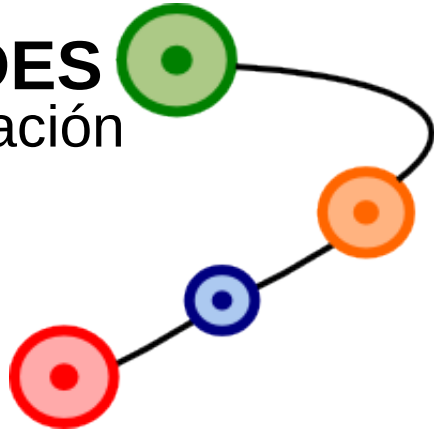
Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Recuperación de Información en la Web y Motores de Búsqueda

Dr.Gabriel H. Tolosa
tolosoft@unlu.edu.ar

Motores de búsqueda

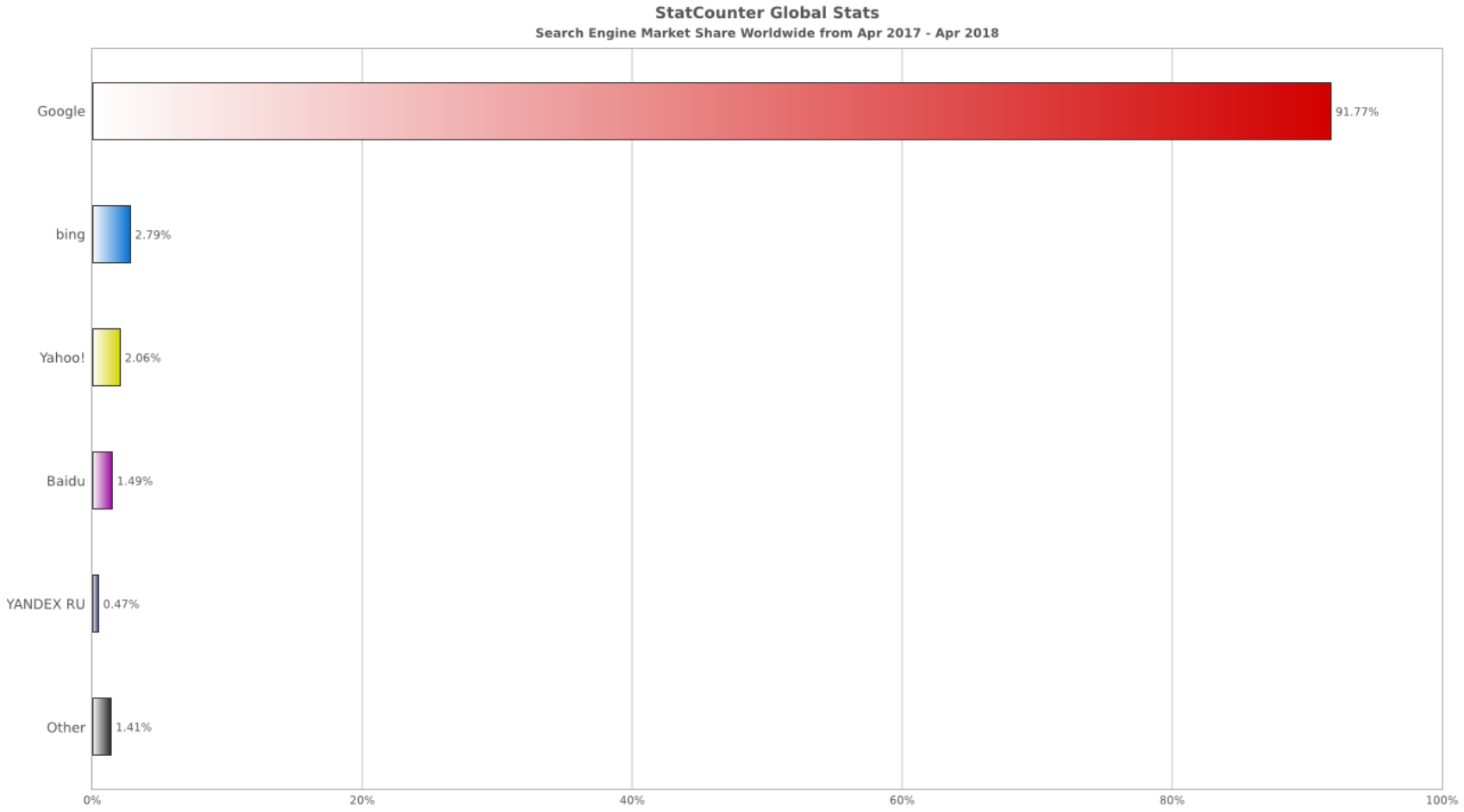
- Escenario/RI Web
- Arquitectura
- Recolección de páginas (Crawling)
- Ranking
- Queries y usuarios
- Escalabilidad (caching)

Motores de búsqueda

- **¿Son importantes?**
  - ~90% del tráfico a la mayoría de los sitios se encuentra mediante un motor de búsqueda
  - Son la primera interface entre los usuarios y la web
    - En el caso de sitios comerciales (productos) estar más allá de la posición 30 es ser “prácticamente” invisible.
  - Atraen la mayor diversidad de usuarios que cualquier sitio.
  - ~ 85% de las sesiones de usuario incluyen el uso de un MB
  - ~ 90% de los usuarios los usan para navegar la web

