Recuperación de Información en la Web

Alumno: Gallozo, Luis Angel 156308

1) **Escriba un script Python que reciba como parámetro una URL y descargue la página HTML correspondiente. Luego, extraiga de la misma los enlaces y los muestre por consola.**

   El script se ejecuta con “python app.py”.

2) **Modifique su programa anterior para implementar un crawler básico de acuerdo al algoritmo presentado en la siguiente figura**

```
1: push(todo_list,initial_set_of_urls)
2: while todo_list[0] ≠ Ø do
3:    page ← fetch_page(todo_list[0])
4:    if page downloaded then
5:       links ← parse(page)
6:       for all l in links do
7:          if l in done_list then
8:             push(todo_list[0].outlinks,done_list[l].id)
9:          else if l in todo_list then
10:            push(todo_list[0].outlinks,todo_list[l].id)
11:         else if l pass our filter then
12:            push(todo_list,l)
13:            todo_list[l].id = no. of url's
14:            push(todo_list[0].outlinks,todo_list[l].id)
15:         end if
16:      end for
17:   end if
18: end while
```

Luego, realice una pequeña recolección con los siguientes parámetros:

a) Conjunto semilla: los 20 primeros sitios del ranking de Netcraft[1]
b) Cantidad Máxima de Páginas por sitio: [20-50]
c) Profundidad Lógica Máxima: 3
d) Profundidad Física Máxima: 3

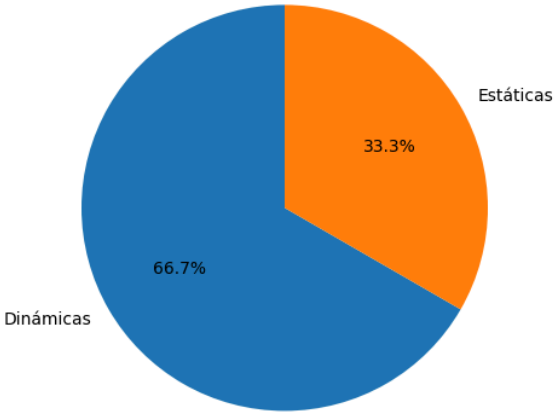
Luego de la descarga, arme el grafo correspondiente. Grafique utilizando la librería pyvis[2]

[1] https://trends.netcraft.com/topsites
[2] https://pyvis.readthedocs.io/en/latest/

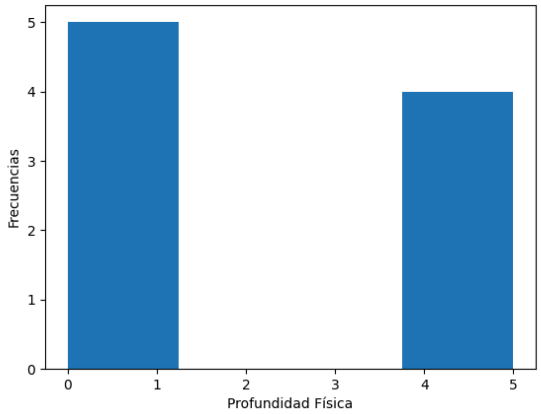
El script se ejecuta con “python app.py”. Los gráficos se generan en la carpeta local. Recordar que en el archivo constantes.py estan definidos:

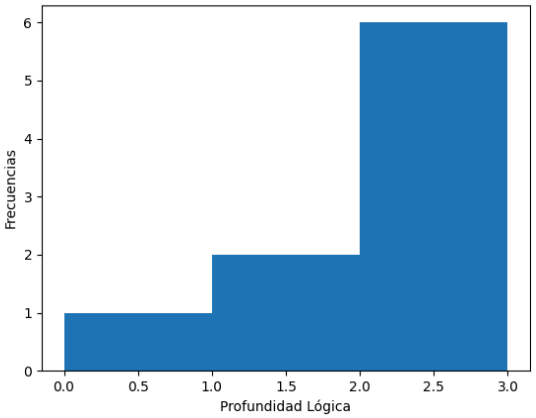
- LIMIT_SEEDS = Límite de url semillas.
- LIMIT_PAGES= Límite de páginas a crawlear.
- LIMIT_PAGES_SITE=Límite de páginas por sitio a crawlear.
- LIMIT_DEPTH_LOGIC = Límite de profundidad lógica.
- LIMIT_DEPTH_PHYSICAL = Límite de profundidad física.
- URL_SEEDS = Url de donde se obtendrán las páginas semilla.

