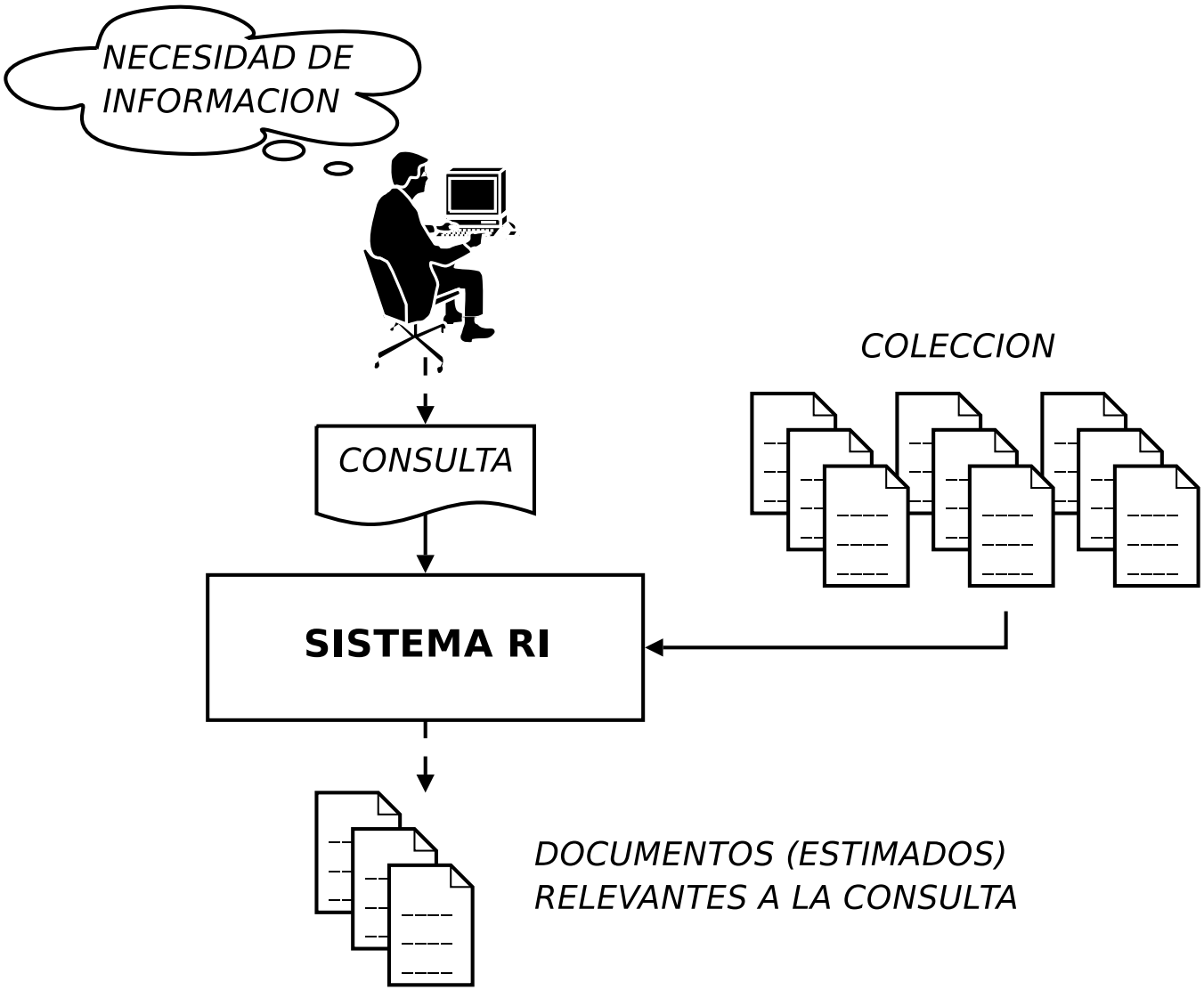
Índice

Recuperación de Información (RI)

- A.k.a. *Information Retrieval (IR)*
- **Def.:** área de la ciencia y la tecnología que trata de la representación, almacenamiento, organización y acceso a elementos de información
- **Objetivo:** dada una **colección de documentos** y una **necesidad de información** del usuario —expresada como una **consulta (query)**—, devolver un conjunto de documentos relevantes para dicha necesidad de información (i.e., cuyo contenido satisface dicha necesidad)
  - No devuelve la información deseada, sólo indica los documentos donde parece estar
- P.ej., buscadores web, filtros de *spam*, clasificadores de noticias, etc.

Proceso de RI



Terminología

- **Documento:** unidad de texto almacenada y disponible para su recuperación; p.ej., páginas web, artículos de prensa, tesis, ...
  - Granularidad variable: documento completo, capítulos, párrafos, ...
- **Colección:** repositorio de documentos en los que buscar
- **Términos:** unidades léxicas (palabras) que componen un documento/consulta
- **Consulta (query): representación** en forma de términos, de la necesidad de información del usuario
- **Relevancia de un documento:**
  - Calculada por el sistema respecto a la *consulta*
  - Juzgada por el usuario respecto a la **necesidad de información** en su cabeza (**subjetividad**)
- **Ordenación (ranking):** los documentos suelen devolverse ordenados por relevancia

Bases de Datos vs. Sistemas de RI

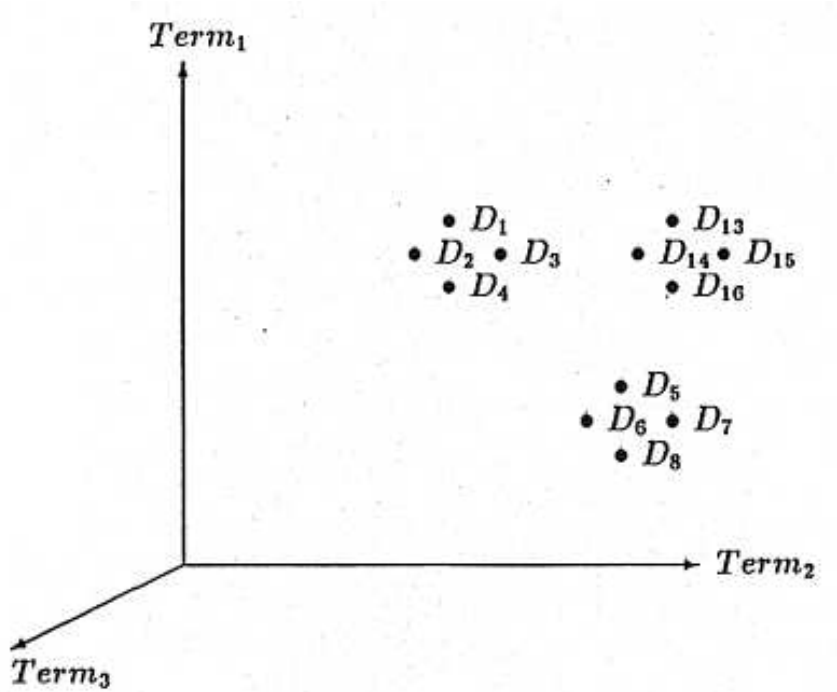
	BD	**RI**
Información	datos estructurados semántica bien definida	lenguaje natural (desestructurado) semántica ambigua
Consulta	formalizada (álgebra relacional)	lenguaje natural
Resultados	todos los relevantes (completitud) todos son relevantes (ningún error)	no necesariamente contiene errores: objetivo • maximizar relevantes devueltos • minimizar no relevantes devueltos

Tareas de RI

- **Recuperación ad hoc**:
  - P.ej., *buscadores web*
  - La colección no varía (estática) o varía "poco" (semiestática) —web
  - Consultas variables (dinámicas), puntuales y específicas
- **Categorización (clasificación de documentos):**
  - Asignar un doc. a una/más clases fijadas a priori
  - Los documentos van llegando poco a poco (colección dinámica)
  - Necesidades (perfiles o *profiles*) complejas, estables en el tiempo (estáticas), y que reflejan varias necesidades a la vez
  - 2 [sub]tareas diferenciadas:
    - Enrutamiento (*routing*): ordenación por similaridad con el perfil
    - Filtrado (*filtering*): acepta o rechaza (binario)

Tareas de RI (cont.)

- **Clustering:**



  - Generar automática. clases (*clústers*) a partir de un conjunto de docs.
    - Maximizar similaridad intra-clúster
    - Minimizar similaridad inter-clúster

- De aquí en adelante: **recuperación ad hoc**

Paradigma *Bag-of-Terms*

- **Def.:** representación de documentos/consultas como **conjunto** de *términos índice* (a.k.a. *términos de indexación* o *palabras clave*)
- **Ppo. de composicionalidad de Frege**: "la semántica de un objeto puede obtenerse a partir de la semántica de sus componentes"
  - Si una palabra aparece en un texto, dicho texto trata dicho tema
  - **Si una consulta y un documento comparten uno/más términos índice, el documento debería tratar el tema de la consulta**

Peso de un Término

- No todos los términos tendrán la misma **importancia/representatividad**: **peso (weight)** w_{ij} de un término t_i en un documento d_j
- Factores para cómputo del peso de un término:
  - Frecuencia dentro del documento
  - Distribución dentro de la colección
  - Longitud del documento
  - Forma combinarlos varía según *modelo* y formúla empleados

Peso de un Término (cont.): Frecuencia en el Documento

- Si un término aparece muchas veces en un doc., se puede suponer que el doc. está más relacionado con este tema $\longrightarrow$ **mayor peso**
- P.ej., si en un documento aparece *chocolatina* repetidamente, es lógico pensar que dicho documento habla sobre chocolatinas
- **Frecuencia del término t_i en el documento d_j (tf_{ij}):** número de veces que aparece el término t_i en el documento d_j

Peso de un Término (cont.): Distribución en la Colección

- A mayor número de documentos en los que aparece un término, menor su poder de discriminación $\longrightarrow$ **menor peso**
- P.ej., si *chocolatina* aparece en gran parte de los documentos de la colección, poco ayuda a diferenciar unos de otros
- **Frecuencia inversa de documento del término t_i (idf_i):**

$$idf_i = log \frac{N}{n_i}$$

donde N es el número total de documentos de la colección, y n_i el número de dichos documentos en los que aparece el término t_i.

