Búsqueda en la Web

Crawling y PageRank

Particularidades de la búsqueda Web

- Retos: búsqueda en condiciones extremas
  - Escala, volatilidad, calidad muy variable, spam…
  - **Colección desconocida**

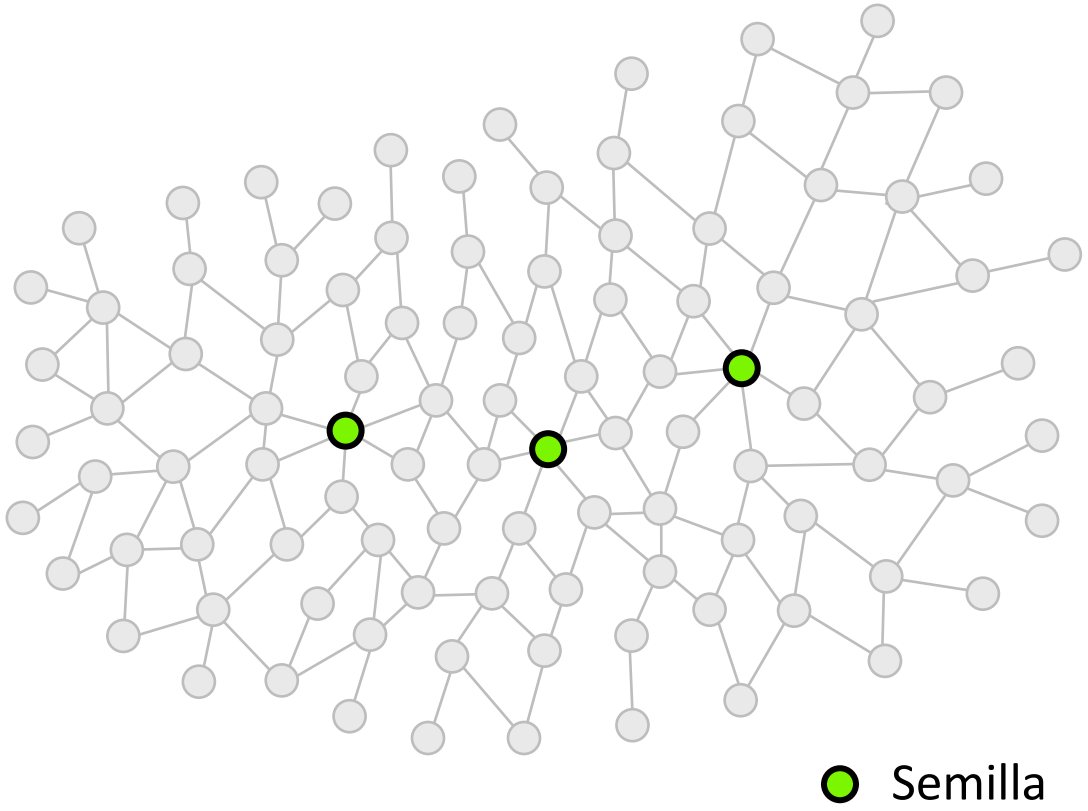
    → Necesidad de un módulo que la descubra: **crawler**
- Ventaja: estructura adicional de hiperenlaces
  - Algoritmos de ránking que explotan el **grafo Web**

    → **PageRank**

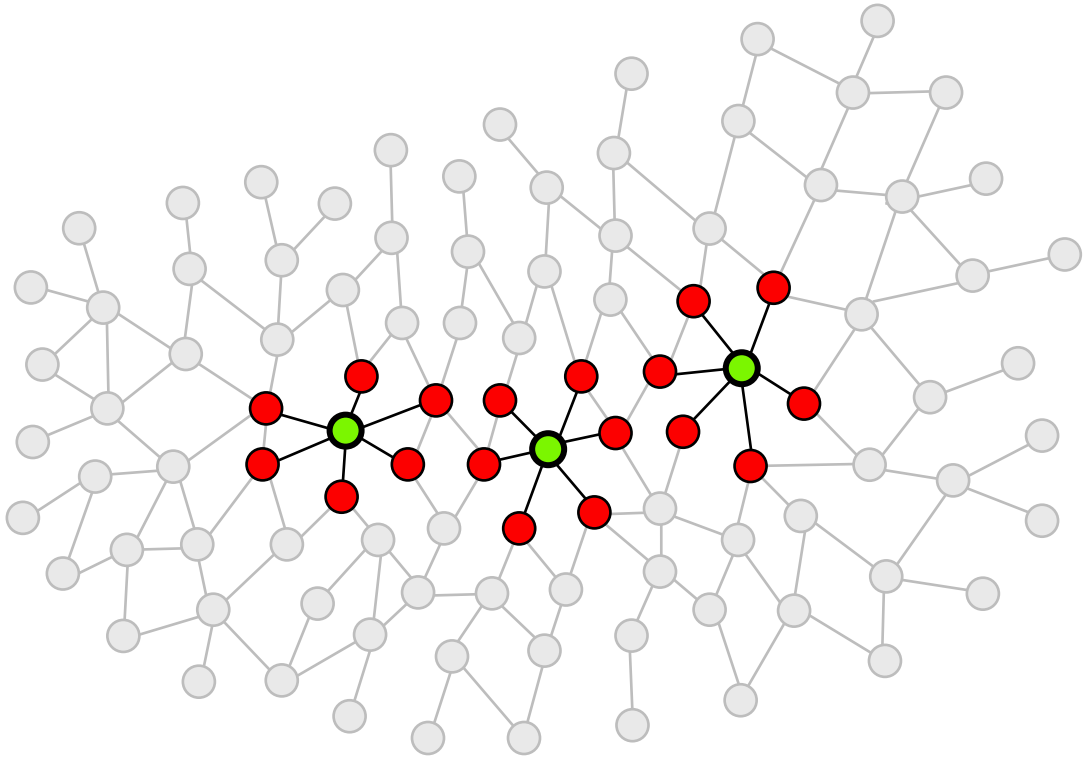
Crawling

- La Web se construye de forma libre y descentralizada
- Para indexar la colección, el buscador tiene que explorar y encontrarla
- La búsqueda de páginas se hace mediante un recorrido del grafo Web
- Lleva tiempo y necesita repetirse continuamente
- Lo importante es alcanzar las páginas que impactan en los resultados de búsqueda

