Web Search Engine Comparison

This exercise is about comparing the search results from Google versus different search engines. Many search engine comparison studies have been done. All of them use samples of data, some small and some large, so no definitive conclusions can be drawn. But it is always instructive to see how two search engines match up, even on a small data set.

The process you will follow is to issue a set of queries and to evaluate how closely the results of the two search engines compare. You will compare the results from the search engine that you are assigned to with the results from Google (provided by us on the class website).

To begin, the class is divided into four groups. Students are pre-assigned according to their USC ID number, as given in the table below.

Note: Please stick with the assigned dataset and search engine according to your ID number. PLEASE don't work on another dataset and later ask for an exception.

USC ID ends with	Query Data Set	Google Reference Dataset	Assigned Search Engine
00~24	100QueriesSet1	Google_Result1.json	Bing
25~49	100QueriesSet2	Google_Result2.json	Yahoo!
50~74	100QueriesSet3	Google_Result3.json	Ask
75~99	100QueriesSet4	Google_Result4.json	DuckDuckGo

THE QUERIES

The queries will be given to you in a text file, one query per line. Each file contains 100 queries. These are actual queries extracted from query log files of several search engines. Here is a sample of some of the queries:

```
The European Union includes how many countries
What are Mia Hamms accomplishments
Which form of government is still in place in Greece
When was the canal de panama built
What color is the black box on commercial airplanes
```

Note: Some of the queries will include misspellings; you should not alter the queries in any way as this accurately reflects the type of query data that search engines have to deal with

REFERENCE GOOGLE DATASET

A Google Reference JSON[1] file is given which contains the Google results for each of the queries in your dataset. The JSON file is structured in the form of a query as the key and a list of 10 results as the value for that key (each a particular URL representing a result). The Google

[1] JSON, JavaScript object Notation is a file format used to transmit data objects consisting of key-value pairs. It is programming language independent. https://www.softwaretestinghelp.com/json-tutorial/

results for a specific query are ordered as they were returned by Google. Namely the *1st* element in the list represents the top result that was scraped from Google, the *2nd* element represents the second result, and so on. Example:


```
{
"A two dollar bill from 1953 is worth what": [
       "http://www.antiquemoney.com/old-two-dollar-bill-value-price-guide/two-dollar-
bank-notes-pictures-prices-history/prices-for-two-dollar-1953-legal-tenders/",
       "https://oldcurrencyvalues.com/1953_red_seal_two_dollar/",
       "https://www.silverrecyclers.com/blog/1953-2-dollar-bill.aspx",
       "https://www.ebay.com/b/1953-A-2-Dollar-Bill/40033/bn_7023293545",
       "https://www.ebay.com/b/1953-2-US-Federal-Reserve-Small-
Notes/40029/bn_71222817",
       "https://coinsite.com/why-the-1953-2-dollar-bill-has-a-red-seal/",
       "https://hobbylark.com/collecting/Value-of-Two-Dollar-Bills",
       "https://www.quora.com/What-is-the-value-of-a-2-dollar-bill-from-1953",
       "https://www.reference.com/hobbies-games/1953-2-bill-worth-c778780b24b9eb8a",
       "https://treasurepursuits.com/1953-2-dollar-bill-value-whats-it-worth/"
       ]
}
```


DETERMINING OVERLAP AND CORRELATION

Overlap: Since the Google results are taken as our baseline, it will be interesting to see how many identical results are returned by your assigned search engine, regardless of their position. Assuming Google's results are the standard of relevance, the percentage of identical results will act as a measure of the quality of your assigned search engine.

Each of the queries in your dataset should be run on your assigned search engine. You should capture the top ten results. Only the resulting URL is required. For each of the top ten results for each query you should compute an overlap score between our reference Google answer dataset and your scraped results. The output format is described ahead.

Note: If you get less than 10 URLs for a particular query, you can just use those results to compare against Google results. For example: if a query gets 6 results from a search engine, just use those 6 results to compare against 10 results of Google reference dataset and produce statistics for that particular query.

Note: For a given query, if the Google result has 10 URLs, but the other search engine has fewer results (e.g. 8), and there are 5 overlapping URLs, the percent overlap would be 5/10

Correlation: In statistics, **Spearman's rank correlation coefficient** or **Spearman's rho**, is a measure of the statistical dependence between the rankings of two variables. It assesses how well the relationship between two variables can be described. Intuitively, the Spearman correlation between two variables will be high when observations have a similar rank, and low when observations have a dissimilar rank.