  - i.e. en cuantos más documentos aparezca (mayor n_i), menor será idf_i

Peso de un Término (cont.): Longitud del Documento

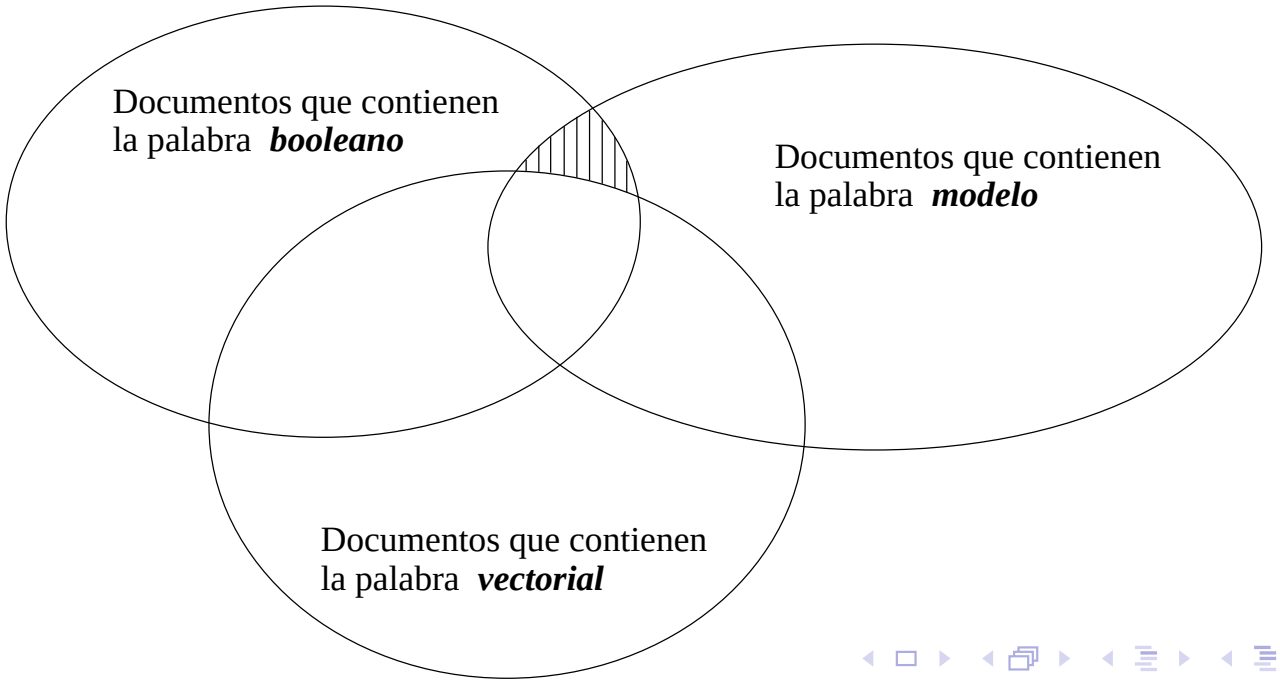
- A mayor longitud, mayor probabilidad de correspondencias, por lo que, a priori, dichos docs. partirían con ventaja $\rightarrow$ **ponderar según longitud**
- No siempre tenido en cuenta

Introducción

- Establece:
  - Cómo representar los documentos
  - Cómo representar las necesidades de información
  - Cómo compararlos

Modelo Booleano

- Base matemática: **teoría de conjuntos** y **álgebra de Boole**
- Documento: **conjunto de los términos** que contiene su texto
- Consulta: expresión booleana de términos ligados por operadores booleanos (AND, OR y NOT)
- Devuelve aquellos documentos que satisfacen la consulta
  - **Binario** (sí/no relevante): **no hay gradación de la relevancia**
- Ejemplo: *modelo* AND *booleano* AND NOT *vectorial*



Modelo Booleano (cont.)

- **Ventajas:**
  - Sencillo
  - Preciso
  - Veloz
- **Desventajas:**
  - Formalizar consulta como expresión booleana:
    - Fácil quedarse corto/largo
    - Pequeñas modificaciones en la consulta pueden provocar grandes variaciones en los resultados. P.ej., AND vs. OR
  - Binario:
    - No permite correspondencias parciales
    - Sin gradación de la relevancia/similaridad: resultados no ordenados + todos los términos valen lo mismo (como si $w_{ij} = \{0, 1\}$)
- Muy popular en el pasado. Hoy relegado a sistemas precisan correspondencias exactas: P.ej., sistemas de información legislativa

Modelo Booleano (cont.): Booleano Extendido

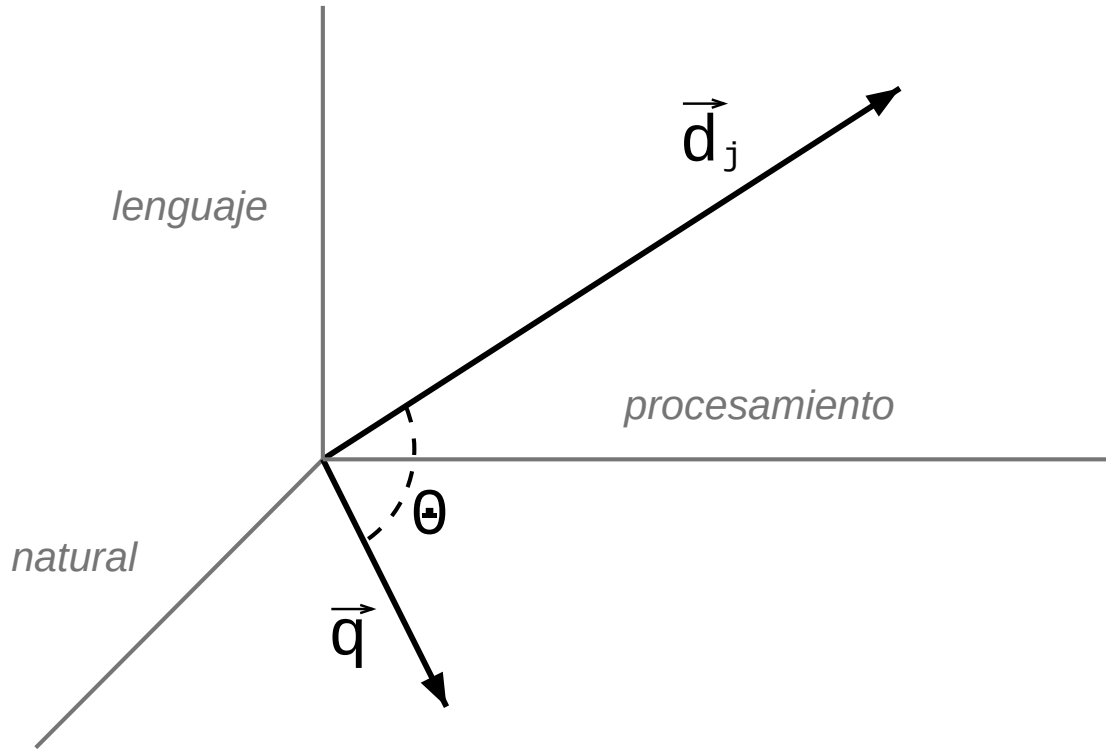
- Variante que permite **ordenación por relevancia** de los docs. Opera en dos fases:
  1. Operar estrictamente a nivel de conjs. booleanos
  2. Ordenar el conj. resultante mediante **medida de similaridad**

Modelo Vectorial

- Base matemática: **álgebra vectorial**
- **Consultas y documentos representados como vectores** en un espacio multidimensional
  - Definido por los términos del vocabulario: 1 dimensión por término
  - P.ej. Vocabulario tamaño $T \rightarrow$ espacio T-dimensional
    - Documento d_j: vector $\overrightarrow{d_j} = (w_{1j}, w_{2j}, \ldots, w_{tj})$
    - Consulta q: vector $\overrightarrow{q} = (w_{1q}, w_{2q}, \ldots, w_{tq})$

    siendo $w_{ij} \geq 0$ y $w_{iq} \geq 0$ los pesos del término t_i en el documento d_j y la consulta q, respectivamente.

Modelo Vectorial (cont.)



- Si los vectores de consulta y documento están próximos, asumimos que documento es similar a la consulta (i.e., posiblemente relevante)

Modelo Vectorial (cont.)

- **Medida proximidad:** coseno del ángulo Θ formado por los vectores:

$$sim(d_j, q) = cos(\Theta) = \frac{\overrightarrow{d_j} \bullet \overrightarrow{q}}{|\overrightarrow{d_j}| \times |\overrightarrow{q}|} = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w^2{}_{ij}} \times \sqrt{\sum_{i=1}^{t} w^2{}_{iq}}}$$

$|\overrightarrow{q}|$ es constante para una consulta dada, luego puede simplificarse (lo importante es conservar la ordenación/*ranking*)

- **Peso de un término:** clásico, esquema *tf-idf*:

$$w_{ij} = tf_{ij} \times idf_i$$

Modelo Vectorial (cont.)

