TAIYŌAI INC DATA EXTRACTION AND STANDARDIZATION (WEB SCRAPPING) USING LLM

Part 1: Research and Data Sourcing

I have researched and identified the top 8 URL websites that provide reliable data on construction and infrastructure projects and tenders in California. To ensure comprehensive coverage and reliability, I utilized ChatGPT to find the best URL sources for data on construction and infrastructure projects and tenders in California.

- https://data.ca.gov/dataset/?q=construction+and+infrastructure+projects&sort=score+desc%2C+metadata_modified+desc', 'https://dot.ca.gov/programs/procurement-and-contracts/bid-opportunities'
- https://www.cityofarcata.org/413/Current-City-Construction-Projects
- 'https://www.cityofsanrafael.org/major-planning-projects-2/
- 'https://www.elkgrovecity.org/southeast-policy-area/development-projects'
- 'https://www.fluor.com/market-reach/industries/infrastructure'
- 'https://www.fluor.com/projects'
- 'https://www.toaks.org/departments/public-works/construction'

I have performed web scraping using the provided URLs as well as additional URLs I identified through my research. This process involved extracting detailed data on construction and infrastructure projects and tenders in California from these reliable sources.

Part 2: Data Extraction and Standardization

Script Explanation

Initialization and URL Extraction:

The input URL is assumed to be the homepage of the target website.

Using the Beautiful Soup Python package, all available URLs on the homepage are extracted by identifying <a> anchor tags. This is achieved with the following URL extraction schema:

```
url_schema = {
        "properties": {
            "url": {"type": "string"},
        },
        "required": ["url"],
    }
```

Web Scraping:

Once the URLs are identified, both the main URL and the extracted sub-URLs are scraped using BeautifulSoup for all possible HTML tags where the text data is present.

This approach ensures that the content from the main URL is thoroughly captured by performing one-step deeper crawling.

Data Extraction and Schema Definition:

The extracted content is processed to match a predefined output schema. The schema format provided to the language model (LLM) is as follows:

```
schema = {
        "properties": {
            "original_id": {"type": "string", "description": "Unique from source"},
            "aug_id": {"type": "string", "description": "Augmented identifier from the context"},
            "country_name": {"type": "string", "description": "Name of the Country"},
              "country_code": {"type": "string", "description": "ISO 3-letter Country Code"},
            "map_coordinates": {
                "type": "object",
                "description": "Geo Point of the region formatted as {'type': 'Point', 'coordinates': [longitude, latitude]}",
                   "properties": {"type": {"type": "string"}, "coordinates": {"type": "array", "items": {"type": "number"}}}},
            },
           "url": {"type": "string", "description": "Url of the website of the source", "format": "uri"},
            "region_name": {"type": "string", "description": "Region Name for a Country according to World Bank Standards"},
            "region_code": {"type": "string", "description": "Region code for a Region according to World Bank Standards"},
              "Project_title": {"type": "string", "description": "A title for this tender/project used as a headline"},
               "Project_description": {"type": "string", "description": "A summary description of the tender/project"},
             "status": {"type": "string", "description": "The current status of the tender/project from the closed tenderStatus codelist"},
           "stages": {"type": "string", "description": "Stages of the tender/project"},
           "date": {"type": "string", "description": "The date on which the information was first recorded or published", "format": "date"},
            "procurementMethod": {"type": "string", "description": "The procedure used to purchase the relevant works, goods or services"},
           "budget": {"type": "number", "description": "The total upper estimated value of the procurement"},
            "currency": {"type": "string", "description": "The currency for each amount specified using the uppercase 3-letter code from ISO4217"},
           "buyer": {"type": "string", "description": "Entity whose budget will be used to pay for related goods, works or services"},
            "sector": {"type": "string", "description": "A high-level categorization of the main sector this procurement process relates to"},
             "subsector": {"type": "string", "description": "A further subdivision of the sector the procurement process belongs to"},
        },
        "required": [
            "original_id", "aug_id", "country_name", "country_code", "map_coordinates", "url",
```

```
            "region_name", "region_code", "title", "description", "status", "stages",
    "date",
            "procurementMethod", "budget", "currency", "buyer", "sector", "subsector"
        ]
    }
```

Instruction to LLM and Data Processing:

To instruct the language model (LLM) for data extraction and formatting, the AsyncChromiumLoader from langchain_community.document_loaders is utilized.

The create_extraction_chain function is employed to generate the formatted output based on the schema. The context provided to the LLM includes the content extracted from the main URL and its sub-URLs.

The data is processed and output in dictionary format, which is subsequently converted into a dataframe for further analysis and utilization.

Tools and Technologies Used:

BeautifulSoup: For web scraping and URL extraction.

AsyncChromiumLoader and create_**extraction**_chain: From langchain_community.document_loaders for instructing the LLM and processing data according to the predefined schema.

Comet_llm: To facilitate and manage the interaction with the LLM.

Why Comet_llm

Comet_llm is utilized to facilitate and manage interactions with language models, enabling efficient and accurate extraction of data from web-scraped content. It supports the automation of data processing tasks, ensuring that the extracted information is formatted according to predefined schemas, thereby enhancing the consistency and reliability of the data outputs.

Part 3: Automation and Continuous Updating

To automate the data scraping and standardization processes, we propose deploying a Streamlit application for user interaction. This application will serve as the interface for users to input URLs of construction and infrastructure project websites. It will be designed to support multiple URLs, allowing users to input them with comma separation for clarity. The Streamlit application will ensure user-friendly interaction and input validation, enhancing the overall user experience and facilitating seamless data collection.

Output (Screenshot)