Motores de búsqueda

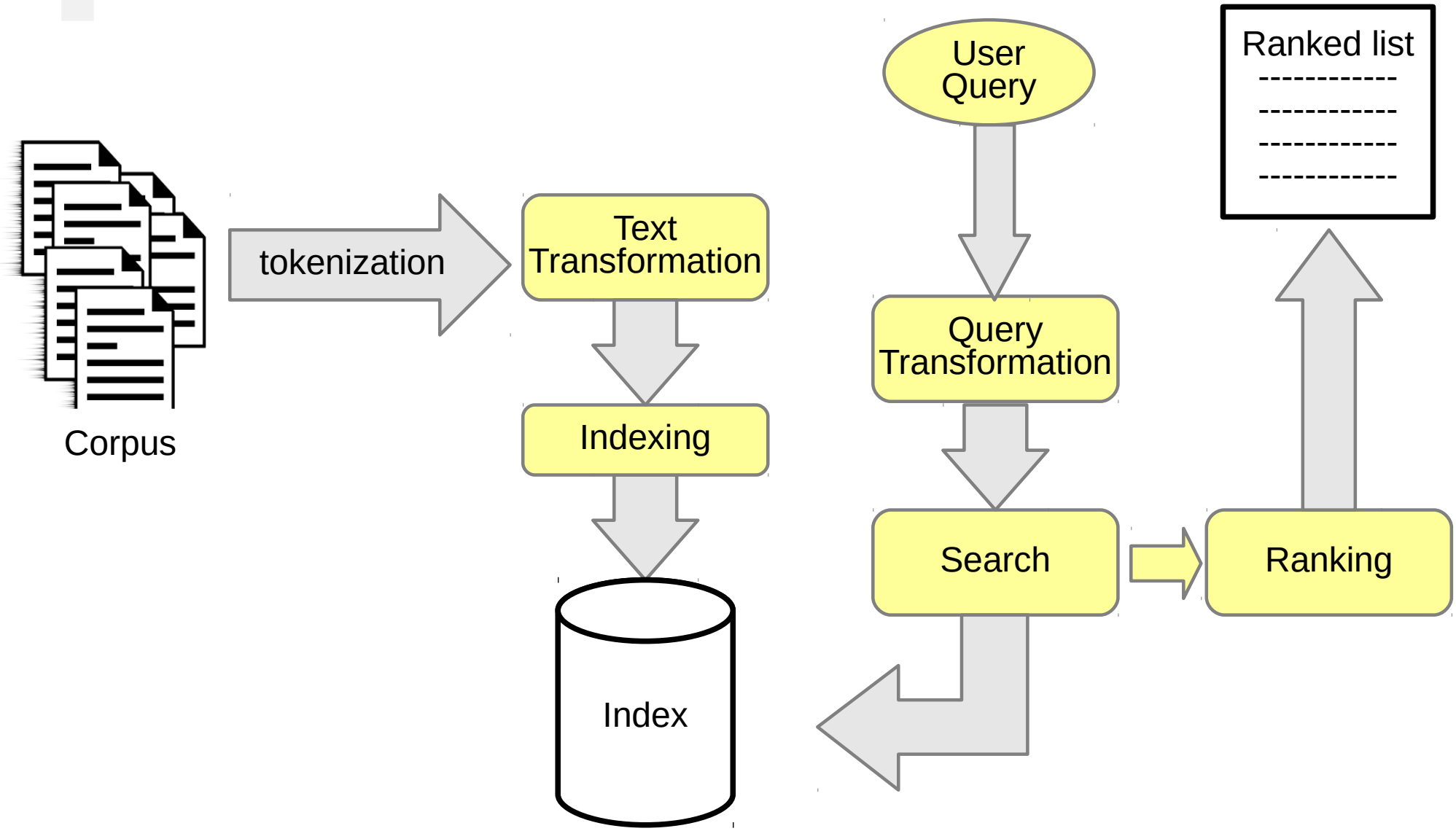
- **¿Cuáles se usan?**



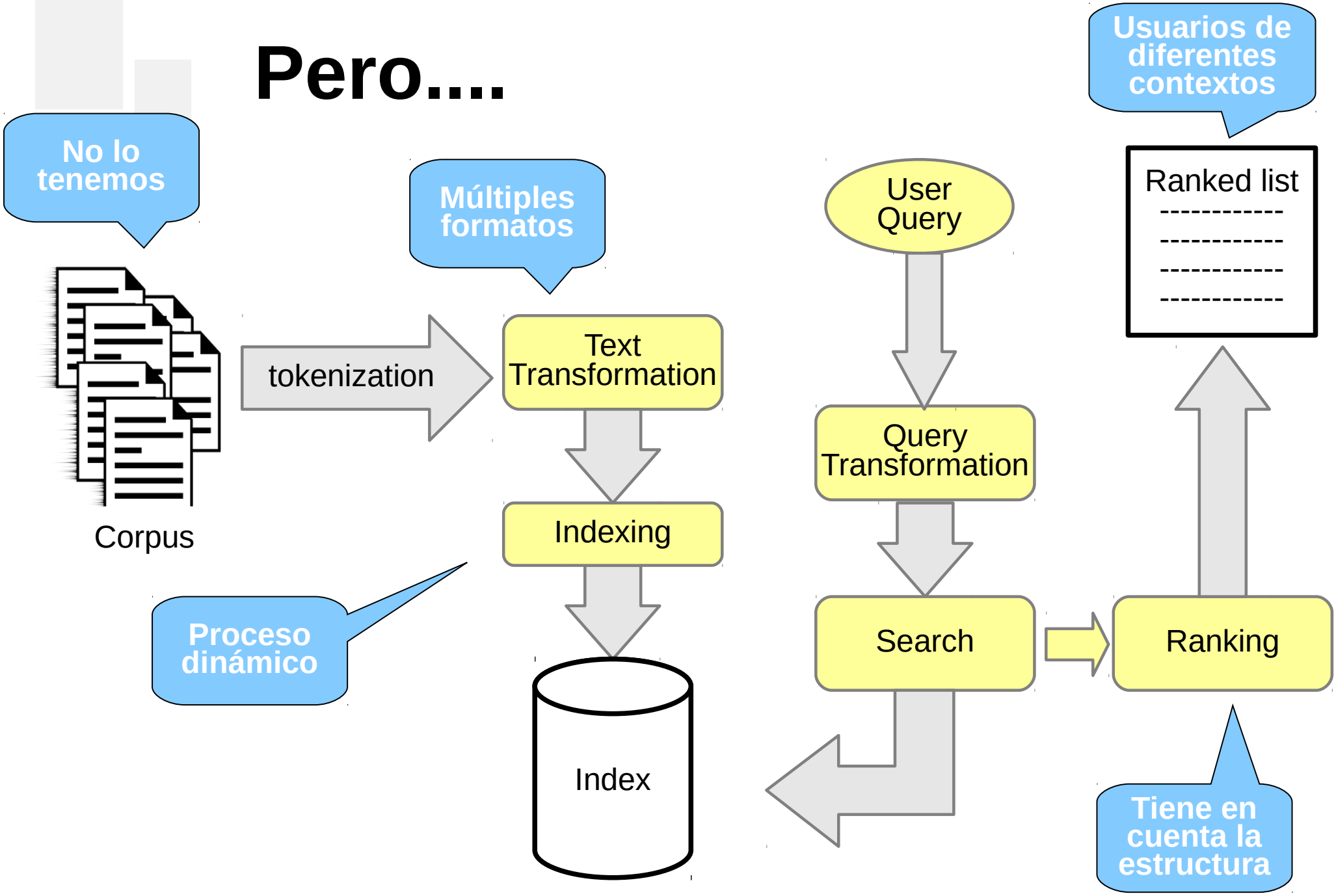
RI tradicional vs web

	RI Tradicional	RI en la Web
Objetivo	Recuperar documentos de texto con contenido relevente a la necesidad de información	Recuperar páginas web (y otros docs) de alta calidad relevantes a necesidad de información
Colección	Conjunto de documentos (generalmente homogénea)	La web pública (heterogénea)
usuarios	# proyectado intereses comunes	# impredecible intereses múltiples
Contenido	Relativamente pequeño y poco dinámico	Masivo y altamente dinámico
Consultas	Específicos	Cortos y poco descriptivos
Ranking	Según “grado” de relevancia	Relevancia + reputación + factores contextuales

SRI tradicional



Pero....


