Ficha 02

Motores de Búsqueda: El Modelo PageRank

1.] Introducción[1].

Hemos presentado en una ficha anterior el problema de la Recuperación de Información, y al menos un par de modelos de base matemática (el booleano y el vectorial) para el diseño de motores de búsqueda que resuelvan ese problema. El modelo vectorial, presentado en 1975 por *Gerald Salton*, ha sido uno de los más utilizados en numerosas aplicaciones que requieren servicios de búsqueda.

Pero en los últimos tiempos se ha impuesto el modelo del motor de búsqueda de *Google*, basado en la idea del *Ranking por Popularidad de Páginas (*o algoritmo *PageRank)*: con este modelo, el motor tiene en cuenta la cantidad de links que hacen referencia a una página determinada para valorar o calificar a esa página, asumiendo que una página muy citada es "mejor" que otras menos referidas.

El modelo de *Ranqueo por Popularidad* fue diseñado y registrado por *Google Inc.* con el nombre de *PageRank (™)*. Sus creadores fueron los fundadores de Google: *Larry Page* (de cuyo apelllido el modelo toma su nombre) y *Sergei Brin*, quienes idearon el modelo mientras eran estudiantes de Ciencias de la Computación en la Stanford University. Fuente: http://es.wikipedia.org/wiki/PageRank.

La empresa *Google Inc*. ha hecho público el uso del algoritmo *PageRank* en su estructura general, pero algunos detalles se han mantenido en reserva: en general, el ranking o calificación de una página se calcula teniendo en cuenta la cantidad de enlaces que la apuntan, pero se tienen en cuenta otros factores y algunos de los ellos aún no han sido publicados.

Por ejemplo, se sabe que al calificar una página *p* se tiene en cuenta también el valor de las páginas desde las cuales se apunta a *p*, asumiendo que no es lo mismo ser referenciada desde una página importante (como la de una universidad) que desde una página o blog personal. También se sabe que los valores que pueden asignarse a una página están en un rango determinado (por ejemplo, entre 0 y 10) y que mientras más alto el valor, mejor es considerada la página, aunque no está claro el mecanismo por el cual se asigna un valor inicial alto a una página específica (muy pocas tienen valor 10).

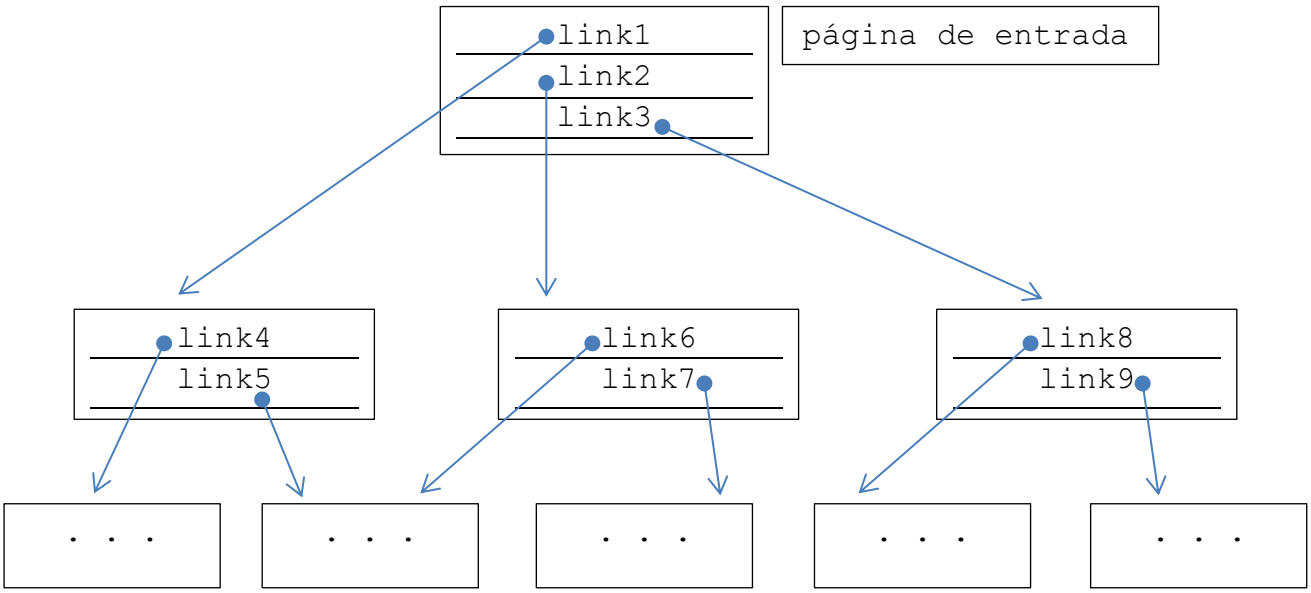
Otros factores que se sabe son usados para el cálculo del ranking son el tamaño de la página, el número de cambios que ha tenido, el tiempo transcurrido desde que la página fue actualizada, el texto contenido en los títulos de la página y el texto contenido en los anclajes de hipertexto.

