



Recuperación de Información en la Web

Fecha entrega: 21/06/2019

Bibliografía sugerida: RFC2616[2], Croft [1] Capítulos 3 y 4, MAN [3] Capítulos 19, 20, 21.

1. ¿Qué es el standard de exclusión de robots? ¿Para qué se utiliza? Brinde un ejemplo del archivo robot.txt y explique su contenido.
2. Explique cómo debería tratar un *crawler* los diferentes códigos HTTP.
3. ¿Cómo se determinan las páginas dinámicas y las estáticas? Escriba un fragmento de código de una página dinámica (que funcione en su servidor HTTP) y explique cómo se pueden pasar los parámetros?
4. ¿Qué problemas hay con el servicio DNS en el crawling? ¿Cómo se resuelven?
5. ¿Qué especifican los META tags NOINDEX y NOFOLLOW? ¿Dónde se utilizan?
6. 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.
7. Modifique su programa anterior para implementar un *crawler* básico de acuerdo al algoritmo de la Figura . Realice una pequeña recolección (no más de 50 páginas) y con la información de enlaces arme el grafo correspondiente. Grafique utilizando la librería GraphViz¹.
8. 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 Tabla con los valores de $sim(Q, D_i)$. Las páginas se encuentran vinculadas de acuerdo al grafo de la Figura .
 - 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-ranquear la salida de la búsqueda interpolando los valores (controlado por un parámetro α). ¿Se altera el ranking? ¿En qué caso? Comente los resultados.
9. Usando su *crawler* realice una recolección de 500 páginas. Luego, calcule para cada una los valores de PageRank y Authorities (HITS) utilizando la librería NetworkX². 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. Establezca un valor de corte para la lista de autoridades (por ejemplo, 50%). Explique su resultado.

Referencias

- [1] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.
- [2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Rfc2616 - hypertext transfer protocol – http/1.1. Technical report, United States, 1999.
- [3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

¹<http://graphviz.org/>

²<https://networkx.github.io/>

```

1: push(todo_list,initial_set_of_urls)
2: while todo_list[0] ≠ ∅ 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
  
```

Figura 1: Pseudocódigo crawler básico.

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

Cuadro 1: Ranking de la búsqueda por Q

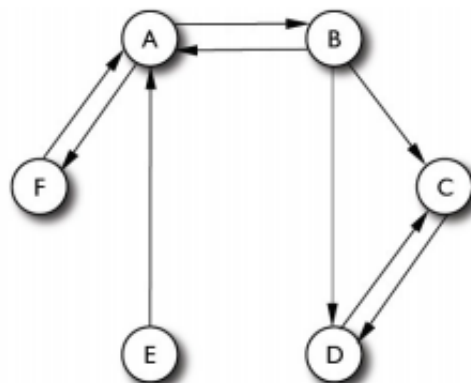


Figura 2: Grafo ejemplo