No lo tenemos
Múltiples formatos
Usuarios de diferentes contextos
Corpus
tokenization
Text Transformation
Indexing
Index
Proceso dinámico
User Query
Query Transformation
Search
Ranking
Ranked list
Tiene en cuenta la estructura

Arquitectura de un MB

Web

Crawling

Repository

Global analyzer

Indexing (+ Text Op.)

Metadata

Index

User Query

Query Transformation

Search

Ranking

Ranked list

Evolución de los MB

Primera generación

Solo utilizaban el texto en las páginas
Altavista, Exite, Lycos

Segunda generación

Analizan la estructura de enlaces de la web y los clicks
“Anchor text”. Google y PageRank

Tercera generación

Tratan de resolver “*la necesidad detrás de la consulta*”.
Ayudan al usuario: speell-checking, sugerencias, refinamiento
Integran múltiples fuentes (news, blogs, imágenes)
Análisis semántico básico. **Aún están evolucionando!**

Cuarta generación

Incrementar el uso de contexto y la actividad del usuario! (“*Information supply*”)

Evolución de los MB

Còmo determinar:
¨**la necesidad detràs de la consulta**¨

Determinación del contexto

- Espacial (ubicación del usuario o del objetivo)
- Stream del query (respecto de los anteriores)
- Información personal (perfil)
- Explícito (elige el usuario, por ej. un MB vertical)
- Implícito (uso de Google Argentina, google.com.ar)

Uso del contexto

- Restricción de resultados (eliminar inapropiados)
- Modulación del ranking (genérico, personalizado)

Evolución de los MB

¿Y los usuarios?

Las consultas:

- Las mayoría tienen de 1 a 3 términos (el 25% tiene 2)
- Términos imprecisos
- Uso subóptimo de la sintaxis (sólo ~10% con operadores)

Mucha variación en:

- Necesidades
- Expectativas
- Conocimiento
- Recursos (ancho de banda)

Comportamiento:

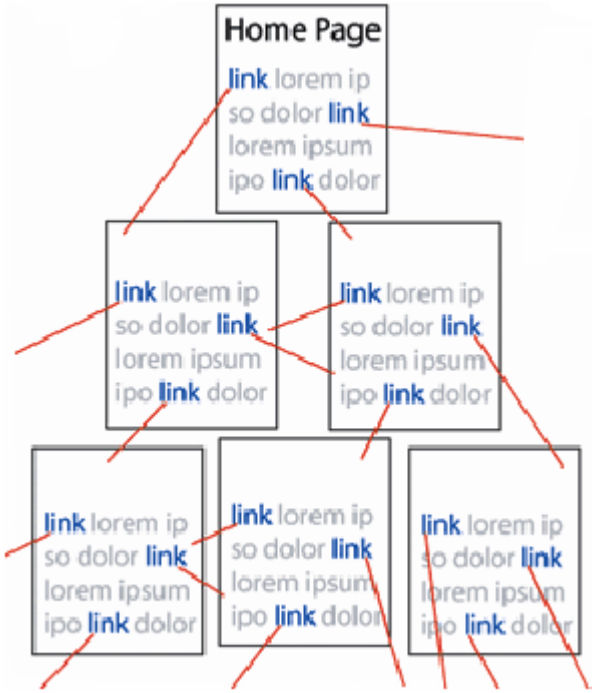
- Sólo examinan unos pocos resultados (2-3 páginas), ~85% sólo la primera
- Poco refinamiento (~80 no modifica la consulta original)
- La interface de búsqueda avanzada es poco utilizada

Crawling

El “corpus” web

- Creación no coordinada, distribuida (democrática)
  - Ni de contenido ni de enlaces
- Diversidad
  - No estructurado (txt, html)
  - Semi-estructurado (XML, objetos 'anotados')
  - Estructurado (BD), en menor medida.
- Tamaño: se duplica en pocos meses!
- Enlaces: 8/pág. en promedio
- Contenido dinámico
  - 'On the fly'
  - HTTP Get/Post

    http://www.google.com/search?hl=en&q=graph+structure+in+de+the+web+slides&btnG=Search
- SPAM



Crawling → Obtener la colección

- “Encontrar” y recuperar páginas automáticamente
- La web está constantemente cambiando
- Las páginas cambian
- La web no está bajo el control del propietario del motor de búsqueda
- Se basa solo en la URL:

`http://www.unlu.edu.ar/academia/unidades.html`

[proto] [hosts] [path] [objeto]

Crawling → Obtener la colección

Web crawling <=> atravezar un grafo

```
S := {páginas iniciales}

mientras no-vacía (S)
{
     tomar s desde S

     si s no fue recuperada antes:
          recuperar s

          parsear s

          para cada link l en s:
               agregar l a S
}
```

Crawling → Atravezar el grafo

Algorithm 6.1 Simple Web-Crawler to save link structure

```
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
```

Crawling → Cuestiones

- ¿Cómo hacer el crawling?
  - Calidad (las mejores páginas primero)
  - Eficiencia (evitar duplicados)
  - Cortesía (con los servidores)
- ¿Cuánto recolectar?
  - Cobertura
  - Cobertura relativa
- ¿Con qué frecuencia?
  - “Frescura”