[1] Toda la explicación que sigue sobre el modelo de *Ranqueo por Popularidad*, está inspirada en el material del curso "*Introduction to Computer Science*" – Organización Udacity (a cargo del profesor doctor *Dave Evans*, University of Virginia): https://www.udacity.com/course/cs101.

2.] El proceso de *Crawling*.

En esencia, el modelo de Google basado en sostener un ranking por popularidad de páginas comienza por parsear cada uno de los *N* documentos html disponibles, y tomar de ellos *cada uno de los links que posea*, lo cual puede hacerse con un proceso de extracción de strings en el documento: un link básicamente se identifica con el tag de apertura ** y por el tag de cierre **.

Sin embargo, como la página inicialmente analizada contiene un cierto número de links que apuntan a otras páginas, y estas nuevas páginas contendrán otros links, el proceso completo debe saltar a cada página detectada y recolectar de ellas cada uno de los links que posea, para armar un índice general de páginas visitadas. En la práctica, el módulo encargado de realizar esta tarea de rastreo de links se denomina *crawler*.

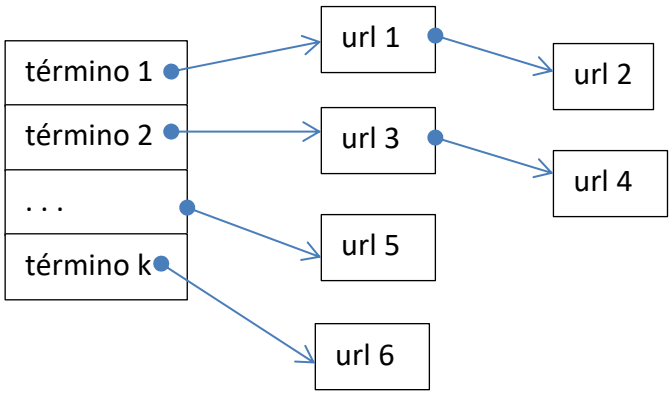


El proceso de crawling debe llevar una lista de las páginas detectadas que aún no han sido parseadas, y una lista de las páginas que ya analizó. Lo primero es importante ya que de otro modo podrían quedar páginas sin analizar, y lo segundo es relevante para evitar que una página ya analizada vuelva a parsearse.

Por otra parte, la gráfica anterior muestra un elemento que debe ser tenido en cuenta: si cada página analizada lleva a otras y estas a otras más, podría ocurrir que el proceso de crawling termine siendo demasiado profundo, innecesariamente. Para evitar esto, el crawler normalmente toma un parámetro que le indica el nivel o profundidad al que debe llegar: en la gráfica anterior, el esquema de páginas mostrado tiene tres niveles de links, por lo cual si el crawler está ajustado para analizar sólo dos niveles, entonces las páginas del tercer nivel serían ignoradas.

3.] El proceso de *Indexado de Términos*.