The rank coefficient r_s can be computed using the formula

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}.$$

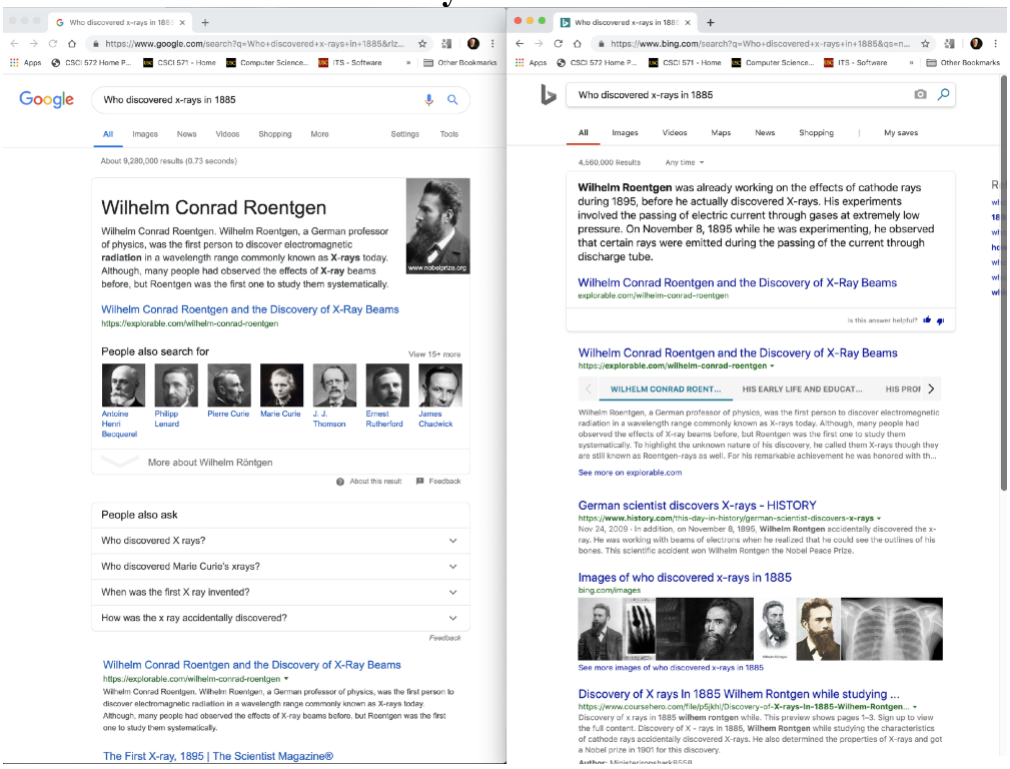
where,

- d_i is the difference in the two rankings, and
- n is the number of observations

Note: The formula above when applied to search results yields a somewhat modified set of values that can be greater than one or less than minus one. However the sign of the Spearman correlation indicates the direction of association between the two rank variables. If the rank results of one search engine is near the rank of the other, then the Spearman correlation value is positive. If the rank of one is dissimilar to the rank of the other, then the Spearman correlation value will be negative.

Note: In the event that your search engine account enables personalized search, please turn this off before performing your tests.

Example1.1: "Who discovered x-rays in 1885"



GOOGLE RESULTS

1. https://explorable.com/wilhelm-conrad-roentgen

2. https://www.the-scientist.com/foundations/the-first-x-ray-1895-42279

3. https://www.bl.uk/learning/cult/bodies/xray/roentgen.html

4. https://en.wikipedia.org/wiki/Wilhelm_R%C3%B6ntgen

5. https://www.wired.com/2010/11/1108roentgen-stumbles-x-ray/

6. https://www.history.com/this-day-in-history/german-scientist-discovers-x-rays

7. https://www.aps.org/publications/apsnews/200111/history.cfm

8. https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/Introduction/history.htm

9. https://www.dw.com/en/x-ray-vision-an-accidental-discovery-that-revolutionized-medicine/a-18833060

10. http://www.slac.stanford.edu/pubs/beamline/25/2/25-2-assmus.pdf

RESULTS FROM ANOTHER SEARCH ENGINE

1. https://explorable.com/wilhelm-conrad-roentgen

2. https://www.history.com/this-day-in-history/german-scientist-discovers-x-rays

3. https://www.coursehero.com/file/p5jkhl/Discovery-of-X-rays-In-1885-Wilhem-Rontgen-while-studying-the-characteristics/

