README

Running the Entire Program:

Run the main.py file which clearly states all the inputs needed and the format.

A sample query - “Phelps” is run as an example.

Output:

```
(base) C:\Users\Nitin\Desktop\IR_Assignment2>python main.py
Give your query to get results/press EXIT to end the loop :
Phelps

Root set : [50]

Base set : [21, 40, 55, 73]

Edges : [(21, 50), (40, 50), (50, 73), (55, 50)]

Converged @ 30th Iteration
Node
21    5.773503e-01
40    5.773503e-01
55    5.773503e-01
50    4.023653e-08
73    0.000000e+00
Name: Hub Score, dtype: float64

Node
50    1.000000e+00
73    6.969172e-08
21    0.000000e+00
40    0.000000e+00
55    0.000000e+00
Name: Authority Score, dtype: float64

Info retrieved in 0.1614229679107666 sec
```

Using the “retrieveResults” function:

Import the function retrieveResults from get_results Module. Input any query to get the Node IDs that contain this query irrespective of its case.

An example of Query - “Sports” is given as an example input.

```
from get_results import retrieveResults

print(retrieveResults("Sports"))
```

Output:

A list of Nodes where “Sports” is present in the text.

```
(base) C:\Users\Nitin\Desktop\IR_Assignment2>python temp.py
[50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74]
```

Using the “get_baseset” function:

Import the function get_baseset from get_results Module. Input the list of nodes to get their base set as a list of Nodes and a list of tuples that contain their corresponding directed edges.

A list of 7,10,18 nodes is given as an example input:

```
from get_results import get_baseset

baseset, edges = get_baseset([7,10,18])

print("Base set :", baseset)
print()
print("Edges :", edges)
```

Output:

```
(base) C:\Users\Nitin\Desktop\IR_Assignment2>python temp.py
Base set : [0, 34, 3, 70, 71, 75, 81, 86, 61]

Edges : [(18, 86), (10, 75), (61, 18), (70, 18), (18, 34), (3, 10), (71, 18), (18, 71), (18, 61), (86, 18), (7, 81), (10, 0), (10, 61)]
```

Using the “error” function:

Import the function error from the scores Module. Input 2 Dictionaries that you wish to find the average deviations from each other. Note: These 2 Dictionaries need to have all the Keys in common. Otherwise, results in an error.

```
from scores import error

print(error({1:21,2:32,4:54},{1:24,2:33,4:52}))
```

Output:

```
(base) C:\Users\Nitin\Desktop\IR_Assignment2>python temp.py
2.0
```

Using the “scores” function:

Import the function scores from the scores Module. Input a list of tuples of 2 integers that signify a directed edge from the first int node in the tuple to the other and a sorted list of Nodes that we’re dealing with. The Function returns 2 dictionaries that have hub scores and authority scores with Node IDs as Key and respective scores as Values.

An example input of [(21,50),(40,50),(50,73),(55,50)],[21,40,50,55,73] are given.

```
from scores import scores

hub_score, authority_score = scores([(21,50),(40,50),(50,73),(55,50)],[21,40,50,55,73])
print("Hub Score :",hub_score)
print("authority score :",authority_score)
```

Output:

```
(base) C:\Users\Nitin\Desktop\IR_Assignment2>python temp.py
Converged @ 30th Iteration
Hub Score : {21: 0.5773502691896253, 40: 0.5773502691896253, 50: 4.023653294216942e-08, 55: 0.5773502691896253, 73: 0.0}
authority score : {21: 0.0, 40: 0.0, 50: 0.9999999999999976, 55: 0.0, 73: 6.969171937625616e-08}
```

Plot between the runtime of HITS and the number of edges: