

# Assignment: Real-Time Market Sentiment Analyzer Using LangChain Chains

## Objective

Build a LangChain-powered pipeline (Chain) that:

1. Accepts a **company name** as input.
2. Extracts or generates its **stock code**.
3. Uses Yahoo Finance tools in LangChain to **fetch news** about the company.
4. Sends the news to an **LLM (Azure OpenAI GPT-4o or GPT-4o-mini)** to generate a structured **sentiment profile**.
5. Outputs the result as a **JSON object**.
6. Uses **Langfuse** for tracing, prompt debugging, and monitoring.

## Tech Stack & Tools

- **Framework:** LangChain
- **LLM:** GPT-4o or GPT-4o-mini (deployed via Azure OpenAI)
- **Data Source:** Yahoo Finance tool in LangChain
- **Prompt Management & Observability:** Langfuse
- **Environment:** Python (v3.10+ recommended)

## Tasks Breakdown

### Step 1: Accept Input

- Accept a company name as input (e.g., "Apple Inc").

### Step 2: Get Stock Code

- Generate or extract the stock ticker/symbol using:
  - Either a static lookup table or an API/tool (like Yahoo Finance Symbol Suggest).
  - Integrate this as the first link in your chain.

### Step 3: Fetch Company News

- Use LangChain's integration with Yahoo Finance tools to fetch the latest news for the extracted stock symbol.

- Extract a concise list of **recent headlines or article summaries**.

#### **Step 4: Analyze Sentiment with GPT-4o / GPT-4o-mini**

- Pass the news summaries to the GPT-4o model via LangChain with a prompt that asks the LLM to:
  - Classify sentiment.
  - Extract named entities: people, places, other companies.
  - Provide a structured JSON with the following fields:

```
{  
  "company_name": "",  
  "stock_code": "",  
  "newsdesc": "",  
  "sentiment": "Positive/Negative/Neutral",  
  "people_names": [],  
  "places_names": [],  
  "other_companies_referred": [],  
  "related_industries": [],  
  "market_implications": "",  
  "confidence_score": 0.0  
}
```

*Tip:* Use StructuredOutputParser from LangChain or PydanticOutputParser for JSON formatting.

#### **Step 5: Integrate Langfuse**

- Log prompts, outputs, and metadata using Langfuse.
- Add tracing spans for:
  - Stock code extraction.
  - News fetching.
  - Sentiment parsing.

## Deliverables

1. **Python Script / Notebook (.py or .ipynb)** with:
  - Full implementation of the chain.
  - LangChain Chain definition.
  - Proper use of Azure OpenAI APIs and Langfuse.
2. **README** with:
  - Setup instructions.
  - Azure and Langfuse API setup steps.
  - Sample command to run the chain.
3. **Sample Output JSON** with a real example for a company like "Microsoft".

## Bonus Ideas (Optional)

- Add **entity linking** for extracted people and companies (e.g., using Wikipedia or LinkedIn).
- Visualize sentiment trend if historical news is considered.
- Use LangChain's **MultiPromptChain** to classify and then process different types of news differently.