Internship Assignment 2024 Report: Retrieval-augmented generation (RAG)

Name: Prasan Navin Hegde

Mobile: +91-8867414205

E-mail: nh1.prasan@gmail.com

The task assigned was to develop a Python script capable of searching the internet for information related to Canoo, a publicly traded company listed on NASDAQ (ticker symbol: GOEV). The script should retrieve data from various search engine results and store it in a structured tabular format in a CSV file for further analysis. This report details the steps taken, challenges faced, solutions implemented, techniques used, and recommendations for future improvements.

Steps Taken:

1. The first step involved understanding the requirements of the task, which included searching the internet for information (reference links attached at the end of the assignment).
2. Research was conducted to find suitable libraries and APIs for web scraping and internet search functionalities. The DuckDuckGo Search API and BeautifulSoup library were selected for this purpose.
3. The Python script was developed to perform the following tasks: Utilize the DuckDuckGo Search API to retrieve relevant links for specified user queries. Scrape HTML data from the retrieved links using BeautifulSoup. Extract relevant information such as titles and content from the HTML data. Store the extracted data in a structured format (CSV file).
4. The script was thoroughly tested with different queries to ensure it could retrieve relevant data from diverse sources. Comments and documentation were added to enhance code readability and explain functionality.

Techniques and Libraries Used:

- csv: Used for storing extracted data in a structured tabular format for further analysis.
- requests: Employed to make HTTP requests to desired web pages and retrieve HTML content.
- BeautifulSoup: Used for web scraping to extract data from HTML content retrieved from the links.
- DuckDuckGo Search API: Utilized to search the internet and retrieve relevant links for specified queries.

Challenges Faced and Solutions:

- Website Resistance to Crawling: Some websites resisted webpage crawling, resulting in unresponsiveness to minimal scraping codes. Try/Except blocks were employed to handle such errors during web scraping.
- Dynamic Search Results: DuckDuckGo search results may vary over time due to various factors of SEO (Search Engine Optimization), making it challenging to focus on specific

websites. Flexibility in handling dynamic search results is a requirement and hence focusing on specific websites may not make sense unless it is very important.

- Substandard HTML Coding: Poorly coded websites led to the scraping of unwanted data. Some attention was given to data cleaning techniques to ensure the extraction of relevant information but this step requires more focus for the final output to be reliable.
- Websites layouts and content evolve and change over time, making standard Python programs impractical in the long run. Continuous monitoring and adaptation of the scraping process needs to be done which is not sustainable or practical.

Summary and Recommendations:

- Explore LLM-based Data Cleaning Techniques: Consider leveraging LLM-based data cleaning techniques, such as those offered by paid API services from OpenAI, to extract relevant information from scraped data more effectively.
- Regularly update and maintain the solution to adapt to changes in web page structures, search engine algorithms, and evolving website content.
- Collaborate with domain experts to ensure that the gathered data is accurate, relevant, and actionable for decision-making purposes.

The developed Python script provides a valuable tool for retrieving and analyzing information related to Canoo and its industry. By addressing the challenges encountered and implementing the recommended improvements, the solution can become more reliable, efficient, and sustainable for future use.

Link to GitHub repository

Reference materials used to understand and solve this assignment:

- https://medium.com/@greyboi/ddgsearch-search-duckduckgo-scrape-the-results-in-python-18f5265f1aa6
- https://www.youtube.com/watch?v=DjuXACWYkkU
- https://gist.github.com/hwchase17/69a8cdef9b01760c244324339ab64f0c
- https://python.langchain.com/docs/use_cases/web_scraping
- https://www.ibm.com/docs/en/watsonx-as-a-service?topic=models-retrieval-augmented-generation