Una vez que se tiene explorado el esquema de encadenamientos por links de la páginas disponibles, se lleva a cabo un proceso de indexación de términos: se analiza cada una de las páginas detectadas en el crawling, se extraen las palabras o términos existentes en ellas, y se conforma una estructura de búsqueda rápida que contenga a cada término más una lista con las direcciones o urls de las páginas que contienen a esos términos (esto es similar al vocabulario y las listas de posteo del modelo vectorial). De hecho, la estructura de este índice puede montarse sobre una tabla hash, en la que cada entrada corresponda a un par *<término, lista de urls>*:



Con este índice, es simple comenzar a armar la respuesta a una consulta *q*: cada término de *q* se busca en el índice, y se recupera la lista completa de urls de las páginas que contienen a ese término. Sin embargo, sólo con esto no lograremos una respuesta óptima: el objetivo de un motor de búsqueda es retornar *la mejor página* para la consulta dada, o *una lista de las mejores páginas ordenadas por relevancia*, y no retornar simplemente una enorme lista de urls que contengan a cada posible término.

4.] El proceso de *Ranking* o *Calificación de Páginas*.

El proceso de crawling identifica la estructura de links de las páginas analizadas, y el proceso de indexado construye un índice general (que puede basarse en diversas estructuras de datos de búsqueda rápida) en el cual cada posible término se asocia a una lista de urls que lo contienen.

Nos ocupa ahora el proceso de ordenamiento de los resultados que mejor encajen con una consulta, llamado en general *proceso de ranking*. La idea es usar el concepto de *popularidad*: una página será más popular mientras más links la tengan referenciada. La medida de la popularidad de una página estará en relación directa con la cantidad de links que la referencien. Sin embargo, esto tiene un problema: no todos los links tienen el mismo valor. No es lo mismo que una página sea referida desde otra página importante (un diario, un portal científico, etc.), a que lo sea desde un blog personal. En principio, la medida o ranking de una página cuyo *url* viene como parámetro será entonces algo como:

```
ranking(0, url) = 1
ranking(t, url) = ∑ ranking(t-1, p)
```

Donde:

- **p** es un link que apunta al url dado.
- **t** es el paso o iteración actual.

En la expresión anterior, el primer parámetro representa el paso o número de visita a la página, suponiendo que un usuario navegue en forma aleatoria de una página a otra. Cada salto es una visita, y por ahora se supone que la primera entrada (la número 0) se valúa como 1.

Para evitar que una página se vea beneficiada por links que salen de páginas muy cargadas de enlaces de salida (como podría ser una página índice hacia diversos servicios), en la fórmula anterior se ajusta dividiendo por la cantidad total de links que salen de las páginas referidas que contienen a *p* (metafóricamente, si se tienen "muchos amigos", esas amistades pueden ser valoradas en menos que si se tienen pocos):

```
ranking(0, url) = 1
ranking(t, url) = ∑[(ranking(t-1, p))/salidas(p)]
```

Donde:

- **p** es un link que apunta al url dado.
- **salidas(p)** es la cantidad total de links de salida de la página que contiene a p.
- **t** es el paso o iteración actual.

Un problema para considerar es que una página que no tenga ningún link de entrada tendría un valor de ranking 0, y eso sería claramente perjudicial para páginas nuevas que recién ingresan a la web. Se puede hacer que cada página tenga un valor de popularidad inicial con base probabilística, para que ninguna página tenga un valor 0 inicial. Esencialmente, se cambiará la forma de la función que calcula el ranking, para sumar algún factor que permita obtener un valor diferente de cero incluso si la página no tiene ningún link de entrada.

Para ello, se aplicará en la fórmula un *factor de amortiguación d* cuyo valor inicial suele fijarse en *0.8* o en *0.85* y representa la probabilidad de que un usuario que navega en la web continúe pulsando links en vez de escribir una url directamente en la barra de direcciones del navegador. El valor del factor *d* es establecido en forma práctica por *Google*. Por lo tanto, la probabilidad de que el usuario *deje de pulsar links* y vaya directamente a una página aleatoria escribiendo su url es *1 - d.* Un valor *d = 0.8* implica entonces que se espera que en 8 de cada 10 veces el usuario preferirá seguir un link que escribir una dirección, y esa medida se considera un valor que describe aceptablemente la realidad.