- **Ventajas:**
  - Sencillo
  - Preciso (buenos resultados)
  - Permite **correspondencias parciales**
  - Permite **gradación relevancia/similaridad**: *ranking* de docs.
    - NOTA: a la hora de calcular el valor en función del cual se hace dicha ordenación, lo importante no es tanto la *magnitud* de ese valor, sino la *relación de orden* que se establece. De ahí que a la hora de calcularlo se apliquen diversas transformaciones y simplificaciones siempre y cuando mantengan la ordenación.
- **Referencia para comparación** de otros sistemas

(Familia) Modelos Probabilísticos

- Modelos booleano y vectorial:
  - Cuentan con una base formal, pero la forma de calcular y ordenar el conjunto resultado, ¿es la más adecuada? –No lo sabemos, no hay resultados teóricos que lo afirmen (mero "ensayo y error").
- **Modelos probabilísticos** (familia):
  - **Teoría de probabilidades** como marco formal de trabajo.
  - Basados en el **Ppo. de Ordenación por Probabilidad**:

    "La recuperación óptima es aquélla en la que los documentos son devueltos ordenados en orden decreciente de acuerdo a su probabilidad de relevancia respecta a la consulta."
  - **Probabilidad de relevancia** vs. medida similaridad.

(Familia) Modelos Probabilísticos (cont.)

- Sean:

$P(R\|d_j, q)$	probabilidad de que un documento d_j **sea relevante** para una consulta q
$P(\overline{R}\|d_j, q)$	probabilidad de que un documento d_j **no sea relevante** para una consulta q

  - **Conjunto resultado óptimo:** documento **es relevante** si $P(R|d_j, q) > P(\overline{R}|d_j, q)$
  - **Ordenación óptima:** documentos **devueltos por orden de probabilidad** de relevancia $P(R|d_j, q)$
- Calcularemos dichas probabilidades en función de los términos índice que contienen él y el resto de la colección

(Familia) Modelos Probabilísticos (cont.)

(1) Modelo de Independencia Binaria

- Modelo básico
- Resultados ordenados por su **Retrieval Status Value (RSV)**:

$$sim(d_j, q) = RSV_{d_j q} = \sum_{\substack{t_i \in Q \\ t_i \in D_j}} c_i$$

donde c_i se denomina **peso Robertson-Sparck Jones**:

$$c_i \approx \log \frac{(|V_i| + 0.5) \,/\, (|V| - |V_i| + 0.5)}{(df_i - |V_i| + 0.5) \,/\, (N - df_i - |V| + |V_i| + 0.5)}$$

con:

- $|V|$, nº docs. relevantes devueltos
- $|V_i|$, nº docs. relevantes devueltos contienen término t_i
- N, nº docs. en colección
- df_i, nº docs. en colección contienen término t_i

(Familia) Modelos Probabilísticos (cont.)

(2) Okapi BM25

- **Evolución** del modelo básico para tener en cuenta:
  - N° apariciones del término en el documento
  - Longitud del documento

$$sim(d_j, q) = \sum_{i=1}^{t} c_i \times \frac{(k_1 + 1) \times tf_{ij}}{K + tf_{ij}} \times \frac{(k_3 + 1) \times tf_{iq}}{k_3 + tf_{iq}}$$

donde t es el número de términos que componen el vocabulario
c_i es el *peso Robertson-Sparck Jones* visto anteriormente
K es calculado como $K = k_1 \times ((1 - b) + b \times dl_j / avdl)$
k_1, b y k_3 son parámetros constantes en función de la consulta/colección
tf_{ij} es la frecuencia del término t_i en el documento d_j
tf_{iq} es la frecuencia del término t_i en la consulta q
dl_j es la longitud del documento d_j
$avdl$ es la longitud media de los documentos de la colección

- Referencia para comparación de otros sistemas

(Familia) Modelos Probabilísticos (cont.)

(3) **Paradigma DFR** *(Divergence From Randomness)*

- No es un modelo, sino una **metodología** para construir modelos de recuperación
- **Diferencias** respecto modelos probabilísticos "clásicos":
  - *Metodología*, no modelo.
  - *No paramétrico*: no hay parámetros a ajustar (ej. k_1, k_3 y b en BM25).
  - *Ganancia de información* vs. probabilidad de relevancia (vs. medida de similaridad)
- **Idea**:
  - Asumir distribución aleatoria de los términos en los docs.
  - Si una palabra aparece en un doc. mucho más de lo esperado, ese doc. trata ese tema.
- Sistema por excelencia: Terrier (`http://ir.dcs.gla.ac.uk/terrier/`)

Proceso

1. **Normalización** (*conflation*):
   i. Tokenización
   ii. Eliminación de *stopwords*
   iii. Paso a minúsculas y eliminación de signos ortográficos
   iv. *Stemming*
   v. Selección de términos índice
2. **Indexación**

Normalización

- Proceso de generación de términos índice de docs./consultas mediante transformaciones sucesivas del texto (*operaciones de texto*)
- **Objetivo:** reducción del texto a una **forma canónica** para facilitar las correspondencias

Normalización (cont.): Tokenización

- **Identificación de las palabras** del texto:
  - **Def. palabra**: "secuencia de caracteres de palabra delimitada por separadores"
  - **Problema:** la noción ortográfica no siempre coincide con la noción lingüística: contracciones, enclíticos, compuestos, locuciones, etc.
- Herramientas sencillas y rápidas

Normalización (cont.): Eliminación de *Stopwords*

- **Def. stopwords:** palabras de escasa o nula utilidad d.p.d.v. recuperación:
  - Escaso contenido semántico; p.ej.: artículos, preposiciones, etc.
  - Frecuencia excesiva (nula capacidad discriminante); p.ej.: formas verbales de *ser* o *estar*
- Su eliminación permite un **considerable ahorro de recursos de almacenamiento**:
  - Mínima parte del vocabulario pero gran parte de los términos del texto (*Ley de Zipf*)
  - Listas preestablecidas de *stopwords*
- En las consultas, eliminar también información de *metanivel*

  P.ej., *"Encuentre los documentos que describan ..."*

Normalización (cont.): Paso a Minúsculas y Eliminación de Signos Ortográficos

- P.ej., ” ... *Paso a Minúsculas ...* ” → ” ... *paso a minusculas ...* ”
- **Objetivo:** facilitar las correspondencias

Normalización (cont.): *Stemming*

- **Def.:** reducción de una palabra a su *stem* o raíz supuesta eliminando su terminación según una **lista de sufijos**
  - *Stem* o raíz contiene semántica básica

$$\left.\begin{array}{l}\texttt{reloj}\\ \texttt{relojes}\\ \texttt{relojero}\end{array}\right\} \rightarrow \texttt{reloj-}$$

- **Objetivo:**
  - Principal: permitir correspondencias entre variantes
  - Secundario: reducir recursos almacenamiento (reducir vocabulario)
- *Stemmer* de Porter
  - Demo: `http://maya.cs.depaul.edu/~classes/ds575/porter.html`
  - `Snowball` (descargables): `http://snowball.tartarus.org`
- Nivel de normalización
  - *Superficial*: sólo morfología flexiva simplificada; p.ej., sólo plurales
  - *Profundo*: flexiva y derivativa (agresivo); p.ej., Porter

Normalización (cont.): Selección de Términos Índice

- Los términos resultantes son tomados como términos índice asociándoseles, si es necesario, el *peso* correspondiente
- Representación:
  - **A texto completo**
  - Selección de términos: manual/automáticamente
    - P.ej., indexación con vocabulario controlado

