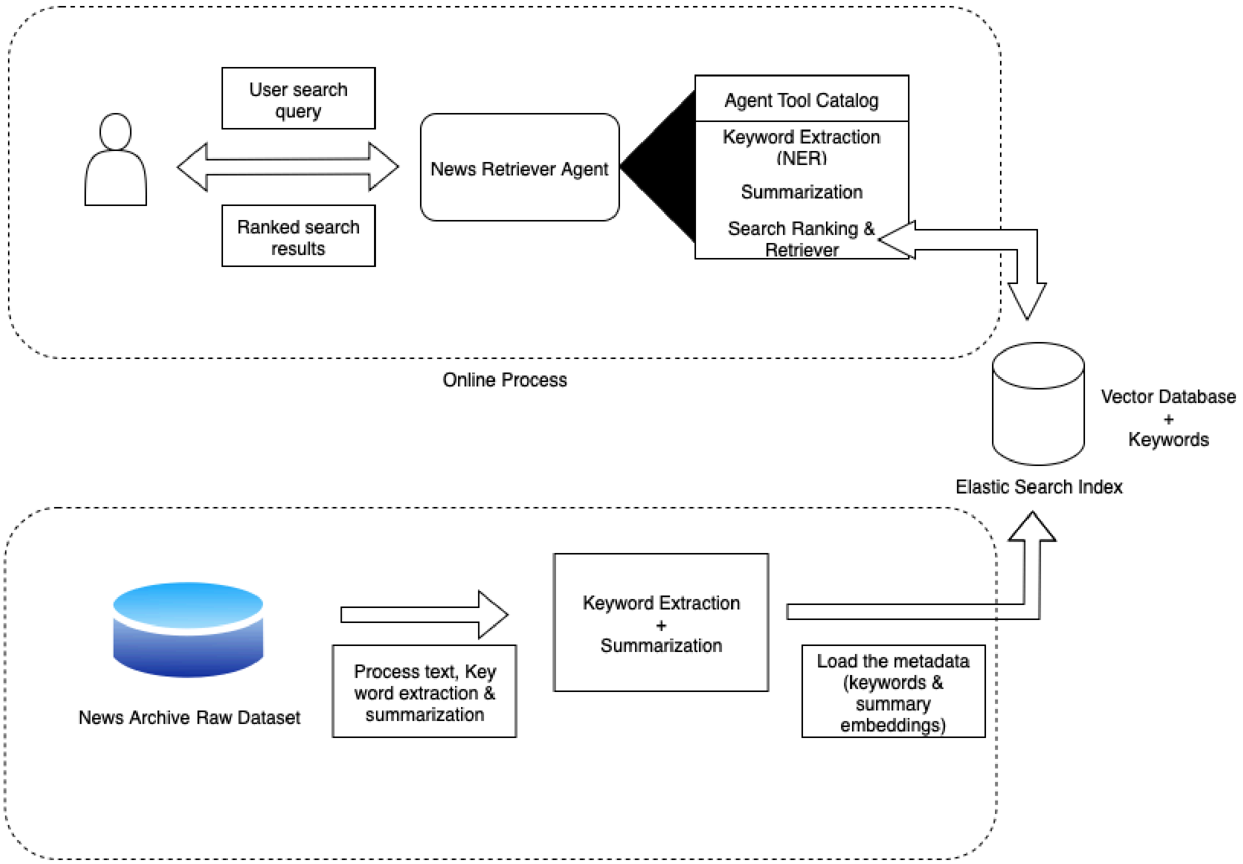
Search Query: News of impact on Apple’s share prices due to Sam Altman’s OpenAI.

Expected News Results:

News 1: Why OpenAI's latest moves could signal more trouble ahead for Apple's stock

News N: Lorem Ipsum

Solution Architecture:


User search query
Ranked search results
News Retriever Agent
Agent Tool Catalog
Keyword Extraction (NER)
Summarization
Search Ranking & Retriever
Online Process
Vector Database + Keywords
Elastic Search Index
News Archive Raw Dataset
Process text, Key word extraction & summarization
Keyword Extraction + Summarization
Load the metadata (keywords & summary embeddings)

Architecture Components:

- **News Retriever Agent (Orchestrator):** The brain of the operation. It receives the user query and determines the best course of action by selecting and chaining the available tools. It's an LLM with access to the following tools.
- **Keyword Extractor Tool:** A function that takes a text input (the user's query) and returns a list of important keywords.
- **Summarizer Tool:** A function that takes a long text (the retrieved article's content) and generates a concise summary.
- **Search Ranking & Retriever Tool:** A complex tool that connects with an Elasticsearch index. This tool's function is to execute a search query against the index. It takes the results from both the semantic and keyword searches, re-ranks them based on relevance scores, and prepares them for the final output.
- **Elasticsearch Index:** This is the core data store. Each document in the index represents a news article and contains the following fields:
    - title: The title of the article.
    - url: The URL of the article.
    - summary_text: The article summary.

    - summary_embedding: A dense vector representation of the article's summary, used for semantic search.
    - keywords: A list of keywords associated with the article, used for traditional keyword matching.
    - publication_date: The date of publication.

Agent Reasoning/Planning + Tool use (Agentic Workflow):

The agent's reasoning process is a chain of thought that combines tool use with logical decision-making.

1. **Receive Query:** The agent receives the user's query: user_query = "2022 world cup expenses & impact on local economy of the host"

2. **Initial Analysis:** The agent's first internal thought is to break down the query and decide which tools are needed. It recognizes the need for tools like keyword extraction, summarization and search ranking to fulfill the request completely.

3. **Tool Call: Keyword Extraction:** The agent decides to use the keyword_extractor tool first to get specific terms.
    - Tool Output: ["World Cup", "2022", "economy", “host”]

4. **Tool Call: Summarization:** The agent summarizes the user query if the provided search query is a long-text.
    - Tool Output: <Concise search query>

5. **Tool Call: Search Ranking & Retrieve:** The agent then calls the search_and_retrieve tool, passing both the original query and the extracted keywords.
    - Internal Logic: The tool will perform two searches against Elasticsearch. The semantic search will find articles based on semantic embeddings & keyword search. It uses a simple internal re-ranking logic to combine the two sets of results based on their relevance.
    - Tool Output: A JSON object with ranked search results and their metadata.
6. **Final Formatting & Presentation:** The agent takes the top N (e.g., 5-10) re-ranked articles. For each article, it formats the output cleanly, including the title, a link, and the article summary from the Elasticsearch index.
    - Reasoning: "The user wants a clear and concise answer. I will present the results as a bulleted list with key information for each article."
    - Final Output (Example):
        - Qatar’s economic boost from World Cup 2022
            - URL: https://example.com/article1

  - Summary: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua"
- FIFA world cup gave a bull run to hotel industry in Qatar
  - URL: https://example.com/article2
  - Summary: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua"

Memory: The agent would not necessarily need a long-term memory for this specific task, as each query is a new, independent request. However, memory could be beneficial for a conversational interface if needed to be built as a chatbot instead of a search engine.