Lo anterior dice que en general, un usuario comienza su navegación en alguna página seleccionada específicamente, luego navega siguiendo los links disponibles en ella, y sólo si llega a una página sin enlaces de salida decidirá saltar a cualquier otra página aleatoriamente, *aún si la misma no tiene links que la apunten*, escribiendo su dirección o tomándola de una lista provista por su motor de búsqueda o seleccionándola de entre sus marcadores de favoritos. La probabilidad de que el usuario escriba directamente el url de una página es entonces más baja (*1 – d = 1 – 0.8 = 0.2)* y describe también la probabilidad de entrar a una página que no tiene links de entrada.

Para mantener todo dentro un rango de valores razonables, si el número total de páginas es *N* se puede hacer un ajuste inicial del valor de cada página a *1/N* en lugar de *1* (en otras palabras, la suma inicial de las calificaciones de las *N* páginas será igual a *1*). La fórmula ya amortiguada, se ve como sigue:

```
ranking(0, url) = 1 / N
ranking(t, url) = (1-d)/N  +  d * ∑[(ranking(t-1, p))/salidas(p)]
```

Donde:

- **p** es un link que apunta al url dado.
- **salidas(p)** es la cantidad total de links de salida de la página que contiene a p.
- **t** es el paso o iteración actual.
- **N** es la cantidad total de páginas.
- **d** es el factor de amortiguación.

Como vemos, a cada calificación se le añade un valor para ajustarse más a la idea de que en cada paso podríamos estar comenzando nuevamente desde una página aleatoria cualquiera, sumando a la fórmula el valor *(1- d) / N*. Esto nos da la noción de las veces en que decidimos *NO* seguir un link (probabilidad *1 – d*) de entre las *N* páginas posibles. Si d = 0.8 entonces *(1- d) / N* es un valor

muy pequeño, pero diferente de cero. La sumatoria principal se multiplica por *d*, para incluir la probabilidad de tomar una página específica como página inicial.

La fórmula final es recursiva: se comienza suponiendo un ranking inicial de *1/N* para cada página en el momento 0 inicial, y luego se ajusta ese valor tantos pasos *t* hacia adelante como se quiera, para mejorar los valores. Por cada link *p* que apunta a una página, se toma siempre su último valor de ranking calculado en la iteración anterior, dividiendo por la cantidad de links que salen de la página que contiene a *p*. La multiplicación por *d* nos da la probabilidad de que esa página haya sido elegida como página inicial por el navegador, y el factor *(1 – d) / N* suma la probabilidad de que el navegador recomience de cero eligiendo una página al azar.

Como dijimos, la importancia de una página se califica teniendo en cuenta la cantidad de links que la apuntan y las páginas mejor calificadas aparecerán en los primeros lugares de las listas de respuesta a una consulta en los motores de búsqueda. El hecho entonces es que lograr que una página aparezca en esos primeros lugares implica una ventaja estratégica si la página pertenece a una organización comercial o con intereses económicos y por ello se han usado procedimientos para cambiar y ajustar artificialmente el valor de ranking de una página (es decir, inducir al motor a sobrevaluar una página mediante un engaño).

Los trucos usados para distorsionar los resultados obtenidos por un motor de búsqueda suelen englobarse bajo la designación general de *spamdexing*, también citado como *search engine spam* (spam de motores de búsqueda) o *search engine poisoning* (envenenamiento de motores de búsqueda). Los métodos de engaño son muy numerosos, aunque una técnica muy usada (designada en forma amplia como *link spam*) consiste en usar blogs, libros de visita, foros y otros sitios de acceso público para añadirles enlaces hacia la página web cuyo ranking se quiere manipular[2].

Para intentar protegerse de este tipo de ataques, en el año 2005 Google propuso un nuevo atributo para enlaces de hipertexto: *rel="nofollow"*, con la intención que cuando se calcule el ranking de una página no se tengan en cuenta los links en ella que tengan este atributo. Aún así, este recurso no termina de evitar potenciales problemas y efectos secundarios[3].

[2] Para mayores datos referidos al *spamdexing*, se recomienda consultar la Wikipedia en el siguiente enlace (¡del cual aseguramos con total honestidad que no ha sido incluido sólo para aumentar el ranking de la Wiki!): http://en.wikipedia.org/wiki/Spamdexing .

[3] Más información sobre el algoritmo PageRank puede encontrase en la Wikipedia (versión en inglés): http://en.wikipedia.org/wiki/PageRank.