3) **Realice un crawling de la página principal de Amazon.com (solamente páginas dentro del dominio). Al finalizar, analice la distribución de páginas dinámicas y estáticas y la distribución de frecuencias por profundidad lógica y física.**
El script se ejecuta con “python app.py”. Generan los gráficos de distribución en la carpeta local. Recordar que en el archivo constantes.py estan definidos:
   - LIMIT_PAGES= Límite de páginas a crawlear.
   - URL_Amazon = Url sitio web de Amazon.



Como vemos en Amazon.com, aproximadamente el 66.7% de las páginas son dinámicas, lo que significa que se generan en tiempo real en respuesta a la solicitud del usuario. En contraste, el 33.3% de las páginas son estáticas, presentando contenido que no varía con frecuencia y es el mismo para todos los usuarios, como páginas informativas, de políticas y de ayuda.





Profundidad Física:

- Profundidad 1 (5 páginas): La mayoría de las páginas se encuentran muy cerca del dominio raíz en términos de rutas URL. Esto incluye la página principal y otras páginas críticas accesibles directamente.
- Profundidad 5 (4 páginas): Estas páginas se encuentran más profundamente anidadas en la estructura física del URL. Pueden ser páginas con contenido que requiera varios niveles de navegación desde la página principal.

Profundidad Lógica

- Profundidad 1 (1 página): La página principal de Amazon.
- Profundidad 2 (2 páginas): Podrían ser páginas que están directamente enlazadas desde la página principal.
- Profundidad 6 (3 páginas): Estas páginas pueden estar más profundamente anidadas en la estructura del sitio.

4) Suponga que se han recuperado un conjunto de páginas de un pequeño repositorio de sólo seis utilizando el modelo de espacio vectorial. Dado un query Q se produjo la salida de la siguiente con los valores de sim(Q, Di).

Pos	Doc	Score
1	E	4.9734
2	C	4.8173
3	A	2.5617
4	B	2.0110
5	D	0.8937
6	F	0.0000

Las páginas se encuentran vinculadas de acuerdo al siguiente grafo:

a) **Calcule los valores de PageRank de las páginas utilizando como factor de damp 0.15 y 0.5. Pruebe iterando 2, 5 y 10 veces.**
b) **Use los valores de PageRank para re-rankear la salida de la búsqueda interpolando los valores (controlado por un parámetro *α*). ¿Se altera el ranking? ¿En qué caso? Comente los resultados.**
Se adjunta link de la planilla excel en el archivo zipeado.

Factor Damping: 0.15

Ranking (Step 2)	
Doc	Score
E	1,591186667
C	1,5681275
A	0,904135
B	0,7129666667
D	0,3910475
F	0,1096666667

Ranking (Step 5)	
Doc	Score
E	1,591186667
C	1,568289484
A	0,904424625
B	0,7126598698
D	0,3912094844
F	0,1093598698

Ranking (Step 10)	
Doc	Score
E	1,591186667
C	1,568289619
A	0,9044237056
B	0,7126601946
D	0,3912096193
F	0,1093601946

Ranking Original		
Doc	Score	score normalizado
E	4,9734	1
C	4,8173	0,9686130213
A	2,5617	0,5150802268
B	2,011	0,4043511481
D	0,8937	0,1796959826
F	0	0

Ranking (Step 10) Score Normalizado	
Doc	Score
C	0,4136835257
E	0,3991666667
A	0,2904377736
B	0,230665539
D	0,1770084141
F	0,1093601946

Comparando con el ranking original, para todas las iteraciones dieron el mismo ranking. Se decidió normalizar utilizando el máximo los valores de similitud para ver cómo afecta el valor de alfa, obteniendo un cambio en los 2 primeros lugares de (E y C). Al parecer el valor de similitud estaba afectando al score final debido a que estaba en distinta escala, haciendo que el documento E tenga mejor valoración debido a la similitud que aporta en la fórmula.