Crawling: exploración de la Web



Crawling: exploración de la Web

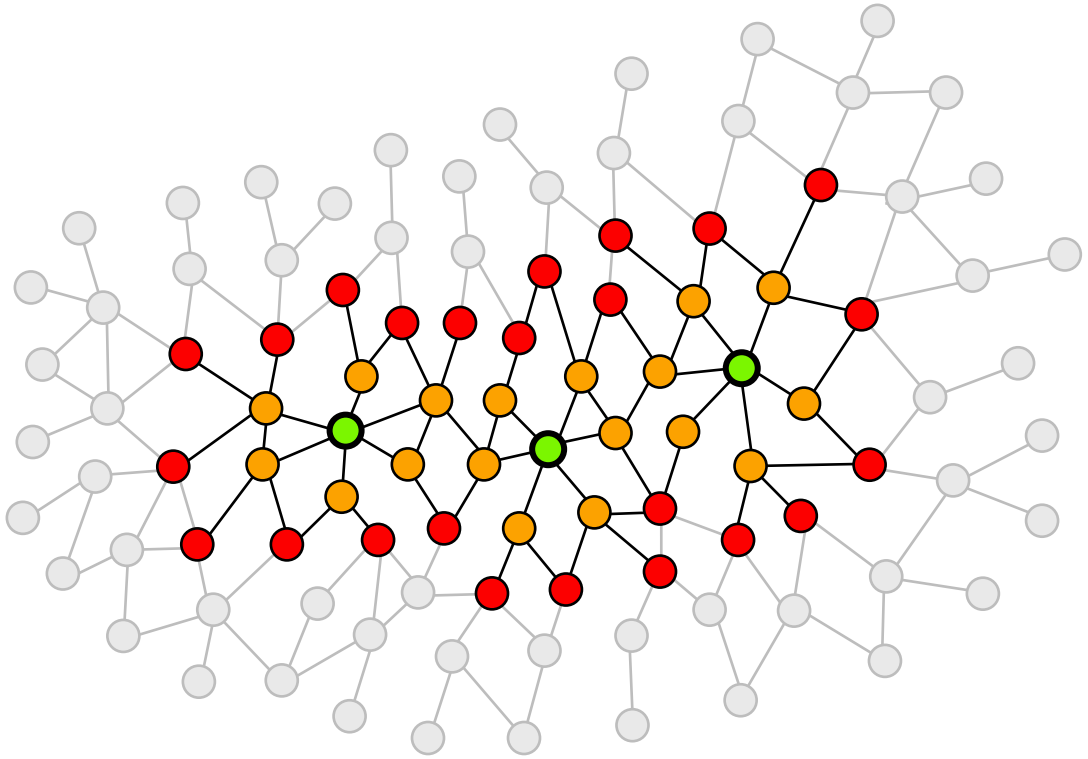


Semilla

Frontera de crawling

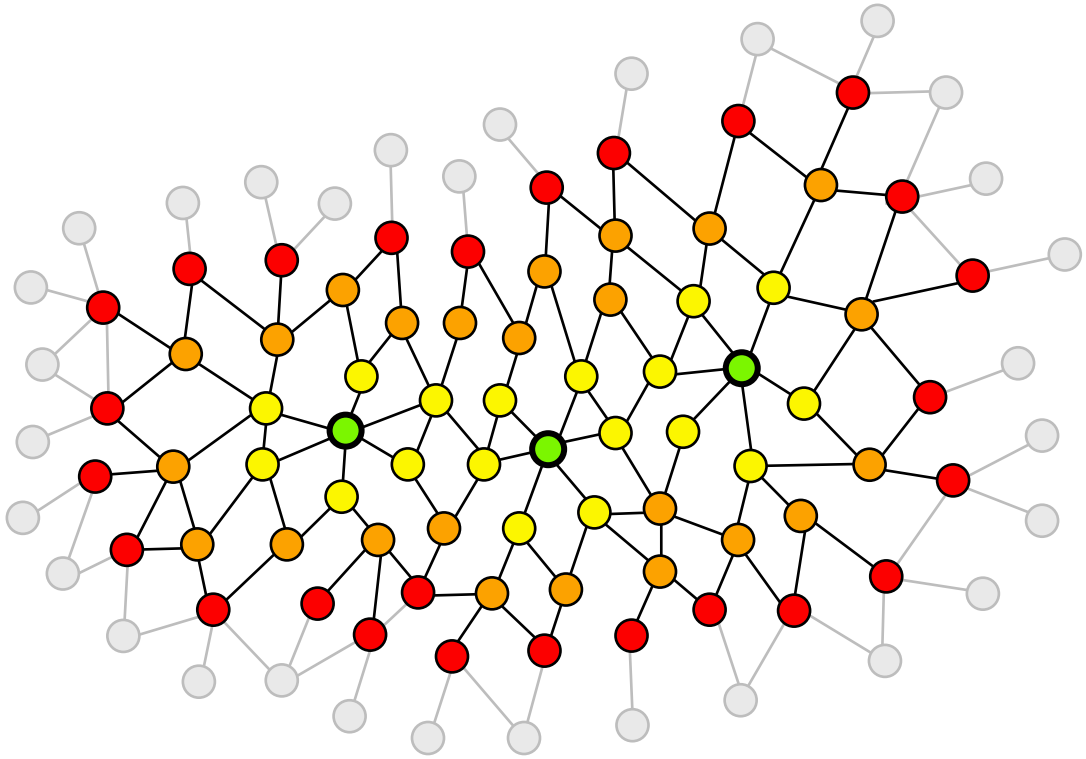
Web no descubierta aún

Crawling: exploración de la Web



- Semilla
- Web indexada
- Frontera de crawling
- Web no descubierta aún

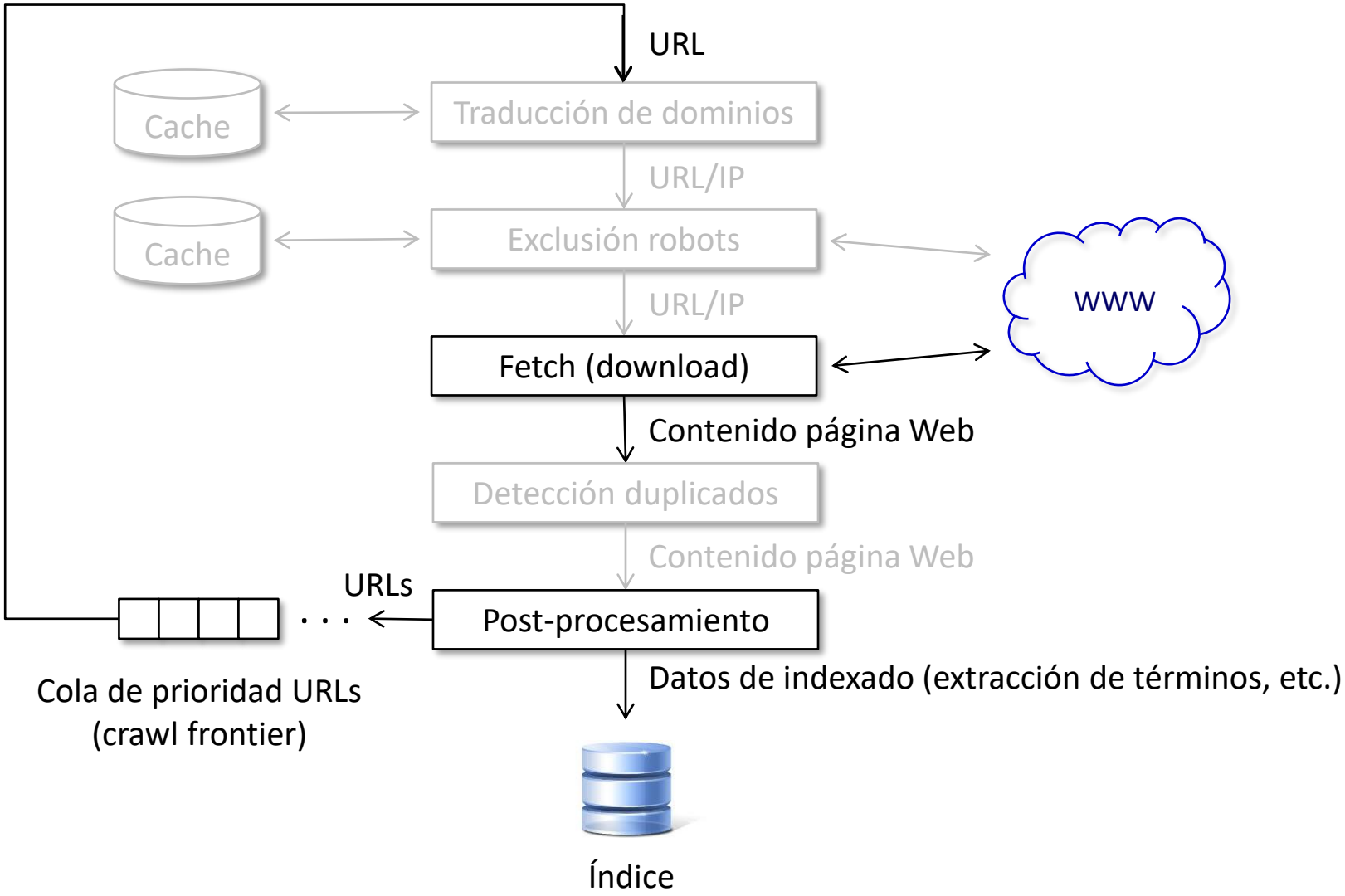
Crawling: cola de prioridad



Cola de prioridad

- Semilla
- Web indexada
- Frontera de crawling
- Web no descubierta aún

Crawling: pasos (cont)