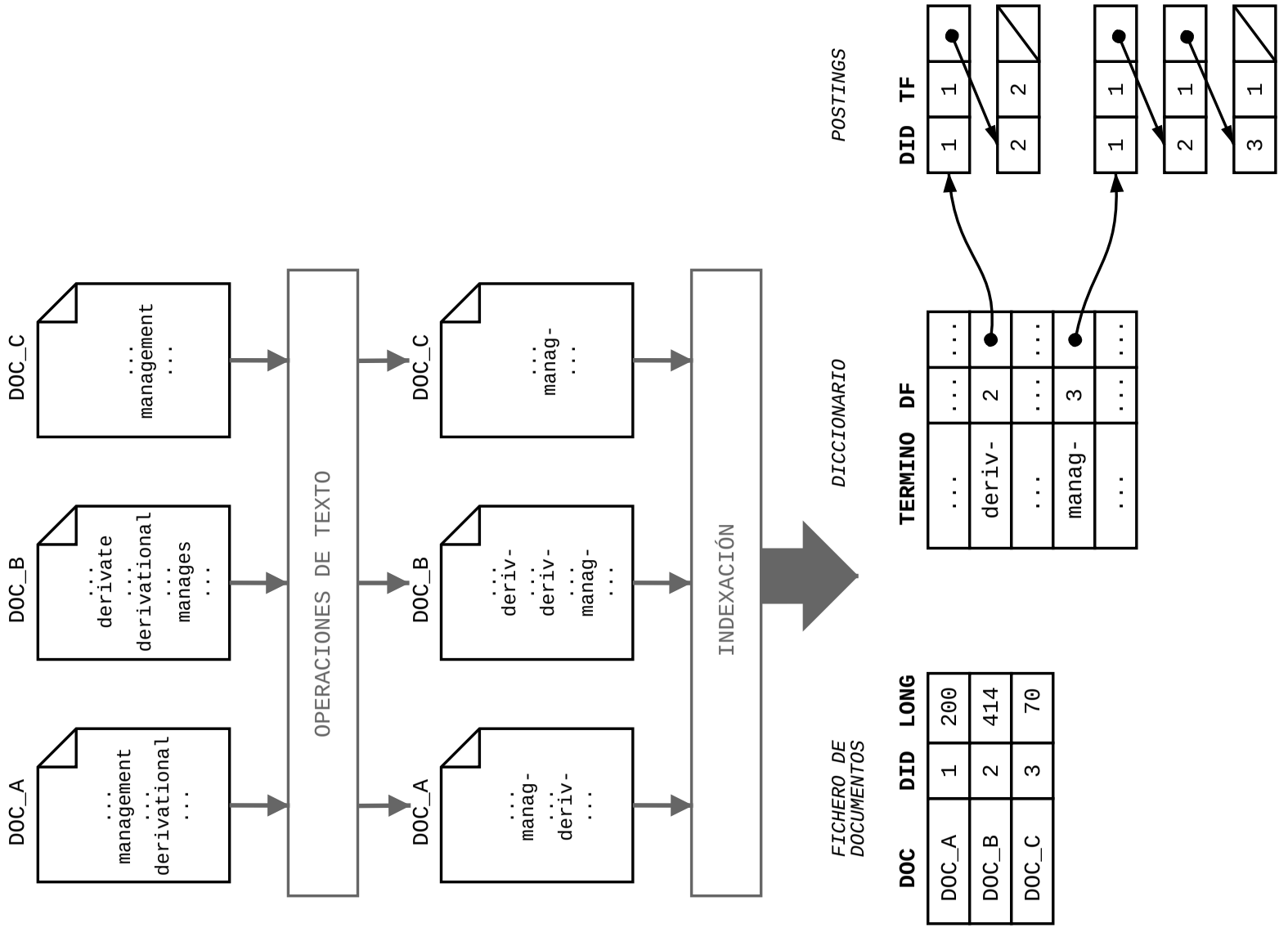
Indexación: Componentes Sistema de RI

- **Índice invertido:** estructuras auxiliares donde almacenar las representaciones de los docs.
  - **Objetivo:** acelerar la búsqueda
  - **Componentes:**
    - **Vocabulario/diccionario:** lista de términos índice en la colección
    - **Postings:** lista de docs. en los que aparece cada término
    - **Fichero de documentos:** lista de documentos del sistema

    Asociado a cada entrada (término/*posting*/documento) se almacenan los datos necesarios para el cálculo del peso/relevancia

- **Motor de indexación:** componente software encargado del manejo de los índices

Indexación (cont.): Generación del Índice



ejemplo_Generacion_Indice.pdf

Introducción

- **Proceso:**
  1. El usuario plasma su **necesidad de información** en una **consulta**
  2. El sistema obtiene la **representación interna** de la consulta aplicando *operaciones de texto* (las mismas que durante *indexación*)
  3. El sistema compara la **representación interna** obtenida con los documentos indexados
- El sistema parte de la consulta formulada por el usuario. **Peligro:**
  - Si la consulta está formulada **de forma incorrecta o insuficiente**
  - Si usuario y autor del documento emplean **términos diferentes (variación lingüística)**
- **Solución paliativa:** *expansión de consultas*

Expansión de Consultas

- A.k.a. **query expansion**
- **Def.:** procesos automáticos/semiautomáticos para la reformulación/refinamiento de la consulta inicial mediante la adición de nuevos términos
  - Relacionados con los términos iniciales
  - Asociados a los documentos [supuestamente] relevantes

Expansión de Consultas (cont.): Mediante Tesauros

- **Def. tesauro (thesaurus):** base de datos lexicográfica que almacena una representación jerárquica de un lexicón de acuerdo a las relaciones semánticas existentes entre sus palabras
  - P.ej., WordNet (más utilizado)
- **Proceso:** reformular consulta inicial
  - Manualmente: navegando por su estructura eligiendo los términos con los que expandir
  - Automáticamente: p.ej., expandiendo un término con sus sinónimos
- Por lo general **no se logran mejoras** en los resultados
  - Introducción de **ruido** durante la expansión: necesario aplicar técnicas de *desambiguación del sentido de las palabras (WSD)*

Expansión de Consultas (cont.): Mediante Realimentación

- A.k.a. **relevance feedback**
- **Proceso:**
  1. Se lanza consulta inicial contra el sistema
  2. Se toman los primeros documentos devueltos por el sistema
     - Manual: usuario los examina y determina cuáles son relevantes
     - Automático: los *n* primeros documentos se suponen relevantes (*expansión ciega* o *por pseudo-relevancia*)
  3. Partiendo de los documentos estimados como relevantes:
     - **Se añaden nuevos términos** que aparezcan en ellos
     - **Se modifican los pesos** de los términos de la consulta inicial
- **Buen comportamiento** general (uso ampliamente extendido)

Colecciones de Referencia/Evaluación

- Composición: 3 elementos
  1. Documentos
  2. Consultas
  3. Lista de los documentos relevantes para cada consulta
- Más importantes (asociadas a instituciones/congresos):
  - TREC
  - CLEF
  - NTCIR
  - FIRE

Colecciones de Referencia (cont.): TREC

- Text REtrieval Conference (`http://trec.nist.gov/`)
  - National Institute of Standards and Technology (NIST): Depto. de Comercio USA
  - Defense Advanced Research Projects Agency (DARPA)
- **Objetivo:** facilitar infraestructura, herramientas y metodologías para la **evaluación a gran escala** de sistemas de RI (**inglés**)
  - Diferentes sesiones (*tracks*): *ad hoc*, *terabyte (web)*, filtrado, etc.
  - Consultas: ∼50 por año y *track*
  - Colecciones:

Ad hoc track:	*L.A. Times*	131,896 artículos	475 MB	media 527 words
Terabyte track:	GOV2	25,205,179 webs (`.gov`)	426 GB	media 17.7 KB

  - Evaluación:
    - **Técnica pooling:** revisan manualmente K *top* docs. (K=100)
    - **Def. relevancia:**

      "Si estuviera redactando un informe sobre el tema del topic en cuestión y pudiese usar para dicho informe la información contenida en el documento examinado, entonces dicho documento será considerado relevante"

Colecciones de Referencia (cont.): CLEF

- Cross-Language Evaluation Forum (`http://www.clef-campaign.org/`)
- TREC europeo:
  - Lenguas europeas (y otras):
    - Colecciones: inglés, francés, español, alemán, holandés, italiano, portugués, búlgaro, checo, húngaro, ruso, sueco y finlandés
    - Topics (a mayores): chino, japonés, griego, arameo, hindi, bengalí, marathi, oromo, tamil, telugu
  - *Tracks* mono e multilingüe (*Recuperación de Información Multilingüe*)

Colecciones de Referencia (cont.): Ejemplo de *Topic*

```
<top>
<num> C044 </num>
<ES-title> Indurain gana el Tour </ES-title>
<ES-desc> Reacciones al cuarto Tour de Francia ganado por Miguel Indurain.
</ES-desc>
<ES-narr> Los documentos relevantes comentan las reacciones a la cuarta
victoria consecutiva de Miguel Indurain en el Tour de Francia. Los
documentos que discuten la relevancia de Indurain en el ciclismo mundial
después de esta victoria también son relevantes. </ES-narr>
</top>
```

- A partir de los cuales se generan (automática o manualmente) las consultas finales
- **3 elementos:**
  - *Título*: breve título
  - *Descripción*: frase de descripción
  - *Narrativa*: pequeño texto especificando los criterios que utilizarán los revisores para establecer la relevancia de un documento respecto a la consulta

Colecciones de Referencia (cont.): Ejemplo de *Documento*

```
<DOC>
<DOCNO>EFE19940101-00002</DOCNO>
<DOCID>EFE19940101-00002</DOCID>
<DATE>19940101</DATE>
<TIME>00.34</TIME>
<SCATE>VAR</SCATE>
<FICHEROS>94F.JPG</FICHEROS>
<DESTINO>ICX MUN EXG</DESTINO>
<CATEGORY>VARIOS</CATEGORY>
<CLAVE>DP2404</CLAVE>
<NUM>100</NUM>
<PRIORIDAD>U</PRIORIDAD>
<TITLE> IBM-WATSON
FALLECIO HIJO FUNDADOR EMPRESA DE COMPUTADORAS
</TITLE>
<TEXT> Nueva York, 31 dic (EFE).- Thomas Watson junior, hijo del fundador
de International Business Machines Corp. (IBM), falleció hoy,
viernes, en un hospital del estado de Connecticut a los 79 años de
edad, informó un portavoz de la empresa.
Watson falleció en el hospital Greenwich a consecuencia de
complicaciones tras sufrir un ataque cardíaco, añadió la fuente.
El difunto heredó de su padre una empresa dedicada principalmente
a la fabricación de máquinas de escribir y la transformó en una
compañía líder e innovadora en el mercado de las computadoras. EFE
PD/FMR
01/01/00-34/94
</TEXT>
</DOC>
```