4. http://www.nde-ed.org/EducationResources/HighSchool/Radiography/discoveryxrays.htm

5. https://www.answers.com/Q/Who_discovered_x-rays

6. https://www.aps.org/publications/apsnews/200111/history.cfm

7. https://www.answers.com/Q/Who_discovered_x-rays

8. https://www.coursehero.com/file/p5jkhl/Discovery-of-X-rays-In-1885-Wilhem-Rontgen-while-studying-the-characteristics/

9. https://www.wired.com/2010/11/1108roentgen-stumbles-x-ray/

10. http://time.com/3649842/x-ray/

RANK MATCHES FROM GOOGLE AND ANOTHER SEARCH ENGINE

1 AND 1

5 AND 9

6 AND 2

7 AND 6

We are now ready to compute Spearman's rank correlation coefficient.

Rank Google	**Rank Other Srch Engine**	d_i	d_{i2}
1	**1**	**0**	**0**
5	**9**	**-4**	**16**
6	**2**	**4**	**16**
7	**6**	**1**	**1**

The sum of d_{i2} = 49. The value of n = 4. Substituting into the equation

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

1 – ((6 * 33) / (4 * 15)) = 1 – (3.3) = -2.30

Even though we have five overlapping results (50% overlap), their positions in the search result list produce a negative Spearman coefficient indicating that the overlapping results are uncorrelated. Clearly the two search engines are using different algorithms for weighting and ranking the documents they determine are most relevant to the query. Moreover their algorithms are emphasizing different ranking features.

Note: the value of ***n*** in the equation above refers to the number of URL matches (in this case, five) and does not refer to the original number of results (in this case, ten).

Note: If *n=1* (which means only one paired match), we deal with it in a different way:
1. if Rank in your result = Rank in Google result → *rho=1*
2. if Rank in your result ≠ Rank in Google result → *rho=0*

TASKS
Task1: Scraping results from your assigned search engine

In this task you need to develop a script (computer program) that could scrape the top 10 results from your assigned search engine. You may use any language of your choice. **Always incorporate random delay between 10 to 100 seconds while scraping multiple queries, else you may be blocked off by the search engine and they may not allow you to scrape results for several hours.** For reference:

- https://pypi.org/project/beautifulsoup4, a python library for parsing HTML documents
- URLs for the search engines:
    - Bing: http://www.bing.com/search?q=
    - Yahoo!: http://www.search.yahoo.com/search?p=
    - Ask: http://www.ask.com/web?q=
    - DuckDuckGo: https://www.duckduckgo.com/?q=

  For each URL, you can add your query string after q=

- Selectors for various search engines, you grab links by looking for href in these selectors:
    - Bing: [“li”, attrs = {“class” : “b_algo”}]
    - Yahoo!: ["a", attrs = {"class" : "ac-algo fz-l ac-21th lh-24"}]
    - Ask: ["div", attrs = {"class" : "PartialSearchResults-item-title"}]
    - DuckDuckGo: ["a", attrs = {"class" : "result__a"}]

By executing this task you need to generate a JSON file which will store your results in the JSON format described above and repeated here.

```
{
      Query1: [Result1, Result2, Result3, Result4, Result5, Result6, Result7,
Result8, Result9, Result10],
      Query2: [Result1, Result2, Result3, Result4, Result5, Result6, Result7,
Result8, Result9, Result10],
      ….
      Query100: [Result1, Result2, Result3, Result4, Result5, Result6, Result7,
Result8, Result9, Result10]
}
```

Here Result1 is the top result for that particular query.

Task2: Determining the Percent Overlap and the Spearman Coefficient

For this task, you need to use the JSON file that you generated in Task 1 and the Google reference dataset which is provided by us and compare the results as shown in the **Determining Correlation** section above. The output should be a **CSV** file with the following information:

1. Use the JSON file that you generated in Task 1 and do the following steps on each query:
2. Determine the URLs that match with the given reference Google dataset, and their position in the search engine result list
3. Compute the percent of overlap. In Example1.1, above the percent overlap is 5/10 or 50%.
4. Compute the Spearman correlation coefficient. In above Example1.1, the coefficient is -1.45.
5. Once you run all of the queries, collect all of the top ten URLs and compute the statistics, as shown in the following example:

```
Queries, Number of Overlapping Results, Percent Overlap, Spearman Coefficient
Query 1, 5, 50.0, -1.45
Query 2, 6, 60.0, -0.4
...
...
...
Averages, 5.2, 55.7, -0.5
```

Note: The above example is a table with four columns, rows containing results for each of the queries, and averages for each of the columns. Of course the actual values above are only for demonstration purposes.