Crawling – operaciones específicas

- Normalización de URLs
  - Mayúsculas en dominio y secuencias de escape, suprimir puerto por defecto, suprimir “.” y “..”, unificar “/” al final de la URL, etc.
- Aprovechar el texto de los enlaces (términos para el doc apuntado)
  - Con qué palabras describen autores externos el contenido de las páginas
  - Vulnerabilidad a ataques de link bombing
- Si una URL no responde repetidas veces, eliminarla
  - Del índice y de la cola de prioridad

Criterios de prioridad de crawling

- Semillas
  - Portales importantes, portales de noticias, ODP, etc.
- Criterios de prioridad en la cola de frontera de crawling
  - Tiempo de permanencia en la cola
  - Frecuencia y tipo de cambios de las páginas

    (p.e. periódico digital vs. portal de una facultad)
  - Impacto de los cambios en los ránkings de búsqueda
- Impacto en los ránkings
  - Nº de veces que la página aparece en resultados de búsqueda
  - PageRank, frecuencia de clicks de las URLs en un log

Crawling – cortesía con el servidor Web

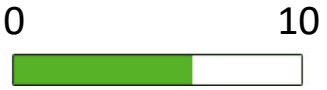
- Exclusión de páginas
  - Protocolo de exclusión robots.txt (también se suele guardar en cache): User-agent, Allow, Disallow
  - <meta name="robots" content="noindex,nofollow">
  - <a href="…" rel="nofollow"> (muy utilizado automáticamente en blogs y wikis)
- Moderar el nº de peticiones por minuto
  - P.e. una cada 1-60s, habitualmente > 20s en promedio
  - robots.txt → Crawl-delay
  - Aun así, los crawlers son los mayores consumidores de ancho de banda en Internet
- Autoidentificarse con el parámetro User-agent en la petición http
  - P.e. Googlebot, Bingbot, Yahoo! Slurp, etc.

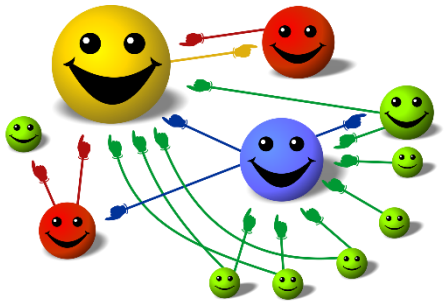
Crawling – páginas dinámicas

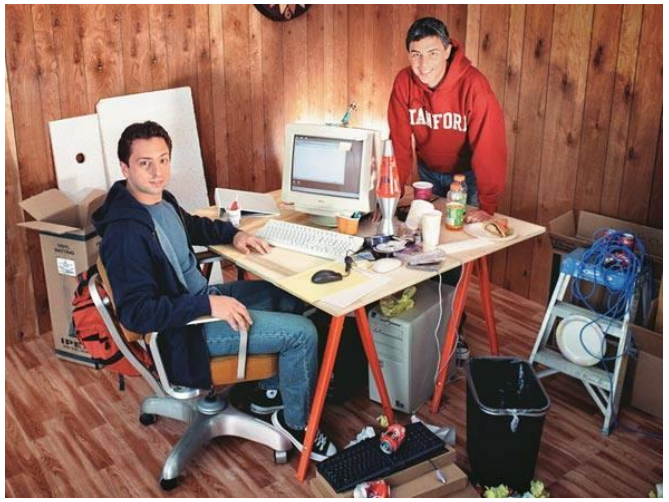
- Muchas no se pueden indexar
  - P.e. toman input del usuario (front-end de aplicaciones), acceso vía enlaces creados con JavaScript, redirects, protegidas por password, etc.
- Otras sí
  - Existe un camino de enlaces visible para el crawler: equivale a una página estática
  - No existe camino de enlaces: archivos sitemap (p.e. catálogos de tiendas online, etc.)
- Archivos sitemap (ver http://www.sitemaps.org)
  - Propuesto por Google en 2005, secundado poco después por Yahoo, MSN, Ask, IBM…
  - Contienen un listado de URLs a indexar, con detalles de prioridad (relativa), periodicidad de actualización, fecha de última modificación, etc.
  - La URL de ubicación del archivo sitemap se indica en robots.txt
  - Se suelen admitir máx 50.000 URLs / 10MB por sitemap, permitiendo un archivo índice de sitemaps con los mismos máximos (generalmente comprimido con gzip)
  - El servidor vuelca sus URLs dinámicas (p.e. generan consultas a una BD) a un listado sitemap en XML, el crawler las incluye en la cola de URLs
- También es posible enviar una URL manualmente para solicitar indexación

Ránking basado en enlaces: PageRank

- Aprovechar la estructura de links para extraer indicios de importancia
- Independiente de la consulta
  - Después se combinará con scores por consulta
- Efectividad de Google
- Múltiples otras aplicaciones posteriores







PageRank: principio general

- Los links entrantes son un indicio de importancia
- Tanto más si el link procede de una página importante
- Pero tenemos en cuenta también si la página de origen es muy “pródiga” con los links

S. Brin & L. Page. The anatomy of a large-scale hypertextual Web search engine. 7th Intl. Conf. on World Wide Web (WWW7), 1998