Colecciones de Referencia (cont.): Ejemplo de *Qrel*

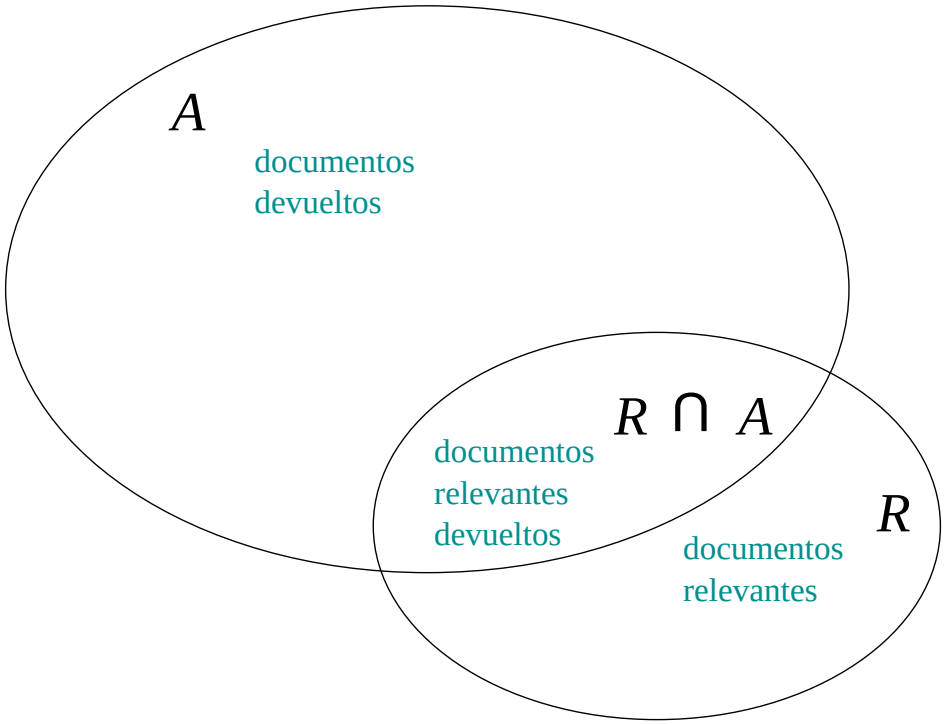
```
<QID> <ITER> <DOCNO> <REL>

...

44 0 EFE19940722-13111 0
44 0 EFE19940722-13237 0
44 0 EFE19940722-13274 0
44 0 EFE19940723-13494 1
44 0 EFE19940723-13513 1
44 0 EFE19940723-13688 0
44 0 EFE19940724-13915 0
44 0 EFE19940724-14076 1
44 0 EFE19940724-14077 1
44 0 EFE19940724-14084 1
44 0 EFE19940724-14086 1
44 0 EFE19940724-14093 1
44 0 EFE19940724-14098 1
44 0 EFE19940724-14101 0
44 0 EFE19940724-14104 1
44 0 EFE19940724-14130 1

...
```

Métricas de Evaluación



- D, conjunto de **documentos** en la colección
- R, conjunto de **documentos relevantes** en la colección
- $\bar{R} = D - R$, conjunto de **documentos no relevantes** en la colección
- A, conjunto de **documentos recuperados** por el sistema
- $A \cap R$, conjunto de **documentos relevantes recuperados** por el sistema

Métricas de Evaluación (cont.)

NOTA: A nivel de **consulta** o **conjunto de consultas** (media)

(1) Medidas que NO dependen de la ordenación de los documentos devueltos:

- **Precisión (precision):** porcentaje de documentos recuperados que son relevantes:

$$precision \;\; Pr = \frac{A \cap R}{A}$$

  Capacidad para recuperar sólo documentos relevantes.

- **Cobertura/exhaustividad (recall):** porcentaje de documentos relevantes que son recuperados:

$$recall \;\; Re = \frac{A \cap R}{R}$$

  Capacidad para recuperar todos los documentos que son relevantes.

Métricas de Evaluación (cont.)

- **Medida-F (F-measure):** pondera ambas conforme a un parámetro $\beta \in [0, \infty)$

$$F = \frac{(\beta^2 + 1) \times Re \times Pr}{Re + \beta^2\ Pr} \quad \text{con} \quad \beta^2 = \frac{1 - \alpha}{\alpha} \quad \text{y} \quad \alpha \in [0, 1]$$

si β=1 (F_1) ambas se ponderan igual

$$F_1 = \frac{2 \times Re \times Pr}{Re + Pr}$$

- **Fall-out:** porcentaje de documentos no relevantes que son recuperados

$$fallout = \frac{A \cap \bar{R}}{\bar{R}}$$

Métricas de Evaluación (cont.)

(2) Medidas que SÍ dependen de la ordenación de los documentos devueltos:

- **Precisión a los *n* documentos recuperados**, mide la precisión obtenida cuando se han recuperado los 4, 10, 30, 100, 500 primeros documentos
- **R-precisión**, mide la precisión obtenida cuando se ha recuperado un número de documentos igual al número de documentos relevantes
- **Precisión media no interpolada**, calcula la media de las precisiones de todos los puntos en que se encuentran documentos relevantes
- **Precisión media interpolada en 11 puntos**, calcula la media de las precisiones en los puntos en que se alcanza el 0%, 10%, 20%, . . . , 100% de los documentos relevantes

Ejemplo

	Ranking 1	Ranking 2	Ranking 3
	d1	d10	d6
	d2	d9	d1
	d3	d8	d2
	d4	d7	d7
	d5	d6	d8
	d6	d5	d3
	d7	d4	d4
	d8	d3	d5
	d9	d2	d9
	d10	d1	d10
precisión	0,5	0,5	0,5
precisión R	1	0	0,4
precisión no interpolada	1	0,3544	0,5726
precisión interpolada 11 pt.	1	0,5	0,6440

PLN & IR

- **Def. Procesamiento del Lenguaje Natural (PLN):** tratamiento computacional del lenguaje humano
  - Objetivo: computadora comprenda el lenguaje humano
- **Objetivo IR:** dada una **colección de documentos** y una **necesidad de información** del usuario —expresada como una **consulta (query)**—, devolver un conjunto de documentos relevantes para dicha necesidad de información (i.e., cuyo contenido satisface dicha necesidad)
- **IR como tarea de NLP:** "comprender" el contenido de los documentos

Keyword Retrieval Hypothesis

- **Def.:** representación de documentos/consultas como **conjunto** de *términos índice* (a.k.a. *términos de indexación* o *palabras clave*)
- **Ppo. de composicionalidad de Frege**: "la semántica de un objeto puede obtenerse a partir de la semántica de sus componentes"
  - Si una palabra aparece en un texto, dicho texto trata dicho tema
- **Hipótesis de recuperación por palabras clave** (*keyword retrieval hypothesis*): "si una consulta y un documento comparten términos índice, es que el documento debe tratar el tema de la consulta"
  - **Problema: es insuficiente**, el lenguaje no es un mero repositorio de palabras
    - Comunicar conceptos, entidades, y relaciones de múltiples maneras
    - Las palabras se combinan en unidades lingüísticas de mayor complejidad, cuyo significado no siempre viene dado por el significado de sus palabras componente

Variación Lingüística

- **Principal problema de la Minería de Textos** (*Text Mining*): variación lingüística
  - El mismo concepto puede expresarse de diferentes maneras (y viceversa)
  - Impide establecer correspondencias
  - Introduce ruido
- Diferentes niveles de variación:
  - Morfológica: modificaciones **flexivas** y **derivativas** (dañan *recall*)

    `cantas` $\sim$ `cantó` `cantar` $\sim$ `cantante`
    - Dependiente de la **complejidad morfológica** del lenguaje
  - Semántica: **polisemia** (dañan *precision*)

    `banda (de música)` $\neq$ `banda (franja)`
  - Léxica: **sinonimia** (dañan *recall*)

    `rápido` $=$ `veloz`
  - Sintáctica: modificaciones de la **estructura sintáctica** (dañan ambas)

    `Juan atacó a Pepe` $\neq$ `Pepe atacó a Juan`

    `Juan atacó a Pepe` $=$ `Pepe fue atacado por Juan`
  - Pueden darse simultáneamente: p.ej. morfo-sintáctica:

    `cambio climático` $=$ `cambio del clima`

Tratamiento de la Variación Lingüística

- **Solución**: técnicas de NLP
- Dos enfoques:
  - **Normalización:** reducir las diferentes variantes de un término a una *forma canónica* común
    - Ej. *stemming*
  - **Expansión:** añadir a la consulta variantes de sus términos originales
    - Ej. añadir sinónimos

Tratamiento de la Variación Morfológica: *Stemming*

- **Def.: reducción de una palabra a su stem o raíz supuesta** eliminando su terminación según una lista de sufijos y reglas de transformación (i.e. normalización)
  - *Stem* contiene semántica básica

$$\left.\begin{array}{l}\texttt{reloj}\\ \texttt{relojes}\\ \texttt{relojero}\end{array}\right\} \rightarrow \texttt{reloj-}$$

- **Objetivo:**
  - Principal: permitir correspondencias entre variantes (incrementar *recall*)
  - Secundario: reducir recursos almacenamiento (reducir vocabulario)
- *Stemmer* de Porter
  - Demo: `http://maya.cs.depaul.edu/~classes/ds575/porter.html`
  - `Snowball` (descargables): `http://snowball.tartarus.org`
- Nivel de normalización
  - *Superficial*: sólo morfología flexiva simplificada; p.ej., sólo plurales
  - *Profundo*: flexiva y derivativa (agresivo); p.ej., Porter

Tratamiento de la Variación Morfológica: *Stemming* (cont.)

