### **Documentation for Data Cleaning and Analysis Script**

#### **Overview**

This script performs data extraction, cleaning, and analysis from a given API endpoint. It fetches messages with sources, processes the data, computes similarity scores between responses and their corresponding sources using spaCy's NLP model, and outputs a JSON structure with IDs and links based on a similarity threshold.

#### **Prerequisites**

* Libraries Required:
  + json: For handling JSON data.
  + requests: For making HTTP requests.
  + pandas: For data manipulation and analysis.
  + spacy: For natural language processing.
* spaCy Model:
  + The script uses the en\_core\_web\_lg model from spaCy.
  + To download the model, run: !python -m spacy download en\_core\_web\_lg

#### **Step-by-Step Code Explanation**

1. Import the Libraries
   * Import necessary libraries for JSON handling, HTTP requests, data manipulation, and natural language processing.
2. Load the spaCy Model
   * Load the en\_core\_web\_lg model from spaCy for NLP tasks.
3. Fetch Data from the API Endpoint
   * Make an HTTP GET request to fetch data from the specified API endpoint and parse the response into JSON format.
4. Initialize Lists for Data Cleaning and Analysis
   * Initialize several lists to store various elements extracted from the JSON data such as IDs, responses, sources, and links.
5. Extract Data from JSON and Populate Lists
   * Iterate over the JSON data to extract and append id, response, and source into their respective lists.
6. Create a DataFrame from Extracted Data
   * Use pandas to create a DataFrame from the extracted data, organizing it into a tabular format.
7. Compute Similarity and Filter Results
   * For each response, compute the similarity between the response and each of its sources using spaCy's NLP model. If the similarity score is above a threshold (0.45), append the relevant IDs and links to the lists.
8. Convert Filtered Results into JSON Format
   * Combine the filtered results into a list of dictionaries, each containing an id and a link. Convert this list into a JSON string with pretty-print formatting.
9. Print the JSON Output
   * Output the JSON string containing the filtered results.

#### **Detailed Description**

Importing Libraries:

* The script starts by importing necessary libraries for JSON handling, HTTP requests, data manipulation, and natural language processing.

Loading spaCy Model:

* The en\_core\_web\_lg model is loaded to perform NLP tasks such as calculating similarity between texts.

Fetching Data:

* An HTTP GET request is made to fetch data from a specified API endpoint, and the response is parsed into JSON format.

Initializing Lists:

* Several lists are initialized to store various elements extracted from the JSON data such as IDs, responses, sources, and links.

Extracting Data:

* The JSON data is iterated to extract and append id, response, and source into their respective lists.

Creating DataFrame:

* A DataFrame is created using pandas to organize the extracted data into a tabular format.

Computing Similarity:

* For each response, the script computes the similarity between the response and each of its sources. If the similarity score is above a threshold (0.45), the relevant IDs and links are appended to the lists.

Converting to JSON:

* The filtered results are combined into a list of dictionaries, each containing an id and a link. This list is then converted into a JSON string with pretty-print formatting.