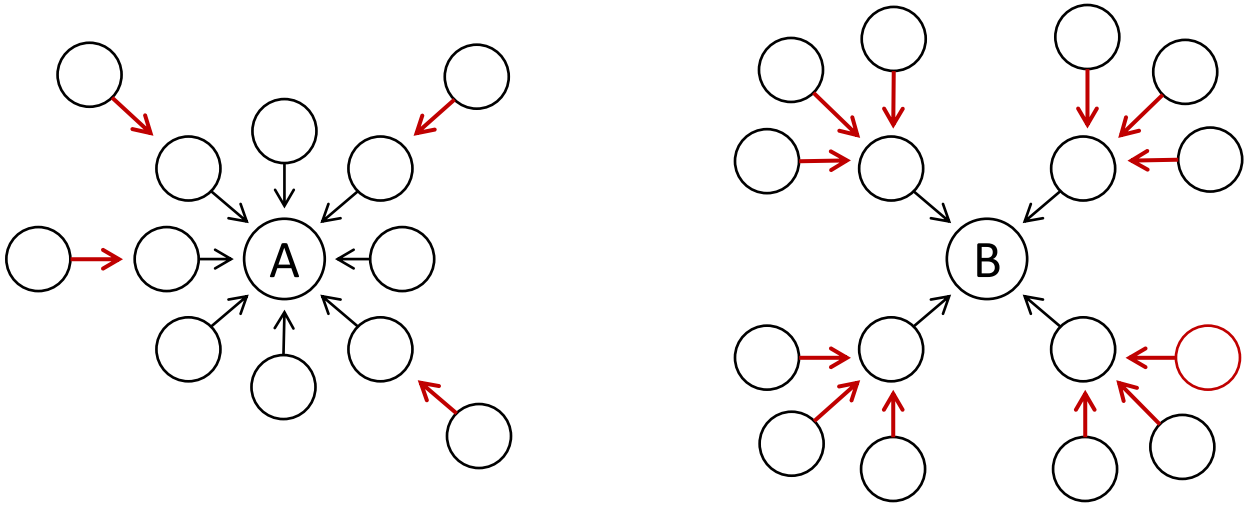
PageRank

$$P(d_j) = \sum_{d_i \to d_j} 1$$

A más importante que B

PageRank

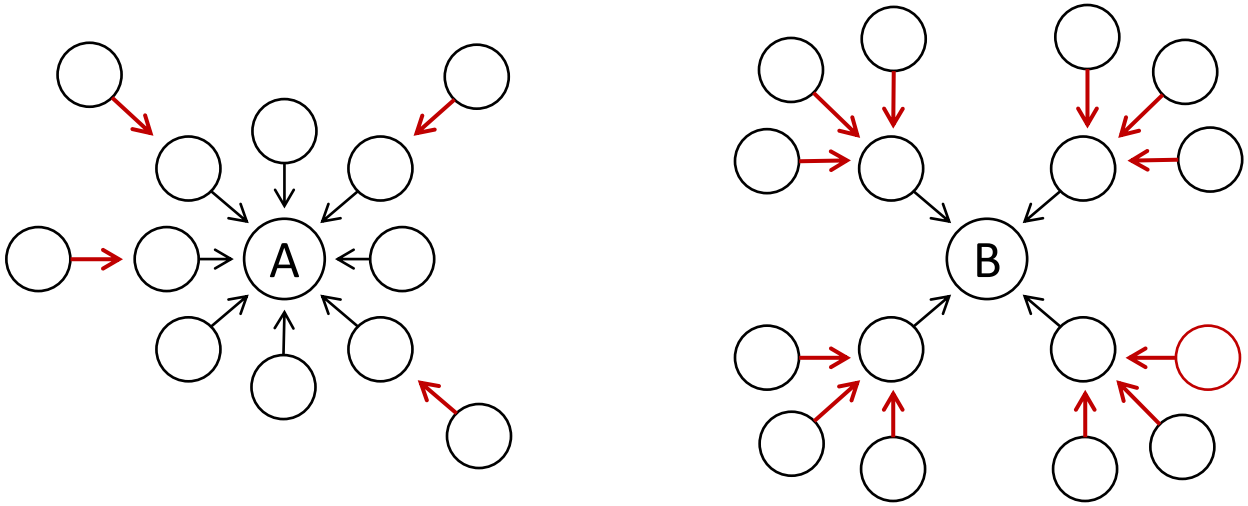
$$P(d_j) = \sum_{d_i \to d_j} 1$$



¿A más importante que B?

PageRank

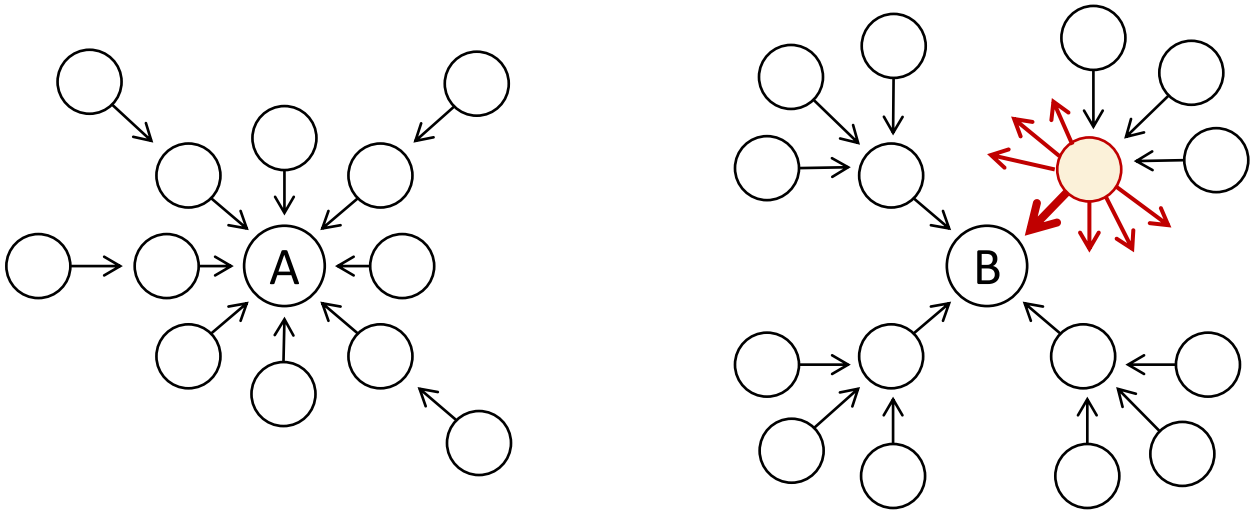
$$P(d_j) = \sum_{d_i \to d_j} P(d_i)$$



¿A más importante que B?

PageRank

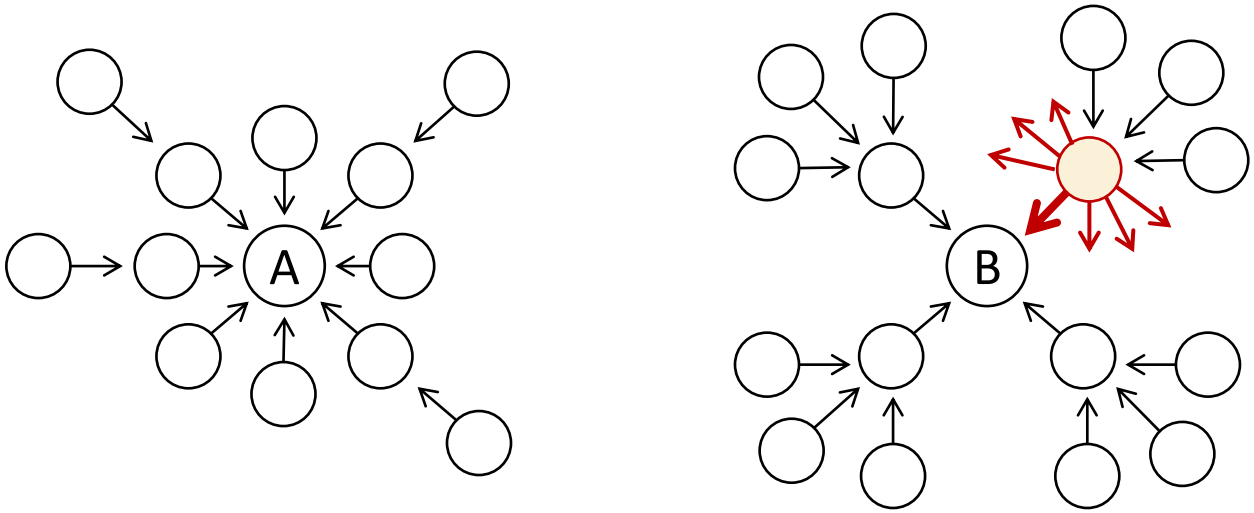
$$P(d_j) = \sum_{d_i \to d_j} P(d_i)$$



¿A más importante que B?

PageRank

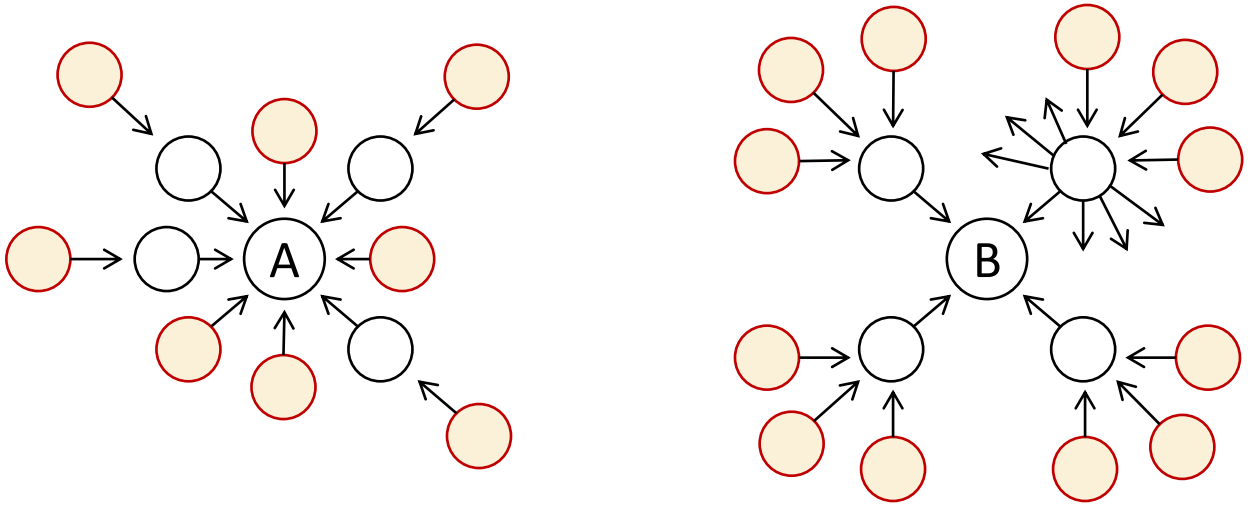
$$P(d_j) = \sum_{d_i \to d_j} \frac{P(d_i)}{\#out(d_i)}$$



¿A más importante que B?

