CS6111 - Project1

Names and UNIs:

Charlie Shen (cs4206) and Zhehong Fu (zf2307)

List of files

Name	Usage
`main.py`	Main python program
`proj1-stop.txt`	Stop words needed to be eliminated to refine the results
`requirements.txt`	Python packages to run the program
`transcript.pdf`	The results of 3 test cases

How to run

First, run the following to install the python packages under proj1 directory:

```
$ pip install -r requirements.txt
```

Then run the following to start the program:

```
$ python3 main.py <google api key> <search engine id> <precision>
<query>
```

Note: if `<query>` has multiple words, be sure to put them between quotes (e.g. `"per se"`).

Credentials

- Google API Key : AIzaSyCqNuAyVdEZHSXjsenXecoF3doj9CmIzzE
- Engine ID: a3e855684755e4cb8

Internal design

1. The program incorporates specific functions for displaying results, namely `print_initial_info`, `print_feedback_achieve`, `print_precision_smaller_than_tar` and `print_precision_equal_zero`. These functions ensure the output is consistent with the examples provided in the sample tests.
2. To retrieve customized search results from Google, the `google_custom_search` function is employed. This function requires the Google API Key, Engine ID, and a search query as inputs. Upon receiving search results, the program evaluates `results['searchInformation']['totalResults']` as detailed on the Google JSON website. If the initial search yields fewer than 10 total results, the program will terminate prematurely.
3. Within the main function, the program continuously initiates searches to compare the obtained precision against the predefined target precision. The process terminates under two conditions:
   - If the precision at the 10th result is zero, the program ends, displaying a message through `print_precision_equal_zero`.
   - If the acquired precision meets or exceeds the target precision, the program concludes with a `print_feedback_achieve` message, indicating successful achievement.

Additional Notes:

1. Handling of Non-HTML Documents: The program utilizes the `fileFormat` key from the search results to identify non-HTML documents, as specified on the Google JSON website. Absence of the `fileFormat` key indicates the document is in HTML format.
2. Interaction with the `Enter` Key: Pressing the `Enter` key during program execution is interpreted as a 'no' response, facilitating user interaction and decision-making processes.

Query-modification method

Our method for refining search queries is both effective and uncomplicated. It hinges on the principle of selecting the most frequently occurring keywords from relevant search results to craft a new query. Here’s how we implement it:

1. We begin by consolidating the title and snippet from each pertinent result to form a potential pool of keywords. This composite string is then standardized by converting it to lowercase and stripping away all non-alphanumeric characters, barring spaces.
2. Crucially, we eliminate commonly used words that carry little informational value, known as stopwords. This step is vital for focusing on the most meaningful terms.
3. The processed results are organized into lists of individual keywords. These lists are collectively stored in a master list named “contents”.
4. To discern the most valuable keywords for our query, we tally the frequency of each word across the "contents" list, employing a dictionary termed “freq”. This dictionary maps each keyword to the number of its occurrences, aggregating data from all relevant results.
5. We then order the dictionary by frequency and select the top two keywords not previously included in the query. These are appended to the query in descending order of frequency.

This approach has proven its efficacy, consistently delivering a 100% precision rate in a single iteration, provided the initial precision exceeds zero.