Crawling → Cortesía

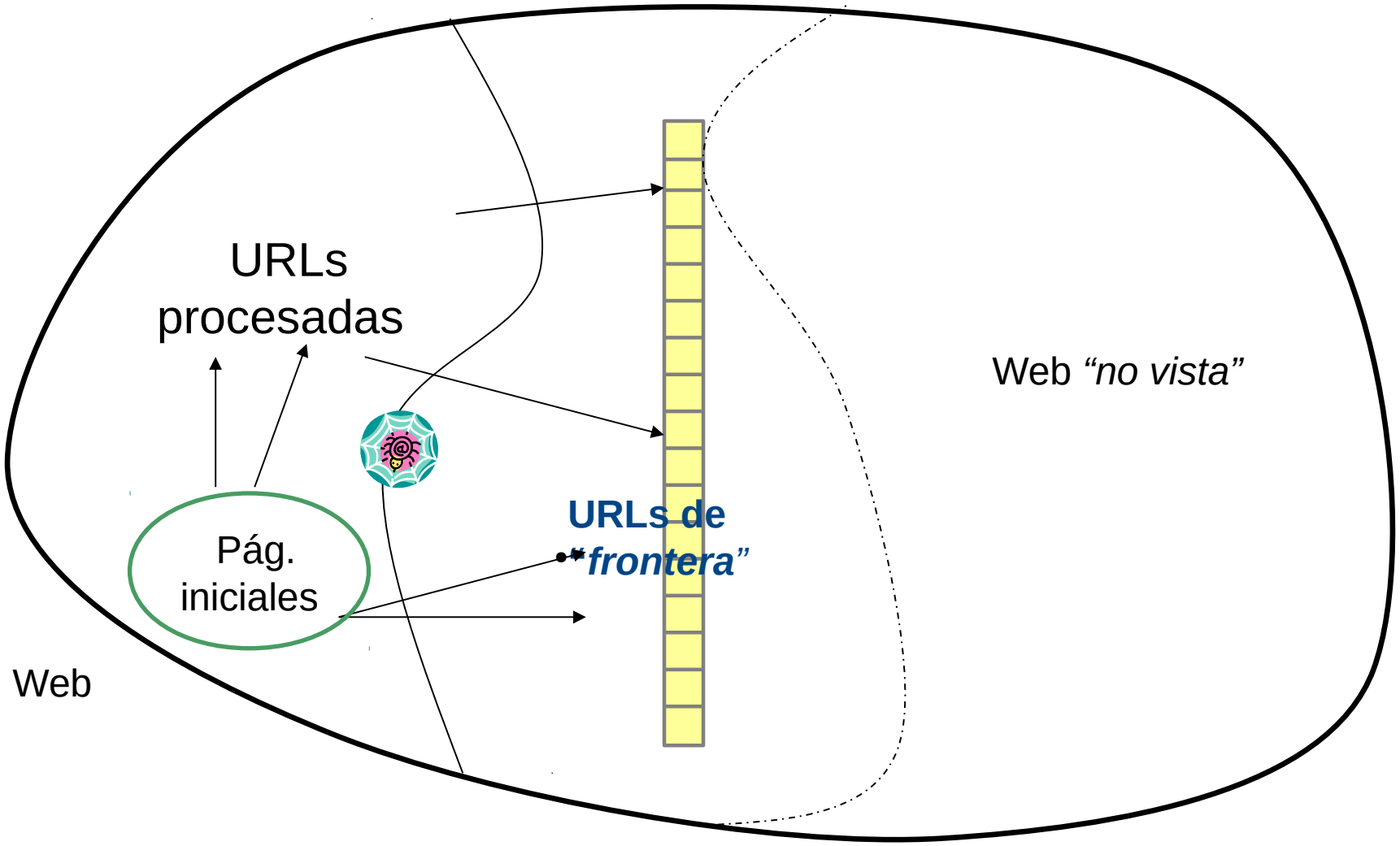
- Para evitar “acaparar” recursos de un servidor
  - Solicitar una página por vez
  - Incluir un retardo entre pedidos sucesivos al mismo server
    - Mantener una cola (para fetch) por servidor
  - Seguir los estándares de robots
  - Usar sitemap.xml

Crawling → Más específicamente

Para cada URL, el crawler:

- Solicita la resolución del nombre a un servidor DNS
- Abre una conexión con el servidor (IP) en un puerto (usualmente 80)
- Envía una solicitud HTTP, generalmente usando la primitiva GET
- Recupera el objeto y se parsea
- Finalmente, actualiza la lista de URLs (frontera)

Crawling → Frontera



Crawling → Control

- Existe un delay hasta recibir las respuestas
  - Eficiencia → múltiples conexiones (hilos). Cientos de páginas en paralelo
  - Robots.txt
    - `User-agent: *`
    - `Disallow: /privado/`
    - `Disalow: /usuarios/`
    - `Allow: /varios/publico/`
    - `Sitemap: http://www.misitio.com.ar/sitemap.xml.gz`

Sitemap ejemplo

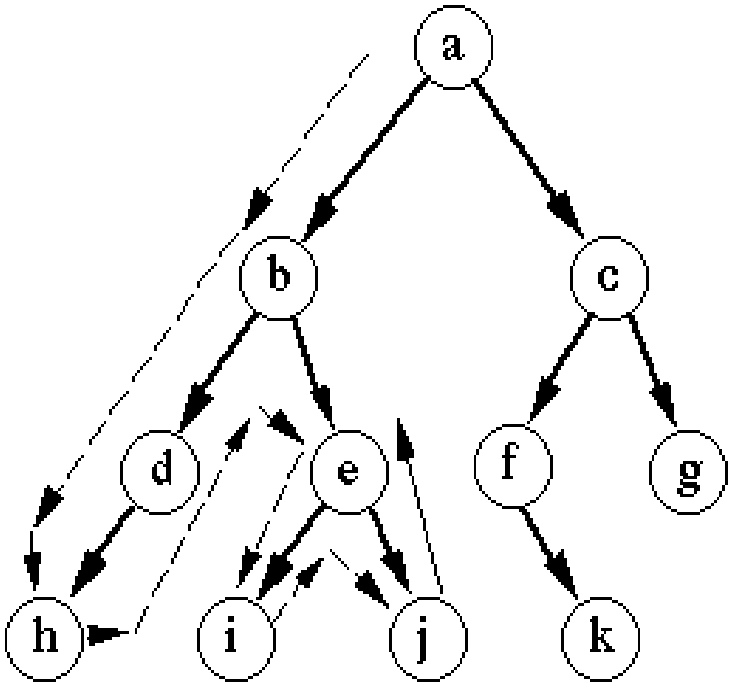
```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.google.com/schemas/sitemap/0.90">
  <url>
    <loc>http://www.sitemappro.com/</loc>
    <lastmod>2011-01-27T23:55:42+01:00</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://www.sitemappro.com/download.html</loc>
    <lastmod>2011-01-26T17:24:27+01:00</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://www.sitemappro.com/order.html</loc>
    <lastmod>2011-01-26T15:35:07+01:00</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://www.sitemappro.com/examples.html</loc>
    <lastmod>2011-01-27T19:43:46+01:00</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
  </url>
  ...
</urlset>
```

Ejemplo [Manning]