PageRank

$$P(d_j) = \sum_{d_i \to d_j} \frac{P(d_i)}{\#out(d_i)}$$

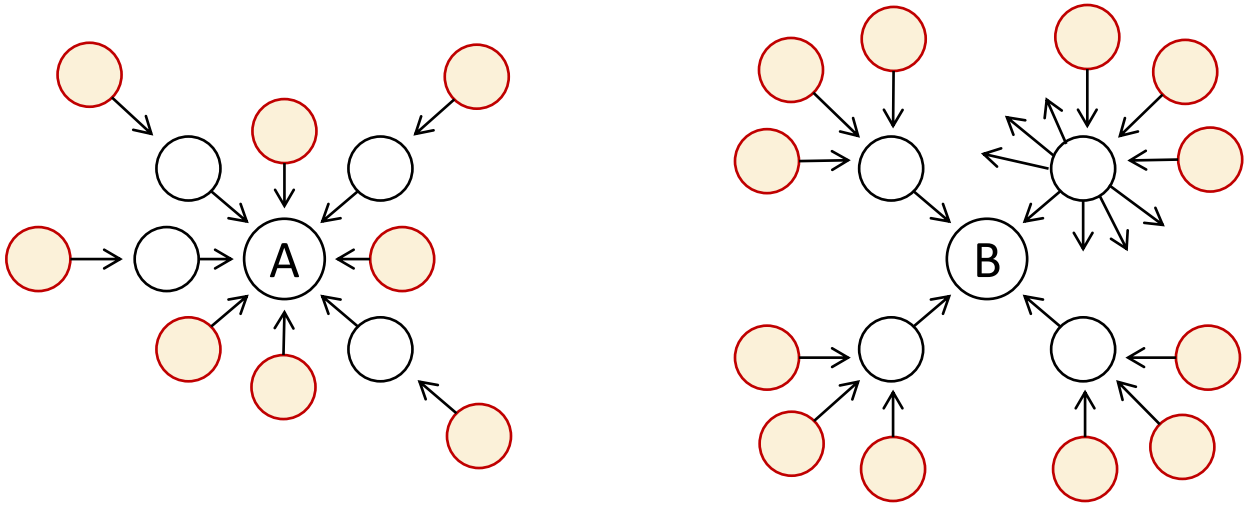


¿A más importante que B?

PageRank

$$P(d_j) = \frac{r}{N} + (1 - r) \sum_{d_i \to d_j} \frac{P(d_i)}{\#out(d_i)}$$

$$r \in (0,1)$$

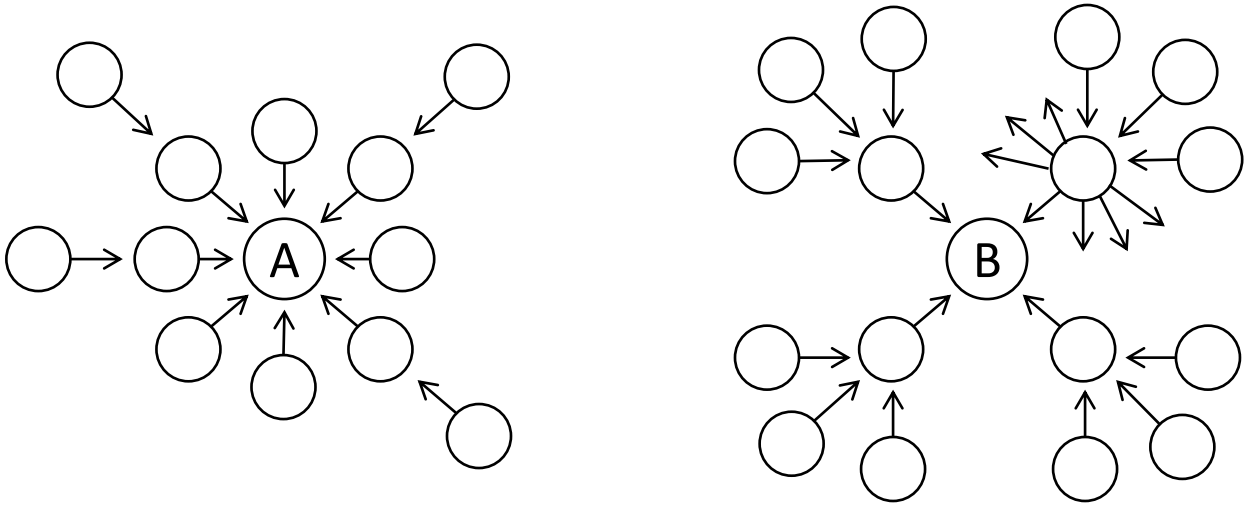


¿A más importante que B?

PageRank

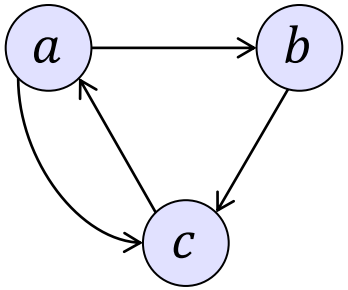
$$P(d_j) = \frac{r}{N} + (1 - r) \sum_{d_i \to d_j} \frac{P(d_i)}{\#out(d_i)}$$

$$r \in (0,1)$$



¿A más importante que B?

Ejemplo



$$P(a) = r/3 + (1-r)P(c)$$
$$P(b) = r/3 + (1-r)\,P(a)/2$$
$$P(c) = r/3 + (1-r)\big(P(a)/2 + P(b)\big)$$

P.e. con $r = 0.5$

- Resolviendo el sistema de ecuaciones $\rightarrow \begin{cases} P(a) = 14/39 \\ P(b) = 10/39 \\ P(c) = 15/39 \end{cases}$
- En general no es viable resolver simbólicamente un sistema de ecuaciones con millones de variables $\Rightarrow$ Cómputo iterativo (solución numérica)

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$\cdots$
$P(a)$	0.3333	0.3333	0.3750	0.3542	0.3594	0.3594	0.3587	0.3590	0.3590	$\cdots$
$P(b)$	0.3333	0.2500	0.2500	0.2604	0.2552	0.2565	0.2565	0.2563	0.2564	$\cdots$
$P(c)$	0.3333	0.4167	0.3750	0.3854	0.3854	0.3841	0.3848	0.3846	0.3846	$\cdots$

PageRank: algoritmo simple

PageRank ($links$)

for $k \leftarrow 1$ to $|links|$ // Compute #outlinks of all nodes

$out[links[k].from] + +$

for $i \leftarrow 1$ to N do // Initial values

$P[i] \leftarrow 1/N$ // (division by N can be omitted)

while *convergence condition* // Compute PageRank iteratively

for $i \leftarrow 1$ to N do

$P'[i] \leftarrow r/N$ // (division by N can be omitted)