Factor Damping: 0.5

Ranking (Step 2)	
Doc	Score
C	1,586162222
E	1,550353333
A	0,9143433333
B	0,7102444444
D	0,4090822222
F	0,1069444444

Ranking (Step 5)	
Doc	Score
C	1,59358892
E	1,550353333
A	0,9213649383
B	0,6993069444
D	0,4165089198
F	0,09600694444

Ranking (Step 10)	
Doc	Score
C	1,593933999
E	1,550353333
A	0,9201721656
B	0,6995582519
D	0,4168539986
F	0,09625825189

Ranking Original		
Doc	Score	score normalizado
E	4,9734	1
C	4,8173	0,9686130213
A	2,5617	0,5150802268
B	2,011	0,4043511481
D	0,8937	0,1796959826
F	0	0

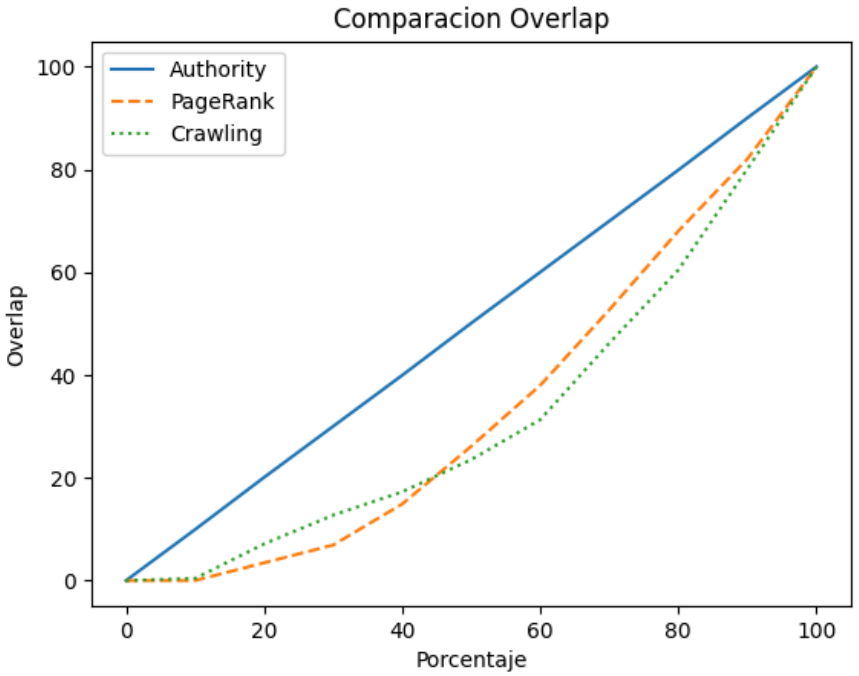
Ranking (Step 10) Score Normalizado	
Doc	Score
C	0,439327905
E	0,3583333333
A	0,3061862337
B	0,2175635963
D	0,2026527934
F	0,09625825189

Comparando con el ranking original, para todas las iteraciones las posiciones de los documentos C y E. Esto ocurre debido a que el valor del factor dumping es más alto, haciendo que cada documento herede solo la mitad del peso de las páginas que le apuntan. Ese ranking se mantiene al realizar el experimento con el score normalizado.

5) **Usando su crawler realice una recolección de 500 páginas. Luego, calcule para cada una los valores de PageRank y Authorities (HITS) utilizando la librería NetworkX[3] Simule un crawling siguiendo el orden de acuerdo al valor de PageRank de las páginas (de mayor a menor). Grafique la evolución del porcentaje de overlap respecto del orden por valor de Auth para ambas estrategias de crawling. Explique su resultado.**

El script se ejecuta con “python app.py”. El gráfico se genera en la carpeta local. Recordar que en el archivo constantes.py estan definidos:

- LIMIT_SEEDS = Límite de url semillas.
- LIMIT_PAGES= Límite de páginas a crawlear.
- LIMIT_PAGES_SITE=Límite de páginas por sitio a crawlear.
- LIMIT_DEPTH_LOGIC = Límite de profundidad lógica.
- LIMIT_DEPTH_PHYSICAL = Límite de profundidad física.
- URL_SEEDS = Url de donde se obtendrán las páginas semilla.



La gráfica muestra que el orden de crawling se aleja de Authority mientras que el orden basado en PageRank se acerca más a Authority. La similitud (overlap) entre PageRank y Authority aumenta debido a que las páginas con alto PageRank tienden a ser también altas en Authorities, por lo que el conjunto de páginas recolectadas refleja de manera natural las autoridades de la web. Sabiendo estos resultados, es más eficiente crawlear antes las páginas de mayor reputación porque esas aparecen con más probabilidad en los primeros lugares del ranking.

[3] https://networkx.github.io/