- Ventajas
  - Simplicidad
- Desventajas:
  - Rendimiento **dependiente de la morfología del idioma**: problemas con lenguas de morfología compleja y muchas irregularidades. Ej. español:
    - Adjetivos/nombres: +20 grupos variación género +10 grupos número
    - Verbos: 3 regulares, ±40 irregulares; 118 formas flexivas cada grupo
  - Pérdida de información de cara a procesamiento futuro: produce formas no lingüísticas

$$\texttt{recognized} \rightarrow \texttt{recogn-}$$

  - *Over-stemming*: palabras no relacionadas dan igual *stem*

$$\left.\begin{array}{l}\texttt{general}\\ \texttt{generous}\end{array}\right\} \rightarrow \texttt{gener-}$$

  - *Under-stemming*: palabras sí relacionadas dan *stems* diferentes

$$\begin{array}{ll}\texttt{recognize} & \rightarrow \texttt{recogn-}\\ \texttt{recognition} & \rightarrow \texttt{recognit-}\end{array}$$

Tratamiento de la Variación Morfológica (cont.): Flexión

- **Expansión flexiva** (expansión): expandir la consulta con formas flexionadas:

$$\texttt{gato} \longrightarrow \begin{cases} \texttt{gata} \\ \texttt{gatos} \\ \texttt{gatas} \end{cases}$$

- **Lematización** (normalización): sustituir palabra por su **lema**

Ej. $\texttt{gatas} \rightsquigarrow \texttt{gato}$

  - Permite abordar la variación **flexiva**
  - Mejora resultados con idiomas de morfología compleja. Ej. lenguas romances
  - Reduce la pérdida de información: siempre obtenemos palabras
  - Google integra dicha capacidad

Tratamiento de la Variación Morfológica (cont.): Derivación

- **Análisis morfológico:**
  - Permite abordar la variación **derivativa**
  - Necesario con las lenguas de morfología más compleja. Ej. árabe
  - Aplicaciones:
    - **Stemming lingüístico** (normalización): obtener la raíz [lingüística] mediante el análisis
    - **Clustering derivativo** (normalización): un conjunto de palabras relacionadas derivativamente son reducidas a un mismo término base común

$$\left.\begin{array}{l}\texttt{sabotear}\\ \texttt{saboteador}\\ \texttt{sabotaje}\end{array}\right\} \longrightarrow \texttt{sabotaje}$$

    - **Expansión derivativa** (expansión): expandir la consulta con palabras derivadas

$$\texttt{sabotaje} \longrightarrow \left\{\begin{array}{l}\texttt{sabotear}\\ \texttt{saboteador}\end{array}\right.$$

Tratamiento de la Variación Léxica y Semántica

- Muy conectadas entre sí
- Requeriría aplicar técnicas de *desambiguación del sentido* (*Word Sense Disambiguation (WSD)*)
  - Necesaria alta efectividad: ∼90% (¿60%?)
- Se suele emplear **WordNet/EuroWordNet**
- Aproximaciones:
  - **Clustering semántico** (normalización): un conjunto de palabras relacionadas semánticamente son reducidas a un mismo término base común
    - Ej. indexación por sentidos (*synsets*) en lugar de palabras
  - *Fuzzy matching* en base a **distancias conceptuales**. Ej.:

$$sim(x,y) \rightarrow \begin{cases} 1 & x = y \\ 0.9 & x \in SYN(y) \\ 0.7^n & x \in HYPON_n(y)^\dagger \\ 0.5^n & x \in HYPER_n(y) \\ 0 & \text{resto} \end{cases}$$

† $x \in HYPON_n(y)$ si x es un hipónimo de y con n niveles de diferencia

Tratamiento de la Variación Léxica y Semántica (cont.)

- Aproximaciones (cont.):
  - **Expansión semántica** (expansión): expandir la consulta con términos semánticamente relacionados. Ej. sinónimos:

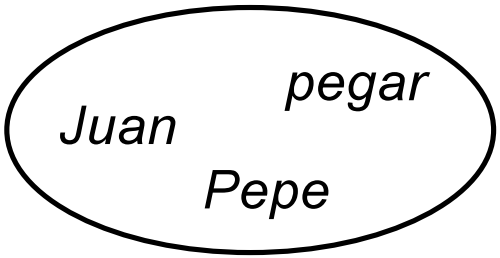
$$\texttt{bonito} \longrightarrow \begin{cases} \texttt{hermoso} \\ \texttt{bello} \end{cases}$$

    - En ocasiones la expansión es **ponderada**
    - **Poco efectivo** salvo con consultas cortas o incompletas
    - Integrado en Google:
      — Implícitamente:
      $\texttt{mantequilla de cacahuete} \leftrightarrow \texttt{crema de cacahuete}$
      — Explícitamente: operador $\sim$

$$\texttt{ape} \longrightarrow \begin{cases} \texttt{monkey} \\ \texttt{gorilla} \\ \texttt{chimpanzee} \end{cases}$$

Tratamiento de la Variación Sintáctica

- **Problema:** *bag-of-terms* insuficiente:



$$\begin{cases} \textit{Juan pegó a Pepe ?} \\ \textit{Pepe pegó a Juan ?} \end{cases}$$

- **2 aproximaciones:**
  1. Representaciones complejas en base a estructuras sintácticas: árboles y grafos
     - Coste muy alto: inadecuado para uso práctico
  2. **Frases como términos índice:**
     - **Hipótesis:** frases denotan conceptos/entidades más significativos que las palabras
     - Términos más precisos y descriptivos
     - Uso combinado con palabras

Tratamiento de la Variación Sintáctica (cont.): Identificación y Extracción

- Técnicas estadísticas
  - Secuencias de palabras coocurren frecuentemente
  - Análisis estadístico (frecuencias, coocurrencias, etc.)
  - No base lingüística (a veces resultados extraños)
  - Mayor simplicidad
- Sintácticas
  - Secuencias de palabras satisfacen relaciones sintácticas
  - Análisis sintáctico (complejidad diversa)
  - Sí base lingüística (teóricamente superiores)
  - Mayor complejidad
- Aproximar sintaxis mediante distancias
  - Palabras cercanas se suponen relacionadas sintácticamente

Tratamiento de la Variación Sintáctica (cont.): Representación y Correspondencias

- Como conjuntos de palabras
- Almacenar árbol de análisis
  - Técnicas de comparación de árboles: gran complejidad
- Almacenar sólo las relaciones sintácticas interesantes
  - Sustantivo–modificador
  - Sujeto–verbo
  - Verbo–Objeto
  - ...

Tokenización

- En cualquier aplicación de *Text Mining* es necesario **dividir el texto en** unidades lingüísticamente significativas (i.e. **palabras**) antes de procesarlo
- Generalmente obviado: los sistemas de IR emplean **tokenizadores muy sencillos** similares a los de compiladores de lenguajes de programación
- **Problemas**:
  - **Concepto lingüístico** de palabra no coincide con concepto ortográfico. Ej.:

    locuciones: `sin embargo`
    compuestos: `lebensversicherungsgesellschaftsangesteller`
    (empleado de cía. de seguros)

  - **Ambigüedad** en la tokenización. Ej.:

`sin embargo` $\longrightarrow$ { "`No tenía ganas, sin embargo lo hizo`" / "`Las relaciones prosiguieron sin embargo económico alguno`" }

`ténselo` $\longrightarrow$ { `ten+se+lo` (tener) / `tense+lo` (tensar) }

  - **Muy dependendiente** de los fenómenos lingüísticos de cada idioma
- **Solución: tokenizadores de base lingüística**

Segmentación de Compuestos

- **Compuestos**: palabras concatenación de palabras ("base")
  - En ocasiones existen interfijos que las conectan.
    Ej. `spokesman`
  - Comunes en alemán, holandés, finés, sueco...

Ej. alemán: `lebensversicherungsgesellschaftsangesteller` *(empleado de cía. de seguros)*

- **Algoritmos de segmentación** de compuestos:
  - **Basados en reglas**: reglas de descomposición creadas manualmente
    - Requieren un profundo conocimiento del lenguaje
  - **Basados en diccionarios**:
    - Lexicones con palabras válidas del lenguaje como referencia
    - Intentan trocear en palabras contenidas en esos diccionarios
    - Problema: los lexicones no son exhaustivos (variantes, desconocidas, etc.)

Segmentación de Compuestos (cont.)

- **Algoritmos de segmentación** de compuestos (cont.):
  - **Basados en corpus**: como los anteriores empleando un corpus de texto como lexicón
    - Reducen el problema de la falta de exhaustividad
    - Frecuencias de aparición en el corpus para calcular la mejor segmentación
    - Dependencia del corpus
  - **Estadísticos**: cálculo de la segmentación más probable en base a ocurrencias y co-ocurrencias en un corpus de entrenamiento segmentado a mano
    - Dependencia del corpus
  - **Híbridos**: combinan basados en diccionarios/corpus y estadísticos
  - **n-Gramas de caracteres**: segmentación en secuencias de n caracteres

$$\texttt{potato} \xrightarrow{n=3} \{\ \texttt{-pot-},\ \texttt{-ota-},\ \texttt{-tat-},\ \texttt{-ato-}\ \}$$

    - Simplicidad
    - Eficiencia
    - Robustez
    - Independencia del idioma

Un Ejemplo Extremo: el Chino

(Y otras lenguas asiáticas) Muy problemático:

- No existen separadores entre palabras: una oración es una secuencia continua de caracteres/símbolos
  - Las aproximaciones clásicas de IR no sirven
- Los caracteres (símbolos) chinos son mucho más significativos que los occidentales
  - La mayoría de ellos son palabras de por sí
- Vocabulario es extremadamente rico: el mismo concepto se puede expresar de múltiples formas (que suelen compartir símbolos). Ej. (Nie & Ren, 1999)

> For example, we can find the common character 建 in all the following words which mean "construction": 建设 (construct), 建筑 (construct), 建立 (construct, establish), 建成 (have constructed), 建造 (construct), 大建 (construct abundantly).