for $k \leftarrow 1$ to $|links|$ *do*

$i \leftarrow links[k].from$

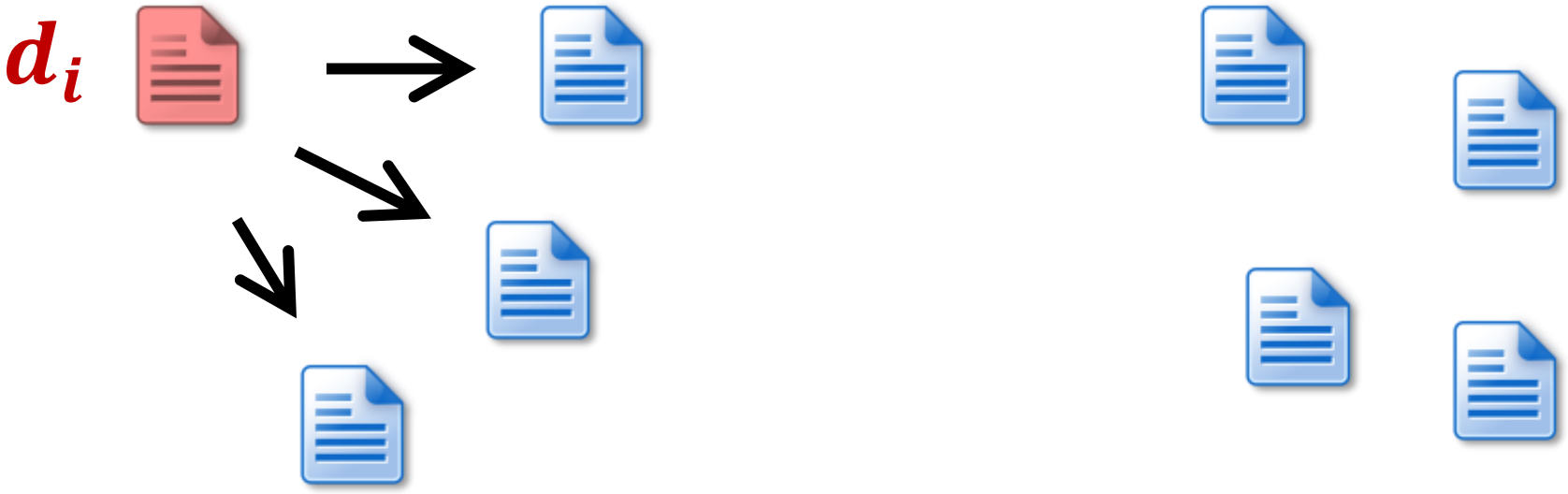
$j \leftarrow links[k].to$

$P'[j] \leftarrow P'[j] + (1 - r)\, P[i]/out[i]$

for $i \leftarrow 1$ to N do

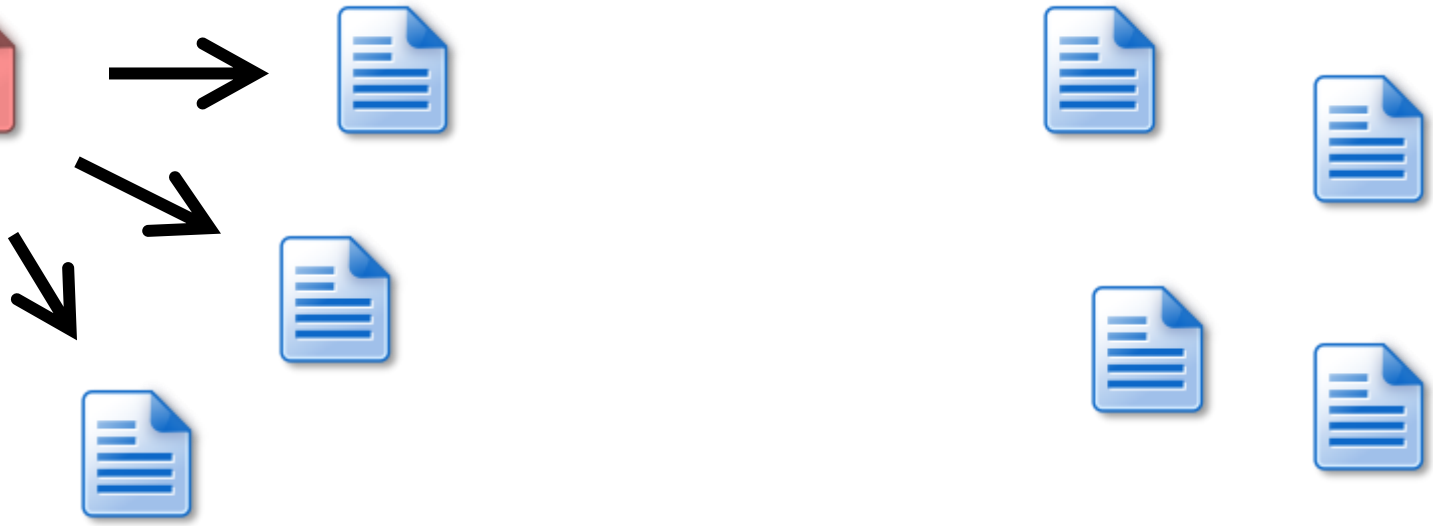
$P[i] \leftarrow P'[i]$ // To handle sinks, add $(1 - \sum_i P'[i])/N$