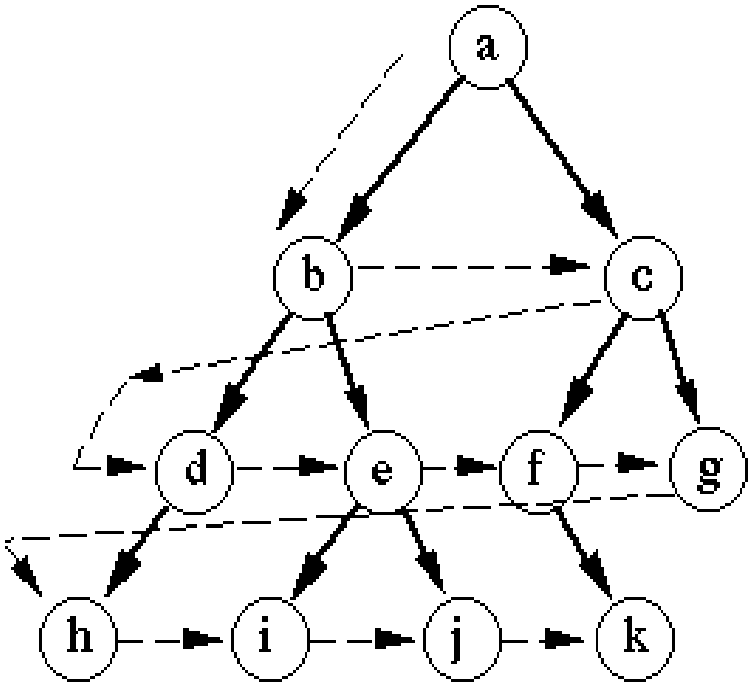
```
procedure CrawlerThread(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

Crawling → Estrategias

- Clásicas → Bread-First y Depth-First
- Otras → URL ordering

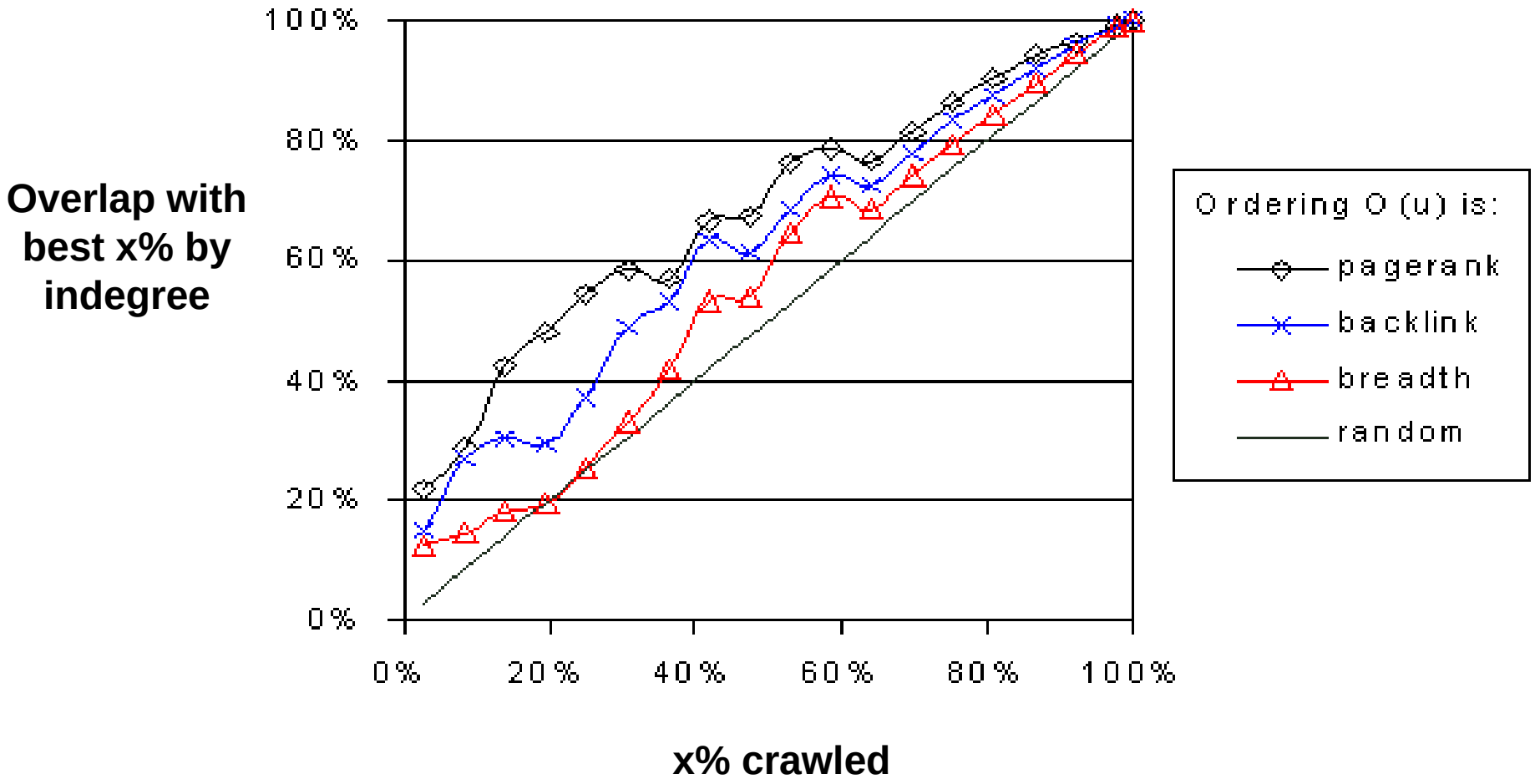


Depth-first search



Breadth-first search

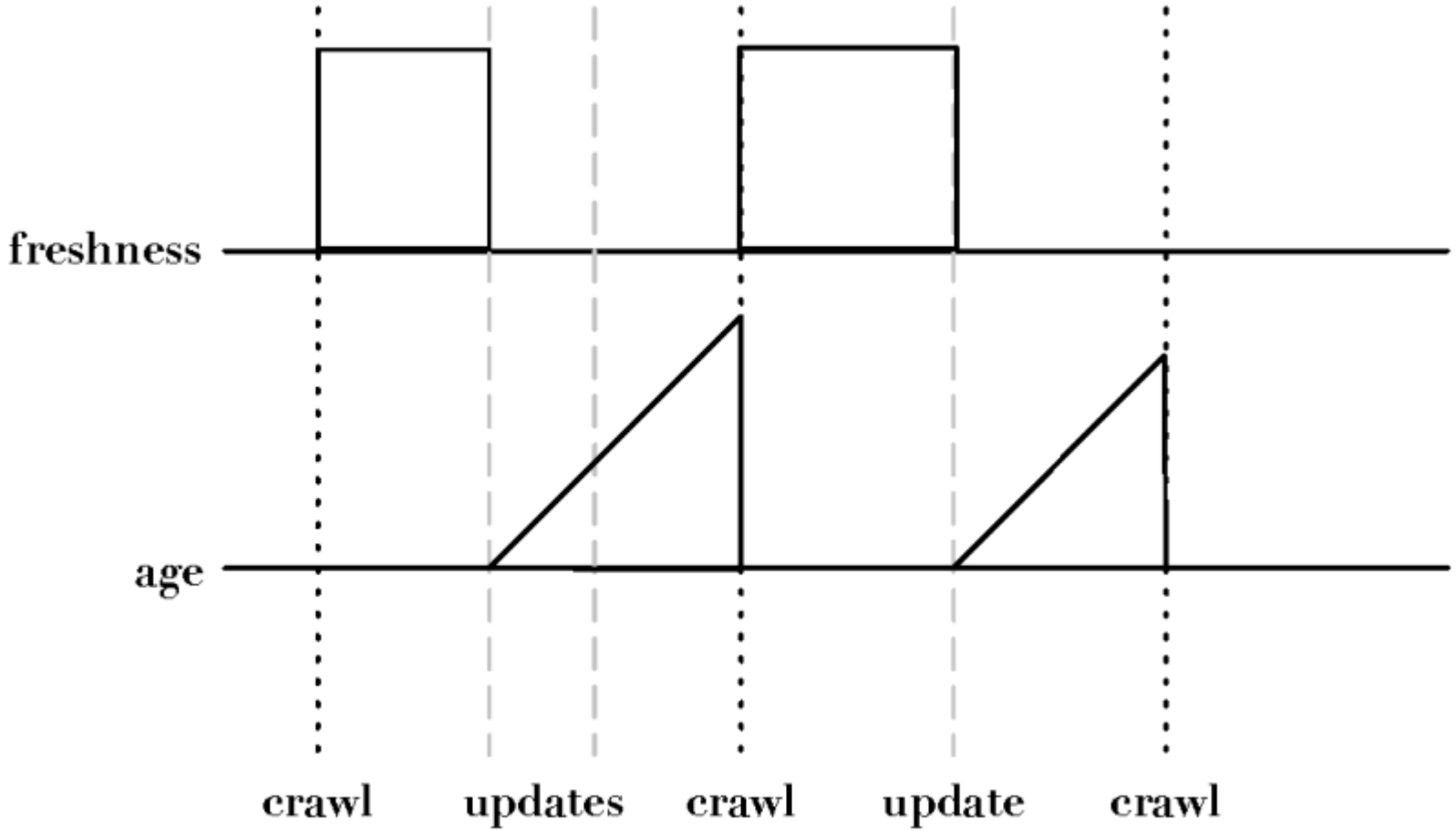
Crawling → Estrategias

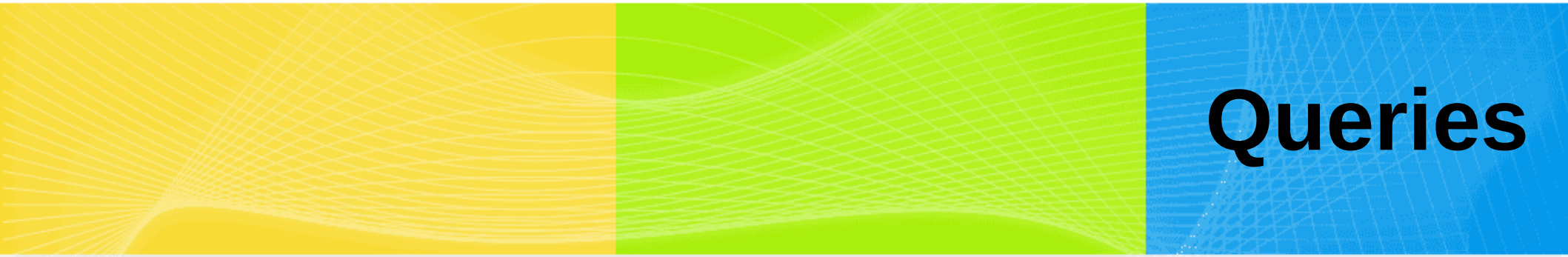


Crawling → Otras cuestiones

- Escalabilidad
- Crawling distribuido
- **Latencia/ancho de banda**
- **Profundidad**
- **Espejos/Duplicaciones**
- **Web SPAM → AIR**
- **DNS**
- **Robustez**
- **Cortesía/Estándares**
  - Explícita: robots.txt [www.robotstxt.org/wc/norobots.html]
  - Implícita: No sobrecargar un servidor

Dinámica → Freshness vs Age




Queries

Lenguajes de Queries

- No hay un lenguaje standard para queries web
  - No hay semántica explícita (c/ MB hace su interpretación)
  - Stemming, AND's... o no?
- Queries “Free-text” son el standard de facto
  - “Cualquier cosa” que el usuario escriba
  - No hay vocabulario controlado
  - Se aceptan errores de ortografía
  - Cuál es la diferencia con el lenguaje natural?
  - Cuál es la diferencia con una “pregunta”?

Operadores comunes en MB

Operator Syntax	Details	Google	Yahoo! Search	Bing	Ask
"." double quotes surrounding a string	Phrase search	yes	yes	yes	yes
+ preceded by a space, operates on the term/phrase that immediately follows	This operator ensures that the associated term is included "as is" in the results	yes	yes	yes	yes
– preceded by a space, operates on the term/phrase that immediately follows, Bing uses NOT as well	This operator ensures that the associated terms do not appear in any result	yes	yes	yes	yes
OR (as well as |) operates on preceding and succeeding terms or phrases	Equivalent to a Boolean OR	yes	yes	yes	yes
site: Followed by a site name	Returns results from the specific site only	yes	yes	yes	yes
hostname: Followed by a host name	Returns results from the specific host only	no	yes	no	yes
url: Followed by a URL	Checks that the following url exists in the engine index	no	yes	yes	no
inurl: Followed by a term	Returns results whose URL contains the specified term	no	yes	no	yes
intitle: Followed by a term	Returns results whose title contains the specific term	no	yes	yes	yes