Un Ejemplo Extremo: el Chino (cont.)

- La ambigüedad en la segmentación es enorme:
  - Chino-parlantes nativos concuerdan menos del 70%
  - Ej. (Nie & Ren, 1999)

现在本所有研究生活动 (there is currently an activity for graduate students in our institute) contains the following legitimate words:

现 (now),	现 在 (now),
在 (at),	
本 (originally),	本 所 (our institute),
所 (institute),	所 有 (all, belong to),
有 (have),	
研 究 (research),	研 究 生 (graduate students),
生 (give birth),	生 活 (life),
活 (live),	活 动 (activity),
动 (move).	

There are 30 possible combinations of legitimate words which cover the sentence. Only the following one is correct:

现在 本所 有 研究生 活动
(now / our institute / have / graduate student / activity)

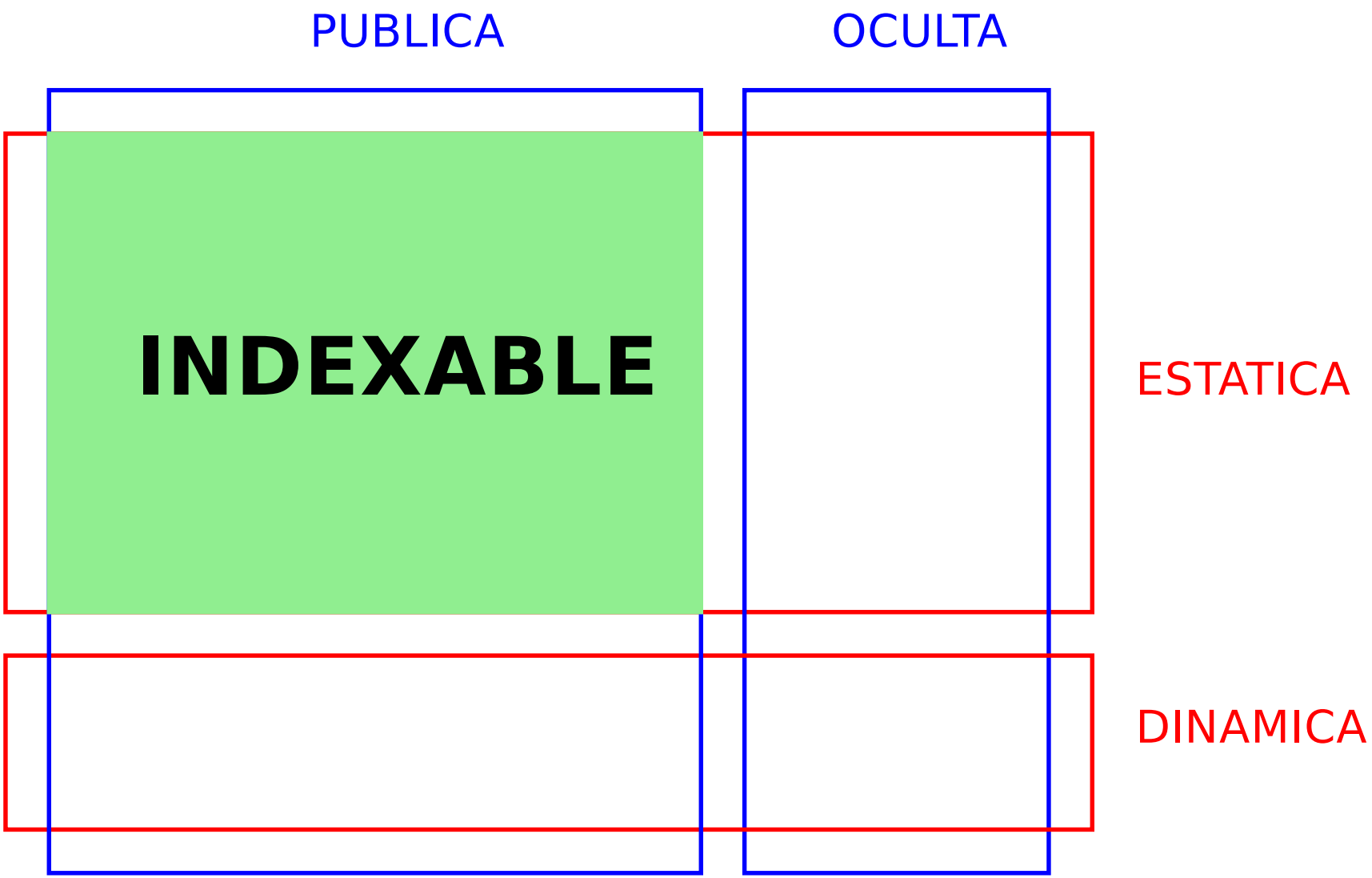
Búsqueda de Información en la Web: Introducción

- Tamaño web: miles de millones de páginas (`http://www.worldwidewebsize.com/`)
- ¿Cómo encontrar algo?
- Sitios web especializados en buscar otros sitios web (**buscadores**):
  - **Directorios**: jerarquizados por temas y categorías

    Yahoo! Directory (`http://dir.yahoo.com`)
  - **Motores de búsqueda** (o buscadores): búsqueda por palabras clave

    Google (`http://www.google.es`)
    Yahoo! (`http://www.yahoo.es`)
    Bing (`http://www.bing.es`)

Estructura de la Web



Breve historia de los buscadores

1990	Archie	Primer buscador de Internet (FTPs)
1992 Dic	Veronica	Buscador de Gopher (menús jerarquizados)
1993 Jun	Wanderer	Primer buscador web
1993 Dic	RBSE	Primero en calcular medida relevancia
1994 Ene	Galaxy	Primer directorio
1994 Abr	Yahoo	Directorio revisado manualmente
1994 Abr	WebCrawler	Salto tecnológico: indexar texto completo
1994 Jul	Lycos	Índice masivo
1995 Feb	Infoseek	Netscape. Amigable, servicios adicionales
1995 Jun	Metacrawler	Primer metabuscador
1995 Dic	Altavista	Muy veloz. Lenguaje natural y ops. lógicos

Breve historia de los buscadores (cont.)

1996 Abr	Olé	Primer buscador hispano
1996 May	HotBot	Tecnología de búsqueda de alto rendimiento
1998	MSN Search	Buscador de Microsoft
1998 Sep	Google	Nuevo salto tecnológico: algoritmo *pagerank*
1999	Baidu	Buscador chino
2005 Nov	Live Search	Plataforma *Windows Live* de Microsoft
2006	Quaero	"Google europeo"
2009 May	Wolfram Alpha	Motor de respuestas factuales
2009 Jun	Bing	Buscador actual de Microsoft

Directorios

- Sitio web que contiene un índice o lista de páginas web estructuradas jerárquicamente en base a categorías y subcategorías temáticas

  Yahoo! Directory (`http://dir.yahoo.com`)

- Estructura navegable
- Generalmente creado/revisado a mano
  - Categorización automática
- Han ido perdiendo importancia frente a los motores de búsqueda
- Actualmente son un "complemento" a éstos
- Para búsquedas muy generales