Random walk

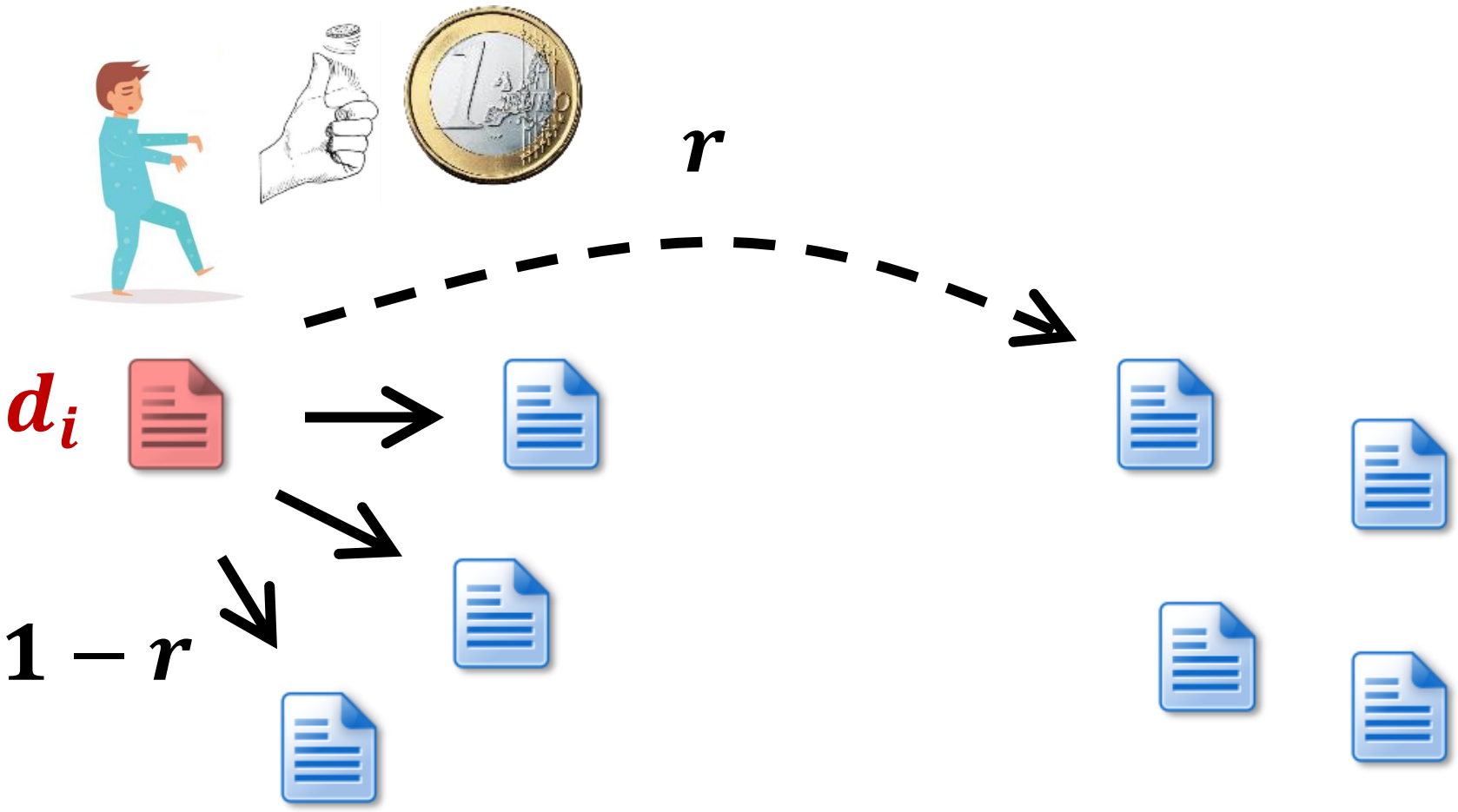


Random walk

d_i

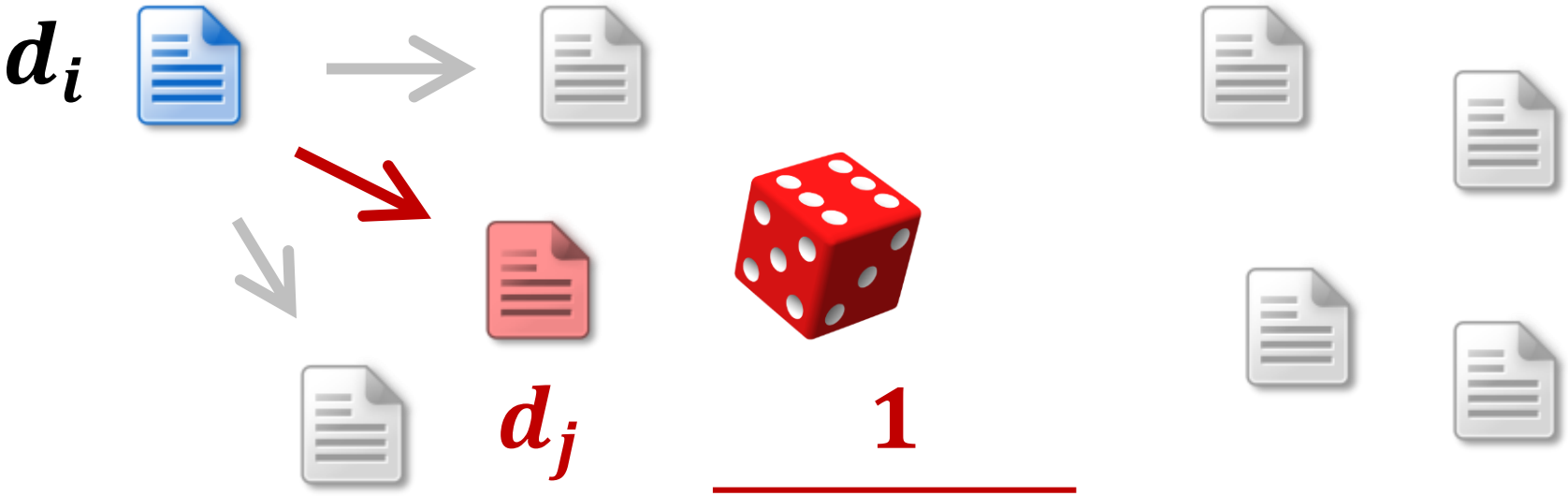


Random walk

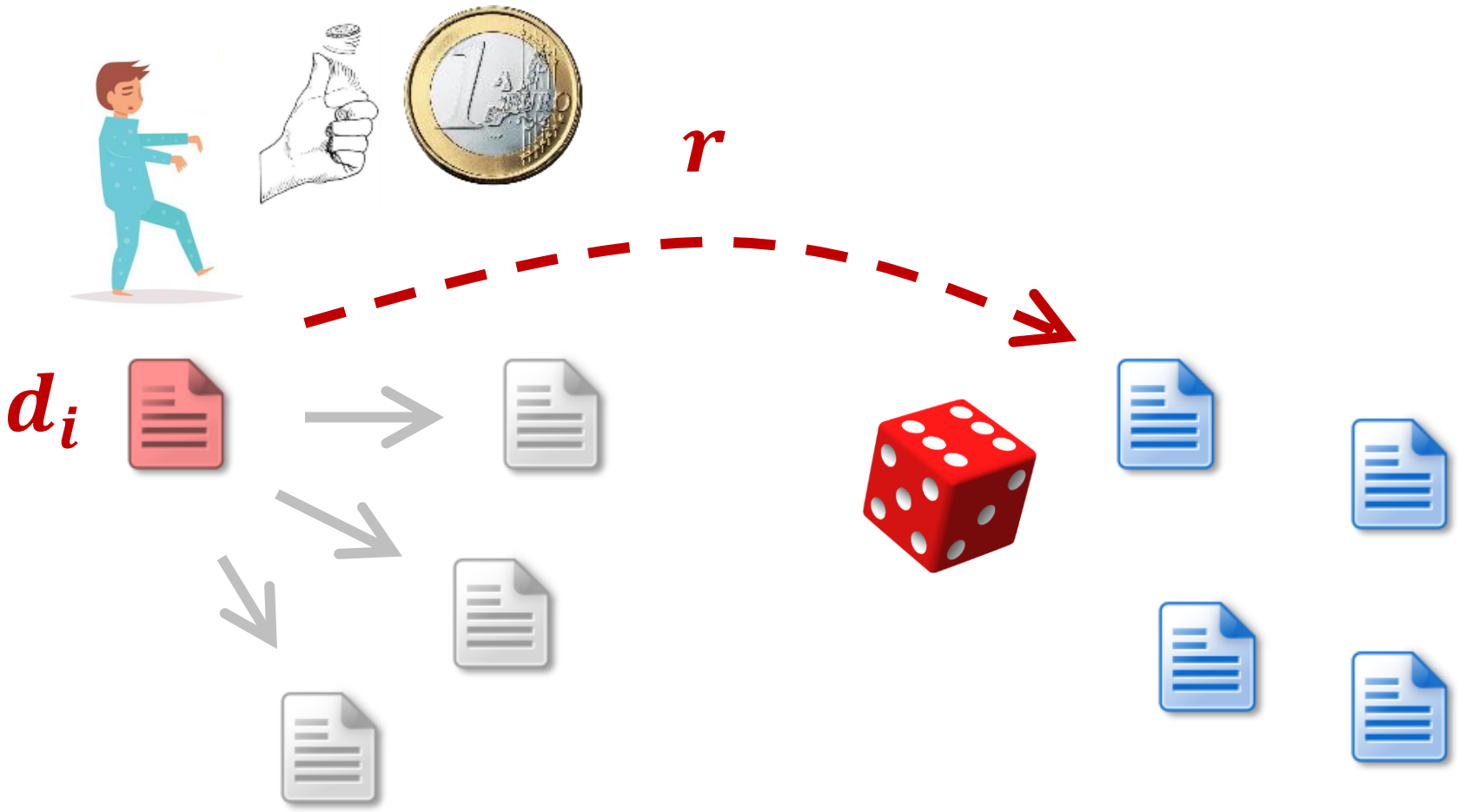


Random walk

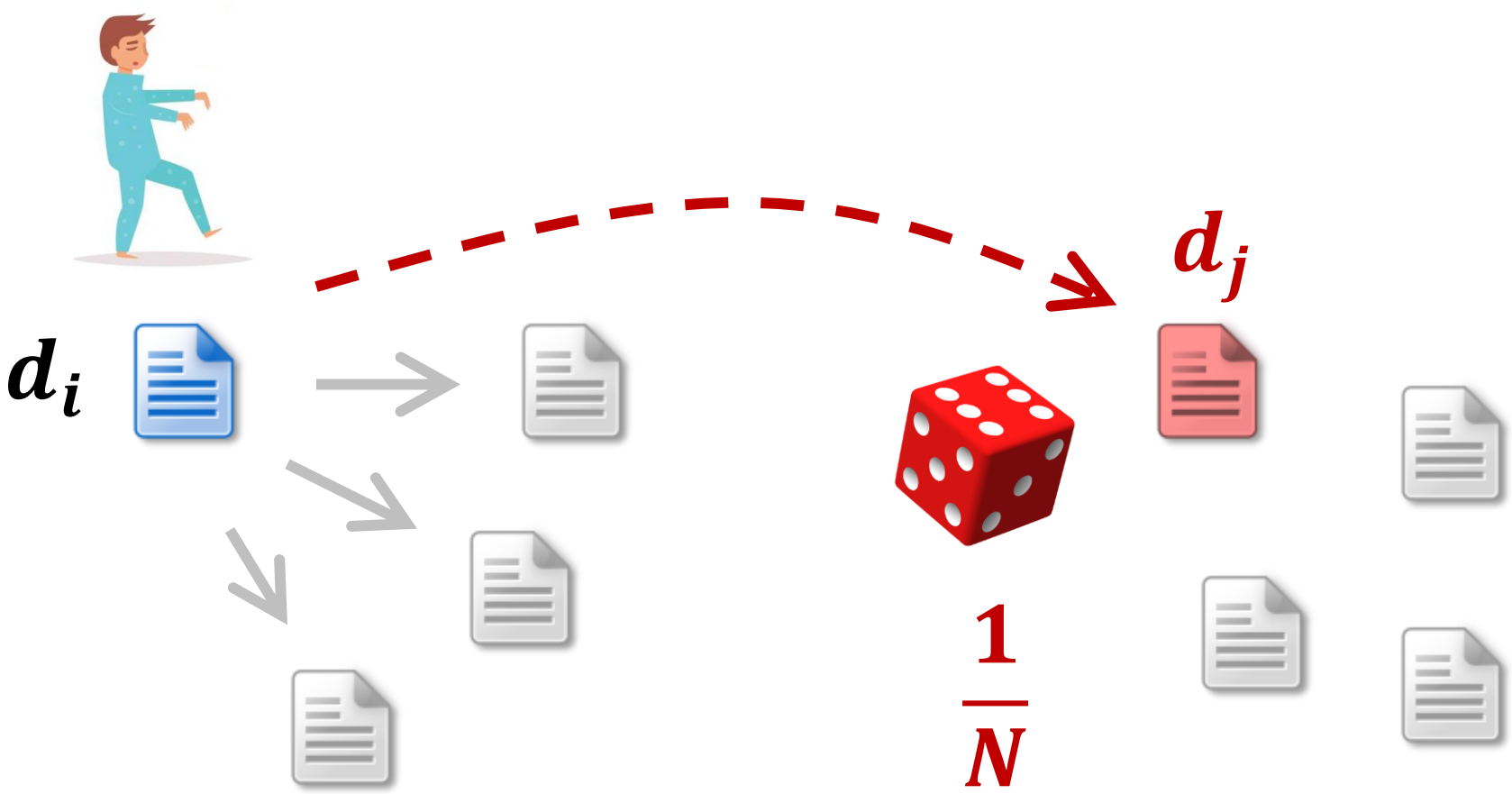
Random walk



Random walk



Random walk



PageRank: interpretación probabilística

- Navegante aleatorio
  - Empieza en una página al azar
  - Con probabilidad $1 - r$ escoge un enlace saliente al azar y lo atraviesa
  - Con probabilidad r escribe directamente la URL de una página al azar
  - Repite este comportamiento indefinidamente
- En un instante dado, ¿cuál es la probabilidad $P(d)$ de que este usuario se encuentre en una página d?
  - $P(d) \equiv$ PageRank de d
- El escenario describe un proceso estocástico que corresponde a una cadena de Markov: **random walk**
  - Las páginas son estados, el paso de una a otra son transiciones
  - La probabilidad de aterrizar en una página sólo depende de la página anterior
  - La probabilidad de transición de una página a otra se puede calcular
- Converge a una probabilidad estacionaria
- Comprobar que resulta la fórmula original…
  - Revela la necesidad de tratamiento de los nodos sumidero

Derivación probabilística

$$p(d_j|t) = \sum_i p(d_j|d_i, t-1)p(d_i|t-1) = \sum_i p(d_j|d_i)p(d_i|t-1)$$

$$p(d_j|d_i) = p(d_j|d_i, click)\, p(click|d_i) + p(d_j|d_i, teleport)\, p(teleport|d_i)$$

$$p(click|d_i) = \begin{cases} 1-r & \text{si } \#out(d_i) > 0 \\ 0 & \text{si } \#out(d_i) = 0 \end{cases} \qquad p(teleport|d_i) = \begin{cases} r & \text{si } \#out(d_i) > 0 \\ 1 & \text{si } \#out(d_i) = 0 \end{cases}$$

$$p(d_j|d_i, teleport) = p(d_j|teleport) = \frac{1}{N}$$ // Probabilidad uniforme teleport

$$p(d_j|d_i, click) = \begin{cases} \dfrac{1}{\#out(d_i)} & \text{si } d_i \to d_j \\ 0 & \text{en otro caso} \end{cases}$$ // Probabilidad uniforme entre enlaces

Derivación probabilística

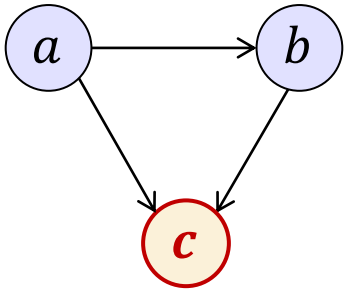