inlink:/inanchor: Followed by a term	Returns results that contain the specific term in their link or anchor metadata	yes	no	yes	yes

[Baeza-Yates & Maarek] Modern Information Retrieval, 2nd edition, 2011

Consultas

- Existen diferentes motivaciones para usar un MB

- Caracterización [Broder et al.]

- **Informational** – ”saber” acerca de algo (~40% / 65%)
- **Navigational** – ”ir” a algún lugar (~25% / 15%)
- **Transactional** – ”hacer algo” (web-mediante) (~35% / 20%)
  - Access a service
  - Downloads
  - Shop
- **Areas “grises**

  - **Encontrar una buena páginas (HUB)**
  - **Explorar para “ver que hay allí”**
    -

Algoritmos evolutivos

Aerolineas Argentinas

Clima en Luján

Imagen Ubuntu 11.04

Canon S410

Alquiler auto Roma

Consultas

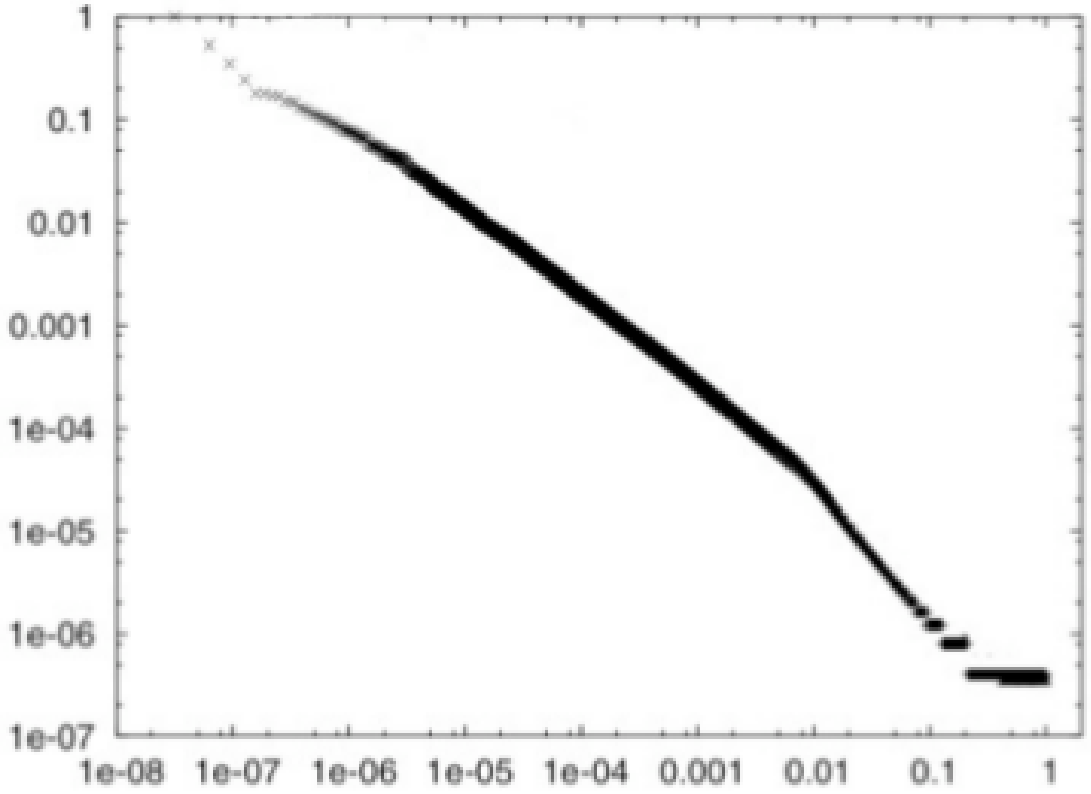
Ejemplo:

- Juan quiere comprar una impresora – **Transactional Query**
- Encuentra 3 posibles impresoras pero quiere más info acerca de éstas **– Infomational Query**
- Luego, se decide por una Lexmark y necesita la URL donde comprar (Lexmark, eBay, Mercadolibre, etc.) **– Navigational Query**
- Juan necesita hacer la compra en línea de la elegida **– Transactional Query**

Consultas

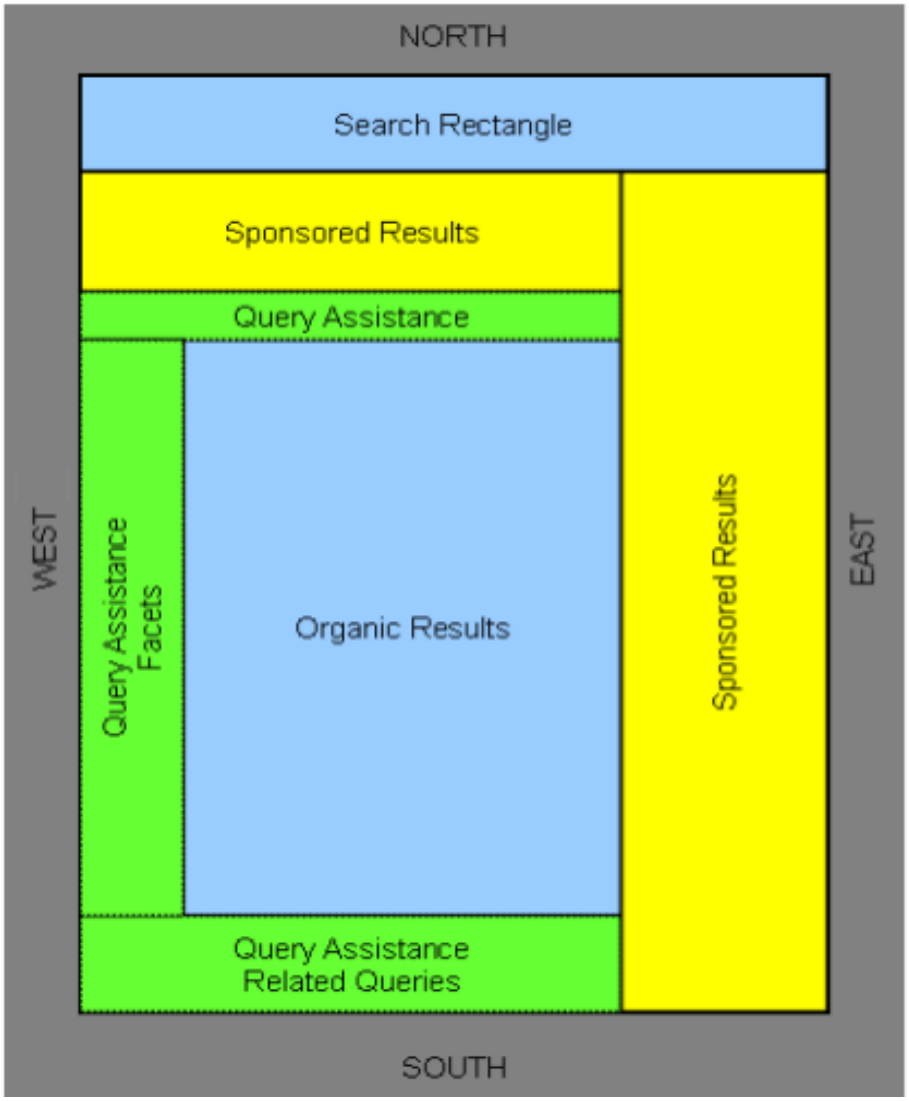
- La frecuencia de las consultas sigue una ley de Zipf con β = [0.6:1.4]

Ejemplo: Yahoo! R. Baeza-Yates, A. Gionis, F. P. Junqueira,V. Murdock,V. Plachouras, and F. Silvestri, “Design trade-offs for search engine caching,” ACM Trans. Web, 2008.



Ranking

Resultados: SERP Layout



Resultados

Google

volkswagen voyage

Búsqueda avanzada

Búsqueda

Aproximadamente 1.620.000 resultados (0,14 segundos)

Todo
Imágenes
Vídeos
Noticias
Más

Chivilcoy, Buenos Aires
Cambiar ubicación

La Web
Páginas en español
Páginas de Argentina
Páginas extranjeras traducidas

Todos los resultados
Sitios con imágenes
Más herramientas

Anuncios

Volkswagen Voyage 2011 - Estás Buscando Tu Nuevo 0Km?
www.**volkswagen**.com.ar/**Voyage** +1
Asesorate Con Expertos Acá!
Asesoramiento Comercial - Atención al Cliente - Amarok - Gol Trend

Venta Autos **Volkswagen** | DeMotores.com.ar
www.demotores.com.ar/Concesionaria +1
¿Buscás Autos **Volkswagen**? Todos los modelos en HausWagen

Voyage > Modelos > **Volkswagen** Argentina
