Automated Translation of Foreign News Headlines through Web Scraping and GPT *(Complete Code: Github Link)*

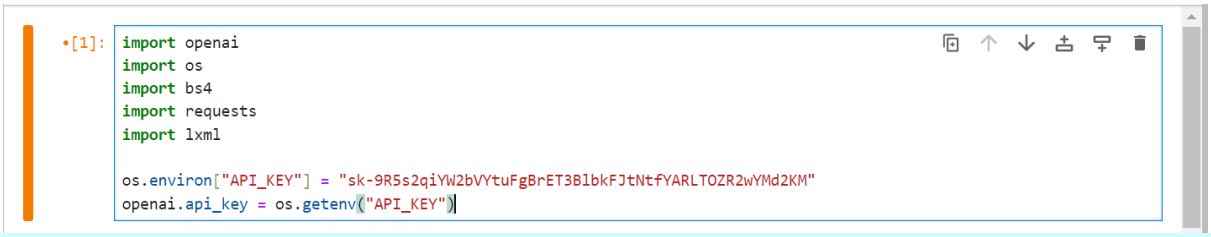
This documentation outlines a project with the goal of integrating web scraping techniques and Open AI's translation capabilities to automatically extract, translate, and summarize content from diverse news websites within a Python application.

The project has two key objectives: the first is to effectively apply web scraping techniques for content extraction using Python libraries, and the second is to translate and summarize foreign language news headlines into concise English summaries.

The project involves a deep dive into web scraping methodologies, Python programming, and leverages Open AI's API for translation and summarization components.

In this project, I used the following Python libraries:

- openai: Used for text generation and AI-powered language tasks.
- os: Used for working with the operating system, including setting environment variables.
- bs4 (BeautifulSoup): Used for web scraping and parsing HTML documents.
- requests: Used for making HTTP requests to fetch web content.
- lxml: A library often used in conjunction with BeautifulSoup for parsing and processing XML and HTML.



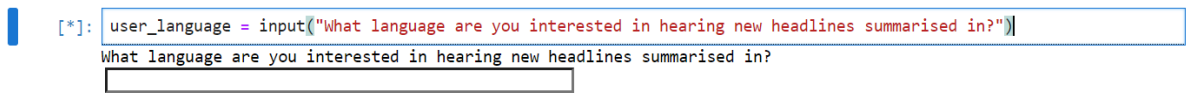
```
[1]: import openai
     import os
     import bs4
     import requests
     import lxml

     os.environ["API_KEY"] = "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"
     openai.api_key = os.getenv("API_KEY")
```

This code segment begins by installing and importing essential libraries, including OpenAI, operating system (os), BeautifulSoup (bs4), requests, and lxml, with a prompt to install any missing ones using "pip install." It then proceeds to set up the OpenAI API key for authentication and configures the OpenAI library to use this key.

```
[2]: news_sites = {
         "chinese": ("https://cn.chinadaily.com.cn", "div.Home_content_Item_Text h1 a"),
         "arabic": ("https://aljazeera.net", "h3.fte-article__title"),
         "english": ("https://www.bbc.co.uk/news", ".gs-c-promo-heading__title")
     }
```

This code defines the news_sites dictionary, which contains website URLs and CSS selectors for different languages. It organizes information for various languages, specifying the website's URL and the CSS selector used to extract headline data.



```
[*]: user_language = input("What language are you interested in hearing new headlines summarised in?")
What language are you interested in hearing new headlines summarised in?
```

This code segment prompts the user to input their preferred language for the summarization of news headlines. The user is asked to specify the language in which they would like to receive summarized news headlines.

```
[*]: def fetch_headlines(language):
         url, tag = news_sites.get(language, (None, None))
         if not url:
             print("language not supported")
             return
         response = requests.get(url)
         soup = bs4.BeautifulSoup(response.text, 'lxml')

         headlines = [h.getText() for h in soup.select(tag)[:10]]
         return headlines

     selected_headlines = fetch_headlines(user_language)
```

The fetch_headlines function retrieves headlines based on the user's language choice from various websites, storing up to 10 of them, and it handles unsupported languages by displaying an appropriate message. The selected headlines are then stored in the selected_headlines variable.

```
[*]: def create_prompt(headlines):
         joined_headlines = "\n".join(headlines)
         prompt = f"Translate the following headlines into English:\n{joined_headlines}"
         return prompt
```

This function generates a prompt by combining multiple headlines and instructs the translation of these headlines into English. It takes a list of headlines, joins them into a single string, and constructs a prompt that requests their translation into English.

```
[*]: prompt = create_prompt(selected_headlines)
     response = openai.Completion.create(
         model="text-davinci-003",
         prompt=prompt,
         temperature=0.1,
         max_tokens=500
     )

     print(response['choices'][0]['text'])
```

This code segment constructs a text prompt using the selected headlines, employs the OpenAI GPT-3 model for text generation, defines specific settings like the temperature, and prints the model's response. It demonstrates the process of using GPT-3 to generate text based on the provided prompt.

```
[*]: user_language = input("What language are you interested in hearing new headlines summarised in?")
What language are you interested in hearing new headlines summarised in?
chinese
```

After selecting 'Chinese' as the preferred language for news headline summarization, the code proceeds to fetch and display the ten latest news headlines from the 'chinadaily.com' website, as per the user's language choice. This follows the initial prompt to input the desired language for summarizing headlines

```
Understand the Golden Value of 5.2% Growth
Why is Chongyang Festival "Elderly Day"?
Financial Departments in Various Places Promote Economic Recovery
Various Places Launch New Consumption Promotion Measures, International Factors Attract Attention
【Qiaoyi Look】Words from Representative Dong Mingzhu to Women
The Conflict between Palestine and Israel Cannot Be Resolved, US Position Causes Outrage
Ministry of Commerce Spokesperson Answers Reporters' Questions on China-Australia WTO Dispute
Hungarian Foreign Minister Calls for EU-China Cooperation, "Decoupling from China" Will Knock Down European Economy
State Administration of Foreign Exchange: It is Expected that China's Current Account Surplus Will Remain at a Reas
onable Scale
Shanghai Launches 12 Measures to Strengthen Innovation in the Consumption Market and Expand Consumption
```

Here is the GitHub link to the complete clean code: ***Github Link***. Please ensure that you create and use your own OpenAI API key.