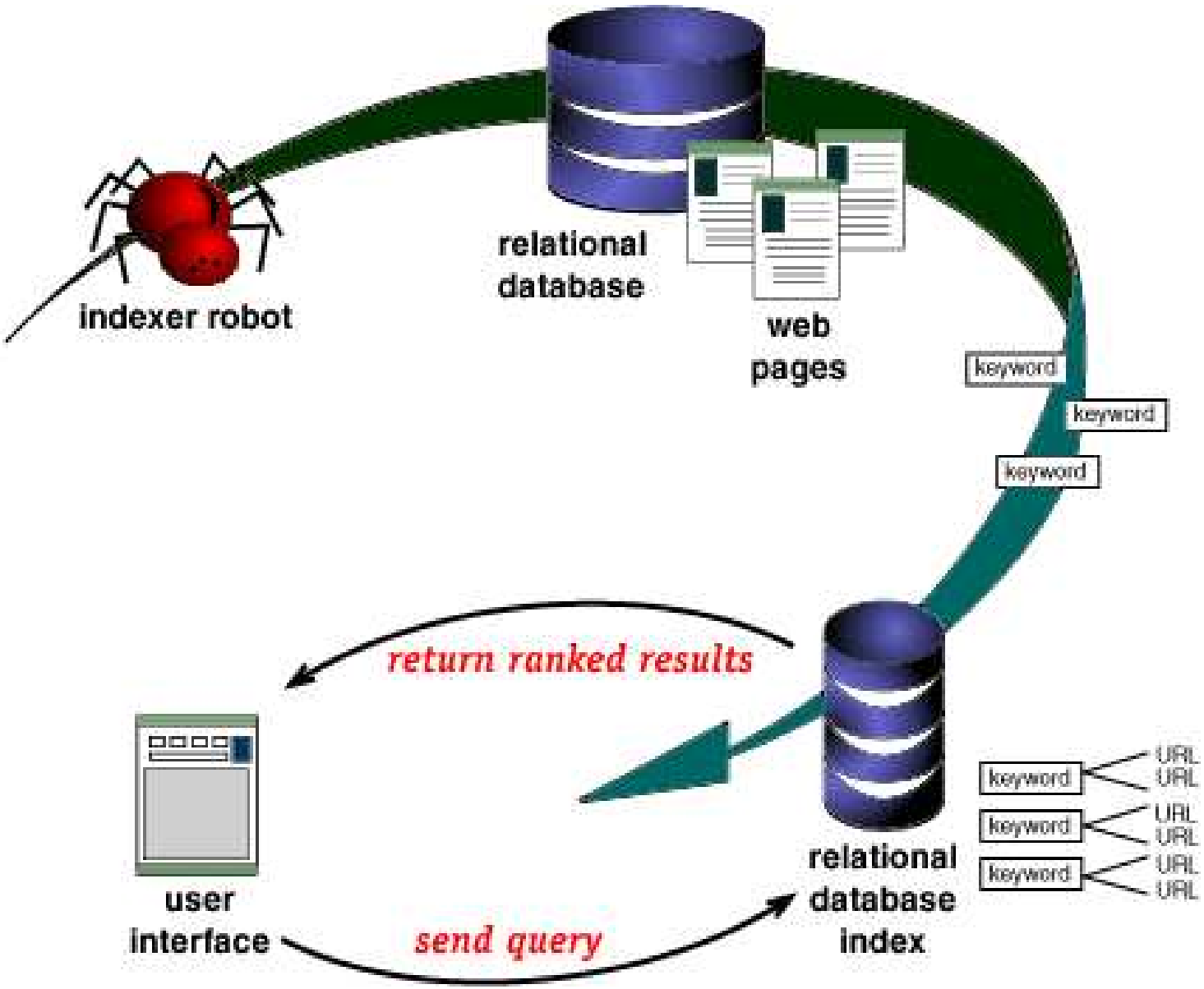
Motores de Búsqueda

- **Def.:** sitio web que contiene una base de datos (índice) donde las páginas web han sido indexadas en base a palabras clave y sobre la cual podemos realizar búsquedas (consultas o *queries*)

  Google (`http://www.google.es`)
  Yahoo! (`http://www.yahoo.es`)
  Bing (`http://www.bing.es`)

- Similar a **recuperación ad hoc**
- Para búsquedas concretas

Motores de Búsqueda (cont.): Funcionamiento



Motores de Búsqueda (cont.): Arquitectura

- **Robots:** Programas que recorren la red buscando documentos (*crawling*):
  - Analizan su contenido (total o parcial)
  - Devuelven las palabras clave o descriptores que lo describen (a indexar)
- **Base de datos:** índice de palabras clave o descriptores asociados a cada documento
  - Actualización periódica (robots)
- **Interfaz de consulta:** parte que ve el usuario
  - Introducir consulta
  - Presentar resultados

Buscadores instalables

- Permiten indexar y buscar tus propios contenidos
- *"De escritorio"*: permiten indexar y buscar en tu PC
  - Incluidos en el sistema operativo (Vista, Win7)
  - "*Spin-offs*" de los principales buscadores
    Google Desktop (`http://desktop.google.es`)
  - Conjuntamente con otro software (ej. Nero)
- Motores de búsqueda independientes
  - Algunos también contenidos web
    Ej. Lucene (`http://lucene.apache.org/`)

PageRank: Introducción

- Los modelos tradicionales de RI no tienen en cuenta la estructura de hipertexto de la red
- Un buen modelo de RI para web debe tener en cuenta:
  - La estructura de las páginas web
  - El texto de los hiperenlaces (que se asocia a la página de destino)
  - La persistencia en el tiempo de una página
  - La **popularidad** de una página web
- El **algoritmo PageRank de Google** permite calcular la popularidad de una página web en base a los enlaces que apuntan a ella
- Desarrollado en la Univ. de Stanford por Larry Page y Sergei Brin:

  S. Brin & L. Page. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *WWW7/Computer Networks* 30(1-7):107-117. `http://dbpubs.stanford.edu:8090/pub/1998-8`.

PageRank: Algoritmo

- Algoritmo de análisis de grafos que asigna a cada página web un **valor numérico (PageRank) en función de su "popularidad"**
- **Formalmente**, el *PageRank* es una distribución de probabilidades que pretende representar la probabilidad de que un usuario llegue a una página en particular recorriendo enlaces de forma aleatoria.
  - e.g., un *PageRank* de 0.5 indica que existe un 50% de probabilidades de que una persona alcance dicha página pulsando links al azar
- **Matemáticamente**, el *pagerank* de una página es un **valor del autovector principal de la matriz de adyacencia de la web**...
- **Informalmente**, **enlace de página B a página A = "voto" de página B a A**
  - Una página con muchos votos debería ser muy popular/importante
  - Un voto desde una página muy popular vale más que un voto desde una página poco popular: es un **algoritmo retroalimentado** que requiere de varias pasadas (iteraciones)

PageRank: Algoritmo (cont.)

- . . . lo que se traduce en la fórmula

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B_1)}{L(B_1)} + \frac{PR(B_2)}{L(B_2)} + \ldots + \frac{PR(B_m)}{L(B_m)}\right)$$

donde

  - A es la página cuyo *pagerank* vamos a calcular
  - B_i son las páginas que contienen enlaces a A
  - $PR(X)$ es el *pagerank* de la página web X
  - $L(X)$ es el número de enlaces diferentes de la página X
  - d es el probabilidad de que una persona siga navegando clickeando alguno de los enlaces de cualquier página (alrededor de 0,85), por lo que $1-d$ es la probabilidad de pararse en una página
  - N es el número total de páginas web

- La fórmula se recalcula iterativamente hasta converger (los valores se estabilizan)

Links sobre PageRank

- Phil Craven. Google's PageRank Explained and how to make the most of it:
  `http://www.webworkshop.net/pagerank.html`
- Ian Rogers. The Google Pagerank Algorithm and How It Works:
  `http://www.ianrogers.net/google-page-rank/`
- **Demo gráfica online:**
  `http://www.search-this.com/pagerank-decoder/`
- Calulcadora:
  `http://www.webworkshop.net/pagerank_calculator.php3`
- Demo gráfica Matlab:

`http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14258&objectType=file`

Referencias

- [CLEF] Cross-Language Evaluation Forum. Site: `http://www.clef-campaign.org`
- [TREC] Text REtrieval Conference. Site: `http://trec.nist.gov`
- [Arampatzis et al., 2000] Arampatzis, A., van der Weide, Th. P., van Bommel, P. & Koster, C.H.A. (2000). Linguistically-motivated Information Retrieval. In vol. 69 of *Encyclopedia of Library and Information Science*, pp. 201–222. Marcel Dekker, Inc.
- [Baeza-Yates & Ribeiro-Neto, 1999] Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval* Addison Wesley and ACM Press.
- [Croft et al., 2009] Croft, W.B., Metzler, D. & Strohman, T. (2009). *Search Engines: Information Retrieval in Practice*. Pearson Education.
- [Foo & Li, 2004] Foo, S. & Li, H. (2004). Chinese word segmentation and its effect on information retrieval. *Information Processing & Management*, 40(1):161-190.
- [Fuhr, 1992] Fuhr, N. (1992). Probabilistic Models in Information Retrieval. *The Computer Journal*, 35(3):243–255.
- [Hollink et al., 2004] Hollink, V., Kamps, J., Monz, C. & De Rijke. (2004). Monolingual Document Retrieval for European Languages. *Information Retrieval*, 7:33–52.

Referencias (cont.)

- [Jackson & Moulinier, 2007] Jackson, P. & Moulinier, I. (2007). Chapter 1: Natural language processing & Chapter 1: Document retrieval. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization (2nd Revised Ed.)*. John Benjamins Publishing
- [Koster, 2004] Koster, C.H.A. (2004). Head/Modifier Frames for Information Retrieval. In vol. 2945 of *Lecture Notes in Computer Science*, pp. 420–432. Springer-Verlag.
- [Kowalski, 1997] Kowalski, G. (1997). *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers.
- [Lazarinis et al., 2009] Lazarinis, F., Vilares, J., Tait, J. & Efthimiadis, E.N. (2009). Current research issues in non-English Web searching. *Information Retrieval*, 12:230-250.
- [Manning et al., 2008] Manning, C.D., Raghavan, P. & Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press.
- [Nie & Ren, 1999] Nie, J.-Y., Ren, F. (1999). Chinese information retrieval: using characters or words?. *Information Processing & Management*, 35(4):443-462.

Referencias (cont.)

- [Palmer, 2000] Palmer, D.D. (2000). *Tokenisation and Sentence Segmentation*. Chapter 2 of *Handbook of Natural Language Processing*. Dale, R. Moisl, H., Somers, H. (eds.). Marcel Dekker, Inc.
- [Vilares, 2005] Vilares, J. (2005). *Aplicaciones del Lenguaje Natural a la Recuperación de Información en Español*, PhD. Thesis, Universidade da Coruña.
- [Vilares et al., 2008] Vilares, J., Alonso, M.A. & Vilares, M. (2008). Extraction of Complex Index Terms in Non-English IR: A Shallow Parsing Based Approach. *Information Processing & Management*, 44(4), 1517–1537.