www.**volkswagen**.com.ar/ar/es/models/**voyage**0.html +1
Sorprende por fuera. Sorprende por dentro., **Voyage**, Descubra un auto con excelente diseño, espacio, confort y versatilidad. Un sedan 4 puertas ...

Imágenes de **volkswagen voyage** - Informar sobre las imágenes




Autos **Volkswagen Voyage** 0 km - DeMotores.com, compra y venta ...
autos.demotores.com.ar/vm-12-**volkswagen-voyage** +1
Venta e información de Autos **Volkswagen Voyage** 0 km. Fichas técnicas, fotos, videos, reviews y vistas 360 de **Volkswagen Voyage** . Compra y venta de Autos ...

Anuncios

Volkswagen Voyage 2011
www.espasavw.com.ar/**voyage** +1
Conseguilo al mejor precio.
También financiación. Contactanos!

Volkswagen Voyage 2011
www.concesionariasenred.com.ar +1
Compra tu 0km - Representantes ofic
Llámanos 011-4762-0144

Volkswagen en DeAutos
www.deautos.com/**Volkswagen** +1
Venta de **Volkswagen voyage** Nuevos y Usados. Contratá el seguro online!

Plan **Volkswagen** Retira Ya
www.modenamotorhaus.com.ar +1
$11000 de descuento y cuotas fijas.
LLama ya:(011) 4343-0321/4343-0291.

Mira tu anuncio aquí »

Resultados

Gooooooooooogle ▶

1 2 3 4 5 6 7 8 9 10 **Siguiente**

Ayuda de búsqueda Enviar comentarios Google.com in English

Página principal de Google Programas de publicidad Soluciones Empresariales Privacidad Todo acerca de Google

Ranking

- Recuperación de Información
  - Términos incluir/excluir
  - Matching parcial → scoring
- **En la Web**
  - Frecuencia/ubicación de las palabras en el doc.
  - Metadatos
  - Existencia en directorio (si hay)
  - Tamaño/Edad del documento
  - Dominio
  - Y **$$$?**

Variables

De acuerdo a Matt Cutts [Ing. De Google] existen más de 200 variables que se tienen en cuenta para el ranking

- **Domain**
  - Age of Domain
  - History of domain
  - KWs in domain name
  - Sub domain or root domain?
  - TLD of Domain
  - IP address of domain
  - Location of IP address / Server
- **Architecture**
  - HTML structure
  - Use of Headers tags
  - URL path
  - Use of external CSS / JS files
- - **Authority Link** (CNN, BBC, etc)

- **Content**
  - Keyword density of page
  - Keyword in Title Tag
  - Keyword in Meta Description
  - Keyword in KW in header tags (H1, etc.)
  - Keyword in body text
  - Freshness of Content
- **Per Inbound Link**
