

Induction - Step 1

Name Disambiguation Library for Companies, Ticker Symbols, and Exchanges

The goal is to build a name disambiguation library that extracts and identifies company names, ticker symbols, and exchange names from a given text. The library should handle both structured inputs (e.g., "NASDAQ:AAPL") and unstructured mentions (e.g., "Tesla") to provide accurate disambiguation. This tool will be useful for processing financial documents, news articles, or user inputs to extract meaningful data. You can not use any LLM model for this.

You must complete and submit this assignment within 48 hours of receiving it.

Requirements:

Core Functionality:

1. Input:

- A string of text that may contain references to companies, ticker symbols, and/or exchange names.

2. Output:

- **Company Name:** The full name of the company.
- **Ticker Symbol:** The stock ticker symbol of the company.
- **Exchange Name:** The name of the stock exchange where the company is listed.

3. Capabilities:

- Parse structured inputs, such as "NASDAQ:AAPL" or "NYSE:MSFT."
- Disambiguate unstructured mentions of companies (e.g., "Tesla") by matching against a predefined database or API.
- Handle case variations and potential typos in text (e.g., "nasdaq:aapl" or "Telsa" should still resolve correctly).
- Return None or a specific error if the input cannot be resolved.

Implementation Steps:

1. Database/Source of Truth:

- Use or integrate with a source of truth for company and ticker data, such as:
 - Public APIs like Yahoo Finance, Alpha Vantage, or IEX Cloud.
 - A static dataset (e.g., CSV) containing S&P 500 tickers, company names, and exchange names.

2. Text Parsing:

- **Structured Inputs:**
 - Extract the exchange name and ticker symbol directly from formats like "NASDAQ:AAPL" or "NYSE:MSFT."
- **Unstructured Inputs:**
 - Use string matching or natural language processing (NLP) techniques to identify company mentions in unstructured text.

3. Disambiguation Logic:

- Map ticker symbols to companies and exchanges using the database.
- For unstructured inputs, search for the company name in the database and retrieve its ticker symbol and exchange.

4. Error Handling:

- Handle ambiguous cases where multiple companies may match a single input.
- Gracefully manage invalid inputs (e.g., "RandomText") by returning an appropriate error or empty result.

5. Optional Enhancements:

- Implement fuzzy matching to handle typos or minor spelling errors.
- Add support for resolving international companies and exchanges.

Examples:

1. Input:

- "NASDAQ:AAPL"
- **Output:**
 - Company Name: Apple Inc.
 - Ticker Symbol: AAPL
 - Exchange Name: NASDAQ

2. Input:

- "Tesla"
- **Output:**
 - Company Name: Tesla Inc.
 - Ticker Symbol: TSLA
 - Exchange Name: NASDAQ

3. Input:

- "NYSE:MSFT"
- **Output:**
 - Company Name: Microsoft Corporation
 - Ticker Symbol: MSFT
 - Exchange Name: NYSE

4. Input:

- "Google"
- **Output:**
 - Company Name: Alphabet Inc.
 - Ticker Symbol: GOOG

- Exchange Name: NASDAQ

Deliverables:

1. Codebase:

- A Python library that includes:
 - Functions for parsing structured and unstructured inputs.
 - Database or API integration for company and ticker data.

2. Documentation:

- A README file with setup instructions and usage examples.
- Docstrings for all functions, explaining inputs, outputs, and exceptions.

3. Test Cases:

- Unit tests to validate parsing logic, disambiguation accuracy, and error handling.
- Example test inputs:
 - "NASDAQ:AAPL"
 - "Tesla"
 - "RandomText"

4. Optional:

- Provide an interactive script or command-line tool to test the library.

Evaluation Criteria:

1. Functionality:

- Accurate parsing and disambiguation of names, tickers, and exchanges.

2. Code Quality:

- Clean, modular, and maintainable code.

3. **Error Handling:**

- Robust handling of ambiguous or invalid inputs.

4. **Documentation:**

- Clear setup instructions and function-level explanations.

5. **Testing:**

- Comprehensive test coverage for all functionalities.

Submission:

The submission must be made through a private GitHub repository, to be shared with launch@panscience.xyz

Alternatively, you can also put all the deliverables on a Google Drive and share it with launch@panscience.xyz

In case of any question, please reach out to launch@panscience.xyz