- Haciendo el desarrollo probabilístico completo, sale un término más

$$P(d_j) = \frac{r}{N} + (1-r)\left(\sum_{d_i \to d_j} \frac{P(d_i)}{\#out(d_i)} + \underbrace{\sum_{\#out(d_i)=0} \frac{P(d_i)}{N}}\right)$$

Páginas d_i que no tienen ningún enlace saliente: nodos "sumidero"

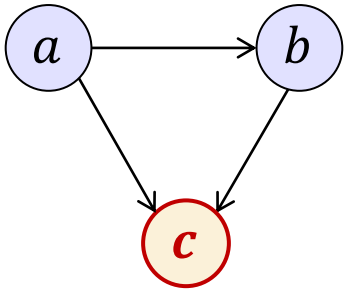
- El término aparece porque cuando el navegante llega a un sumidero, la probabilidad de que atraviese un enlace es cero (y no $1-r$)
- Ésta es la fórmula completa y más correcta de PageRank
- El término extra es lo mismo que saldría si todos los nodos sumidero tuviesen un enlace a todos los demás nodos del grafo
- Pero no quiere eso decir que haya que añadir esos enlaces al grafo!

Tratamiento de sumideros



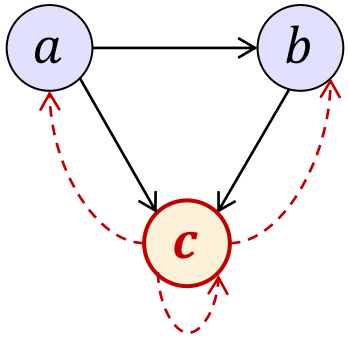
$$P(a) = r/3$$
$$P(b) = r/3 + (1-r)\,P(a)/2$$
$$P(c) = r/3 + (1-r)\big(P(a)/2 + P(b)\big)$$

Tratamiento de sumideros



$$P(a) = r/3 + (1-r)\,\boldsymbol{P(c)/3}$$
$$P(b) = r/3 + (1-r)(P(a)/2 + \boldsymbol{P(c)/3})$$
$$P(c) = r/3 + (1-r)(P(a)/2 + P(b) + \boldsymbol{P(c)/3})$$

Tratamiento de sumideros



$$P(a) = r/3 + (1-r)\,\boldsymbol{P(c)/3}$$
$$P(b) = r/3 + (1-r)(P(a)/2 + \boldsymbol{P(c)/3})$$
$$P(c) = r/3 + (1-r)(P(a)/2 + P(b) + \boldsymbol{P(c)/3})$$

- *Como si* hubiese estos enlaces
- Pero no quiere decir que haya que añadirlos!

Integración de PageRank en un sistema de búsqueda

- Elaboraciones monótonas del valor, por ejemplo:
  - No es necesario dividir por N (evita $P(d) \ll 1$ pues $\sum_d P(d) = 1$)
  - $\log P(d)$ modera la distribución tendente a power law
- Optimización de la convergencia
- Detectar spam (link farms, etc.)
- Particularizar el vector de teleportación (p.e. personalización, p.e. en Twitter WTF) y las probabilidades de transición
  - Muchas otras elaboraciones…

☞ Combinar $P(d)$ con $sim(q, d)$

  - Los tres ingredientes principales de Google: contenido + links + RankBrain