  - Quality of website linking in
  - Quality of web page linking in
  - Age of website
  - Age of web page
  - Relevancy of page's content
  - Location of link (footer, navig., body)
  - Anchor text if link
  - Title attribute of link
  - Alt tag of images linking
  - Country specific TLD domain
  - Authority TLD (.edu, .gov)
  - Location of server

Variables

- **Cluster of Links**
  - Uniqueness of Class C address.
- **Internal Cross Linking**
  - No of internal links to page
  - Location of link on page
  - Anchor text of FIRST text link (Bruce Clay's point at PubCon)
- **Miscellaneous**
  - JavaScript Links
  - No Follow Links
- **Pending**
  - Performance / Load of a website
  - Speed of JS

- **Misconceptions**
  - XML Sitemap (Aids the crawler but doesn't help rankings)
  - PageRank (General Indicator of page's performance)

- **Penalties**
  - Over Optimisation
  - Purchasing Links
  - Selling Links
  - Comment Spamming
  - Cloaking
  - Hidden Text
  - Duplicate Content
  - Keyword stuffing
  - Manual penalties

Variables

Los más importantes según Eric Smidt [CEO de Google]

- Uso de negrita alrededor del término
- Uso de “header-tags” alrededor del término
- Presencia del término en “Anchor-text” entrante
- Pagerank
- Pagerank / autoridad del sitio
- Velocidad del sitio
- Presencia del término en el título HTML (Title-Tag)

Métricas complementarias

Discounted Cumulative Gain

"Mientras mas abajo se encuentre rankeado un documento relevante, menos últil es para el usuario"

DCG: *Es la ganancia acumulada en un ranking p*:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

Alternativa (usada por algunas empresas)

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1+i)}$$

DCG Ejemplo

Suponiendo 10 documentos rankeados en una escala 0-3

3, 2, 3, 0, 0, 1, 2, 2, 3, 0

Discounted gain:

3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

= 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0

DCG: 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Normalized DCG: *comparación con el ranking "perfecto"*

Ejemplo: Ranking perfecto: 3, 3, 3, 2, 2, 2, 1, 0, 0, 0

- **Valores ideales:** 3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10
- **NDCG (actual / ideal):** 1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88,0.88