Documentation for News Agent

Project Overview

This project builds an AI-powered backend agent for analyzing a news articles dataset and answering natural language queries. The agent supports:

- Rich **preprocessing** of news data for better text analysis
- **Entity extraction** and **statistical analysis** of news content
- **Semantic search** and **natural language query interpretation** over tabular news data
- **LLM-powered reasoning** to generate human-friendly answers and insights
- Extensible architecture for summarization, exploration, and dynamic query answering

---

Dataset Description

The dataset contains news articles with the following main columns:

Column	Description
`link`	URL to the news article
`headline`	Title of the news article
`category`	News category (e.g., U.S. NEWS, COMEDY)
`short_description`	Brief excerpt or summary of the article
`authors`	Author(s) of the article
`date`	Publication date (datetime)

Additional **preprocessed columns** generated include:

- Tokenized and stop-word removed versions of `headline` and `short_description`
- Word count lengths of headlines and descriptions
- Extracted year from the publication date

- Extracted entities such as dates and numbers (using regex)

Preprocessing Steps

- **Tokenization:** Split headlines and descriptions into word tokens.
- **Lowercasing & Stopword Removal:** Clean text by lowercasing and removing common English stopwords for better analysis.
- **Length Calculation:** Compute word counts per article headline and description.
- **Extraction (Regex):** Extract date-like and number-like tokens from descriptions and publication year from `date` for time-based analysis.

These steps enrich the dataset for effective querying, semantic search, and statistics.

Agent Architecture

Core Components

- **Data Loading:** Reads preprocessed CSV data into a pandas DataFrame.
- **Embedding Creation:** Uses SentenceTransformer model (`all-MiniLM-L6-v2`) to embed text rows into a Chroma vector store for semantic search.
- **Large Language Model (LLM):** Groq-based LLM (`llama3-70b-8192`) is used for:
    - Generating pandas code from natural language queries
    - Answering questions based on data summaries and semantic search results
- **Pandas Query Executor:** Executes safe pandas code generated by LLM to return tabular data or statistics.
- **Natural Language Interface:** FastAPI endpoints receive user questions and return AI-generated answers or pandas query results.
- **Intent Routing:** Simple keyword-based classification routes queries to:
    - Statistical summaries (counts, distributions)
    - Semantic search + LLM reasoning
    - Code generation and execution for complex data queries

Workflow

1. **User sends a natural language query** to `/ask`.
2. The agent detects intent (summary/statistics, direct data query, or semantic search).
3. For stats/summaries, a prompt is constructed with descriptive stats fed to the LLM for a natural language answer.

4. For direct data queries, LLM generates pandas code to filter or analyze data, which is executed and results returned.
5. For semantic queries, embedding-based search retrieves relevant rows, which are provided as context to LLM for answering.
6. Results are returned as JSON responses with explanations and answers.

Installation & Setup

Prerequisites

- Python 3.9+
- Install dependencies:

bash
CopyEdit
```
pip install fastapi uvicorn pandas sentence-transformers chromadb
langchain_groq pydantic python-dotenv matplotlib seaborn
```

- Obtain API keys for Groq LLM and set in environment variables:

bash
CopyEdit
```
export GROQ_API_KEY="your_api_key_here"
```

Data Preparation

- Place your cleaned/preprocessed news data CSV as `data/cleaned_news_data.csv`.
- Ensure it contains the required columns as described above.

Running the API

Start the FastAPI server locally on port 5001:

bash
CopyEdit
```
uvicorn app.main:app --reload --port 5001
```

Access the root endpoint to check status:

cpp
CopyEdit

```
GET http://127.0.0.1:5001/
Response: {"message": "FastAPI backend is running."}
```

Ask Endpoint

Send POST requests to `/ask` with JSON body:

json
CopyEdit

```
{
  "question": "How many news articles were published in 2022?"
}
```

The API responds with a natural language answer generated by the agent.

Example Queries Supported

- “How many news articles are in each category?”
- “List the top 10 authors by number of articles.”
- “What is the distribution of news by year?”
- “Show me all articles published after 2022-01-01.”
- “Summarize the main topics covered in parenting news.”