Points to note:

- Always incorporate a delay while scraping. We recommend that you use a random delay with a range of 10 to 100 seconds.
- You will likely be blocked off from the search engine if you do not implement some delay in your code.
- You should ignore the People Also Ask boxes and any carousels that may be included in the results.

SUBMISSION INSTRUCTIONS

Please place your homework in your Google Drive CSCI572 folder that is shared with your grader, in the subfolder named hw1. You need to submit:

- JSON file generated in Task 1 while scraping your assigned search engine, call it `hw1.json`

- CSV file of final results after determining relevance between your assigned search engine and Google reference dataset provided by us, call it `hw1.csv`. **Note**: you need not format the numbers.
- TXT file stating why the assigned search engine performed either better/worse/same as Google, call it `hw1.txt`. For the txt file, we are just looking for a paragraph which states how much is your assigned search engine similar to Google based on the Spearman coefficients and percent overlap.

SAMPLE SCRAPING PROGRAM IN PYTHON

Here is a program you can use to help you get started

```
from bs4 import BeautifulSoup
from time import sleep
import requests
from random import randint
from html.parser import HTMLParser

USER_AGENT = {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100
Safari/537.36'}

class SearchEngine:
    @staticmethod
    def search(query, sleep=True):
        if sleep: # Prevents loading too many pages too soon
            time.sleep(randint(10, 100))
        temp_url = '+'.join(query.split()) #for adding + between words for
the query
        url = 'SEARCHING_URL' + temp_url
        soup = BeautifulSoup(requests.get(url, headers=USER_AGENT).text,
"html.parser")
        new_results = SearchEngine.scrape_search_result(soup)
        return new_results

    @staticmethod
    def scrape_search_result(soup):
        raw_results = soup.find_all("SEARCH SELECTOR")
        results = []
        #implement a check to get only 10 results and also check that URLs
must not be duplicated
        for result in raw_results:
            link = result.find('a').get('href')
            results.append(link)
        return results

#############Driver code############
SearchEngine.search("QUERY")
###################################
```

FAQs

1. What do I need to run Python on my Windows/Mac machine?
   You can refer to the documentation for setup:
   https://docs.python.org/3.6/using/index.html
   We encourage you to use Python 3.6. You can find many tutorials on Google.
2. Given that Python is installed what lines of the sample program do I have to modify to get it to work on a specific search engine?
   In the reference code, you need to:
   - Supply in query variable
   - Change SEARCHING_URL and SEARCH SELECTOR as per the search engine that is assigned to you
   - Implement the code that extracts only top 10 URLs and make sure that none of them is repeated
   - Implement the main function
3. What to do if the query does not produce ten results.
   You can modify the URLs to get 30 results on single page:
   - For bing use count=30 - http://www.bing.com/search?q=test&count=30
   - For yahoo use n=30 - https://search.yahoo.com/search?p=test&n=30
   - For Ask there does not appear to be a parameter which could produce n results on single page, so instead you can update the URL in such a manner which increments page number
   - For Ask use page=2 - https://www.ask.com/web?q=testn&page=2
   - If, after trying the above hints you are unable to get 10 results for a particular query, you can just use those results to compare against Google results. For example: if a query gets 6 results from a search engine, just use those 6 results to compare against 10 results of Google reference dataset and produce statistics for that particular query.
4. Two URLs that differ only in the scheme (http versus https) can be treated as the same.
5. Metrics for similar URLs:
   a. As browsers default to www when no host name is provided, so xyz.com is identical to www.xyz.com
   b. URLs that only differ in the scheme (http or https) are identical
   c. www.xyz.com and www.xyz.com/ - You need to remove slash(/) at the end of URL
   d. Convert all URLs to lowercase, www.xyz.com/ab and www.xyz.com/Ab.
6. Value of rho:
   a. If no overlap, rho = 0
   b. If only one result matches:
      i. if Rank in your result = Rank in Google result → rho=1
      ii. if Rank in your result ≠ Rank in Google result → rho=0
7. Rho value may be extremely large/small

    a. In some cases the value may >100 or <-100. We just accept the value and calculate the average like nothing happens
    b. How to calculate average rho? We calculate rho for each query and sum them up. Then we get the average
8. Save order as JSON
    a. You can save the dictionary in python as JSON directly by importing json library calling json.dump(args)