

Information Retrieval – IR

Franco Giustozzi
Bases de Datos Avanzadas

REPASO



- Hacia una definición de Recuperación de Información.
- Sistemas de Recuperación de Información.
- Modelos de Recuperación de Información.
- Evaluación de la Recuperación de Información.

AGENDA



- Lenguajes de Consulta.
- Estrategias de Búsqueda.
- Indexado y Búsqueda.



LENGUAJES DE CONSULTA

Lenguajes de Consulta



- Vamos a ver los diferentes **tipos de consultas** que se plantean normalmente a los sistemas de recuperación:
 - Consultas basadas en Keywords.
 - Single-Word Queries.
 - Context Queries.
 - Boolean Queries.
 - Natural Language.
 - Consultas basadas en reconocimiento de patrones (Pattern Matching).
 - Consultas basadas en la estructura del texto (Structural Querying).
- Esto depende en parte del modelo de recuperación que el sistema adopta.

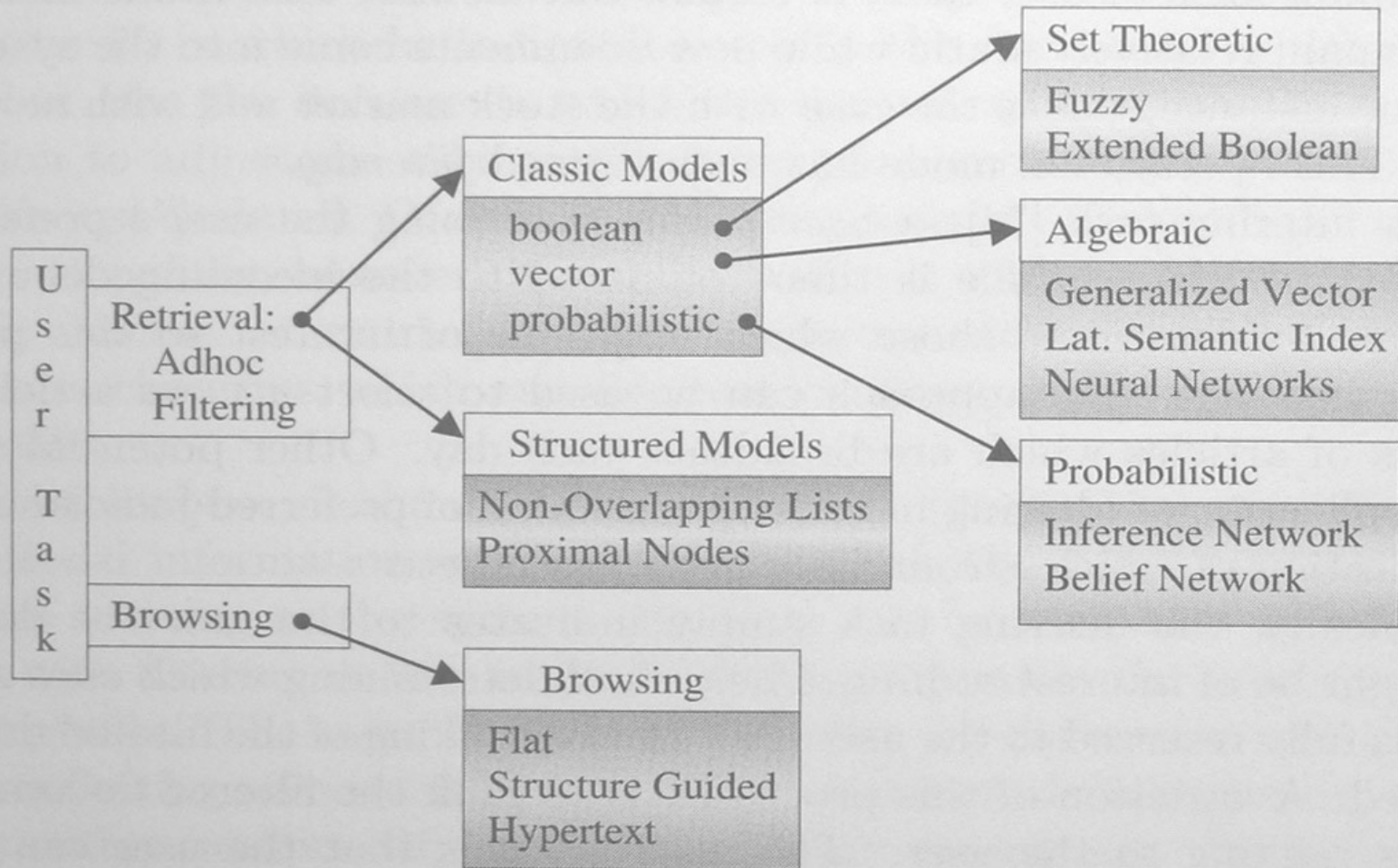
Lenguajes de Consulta



- Sabemos que hay una diferencia entre Recuperación de Información y Recuperación de Datos.
 - Los lenguajes para IR permiten **rankear** las respuestas.
- Hay varias técnicas para mejorar la utilidad de las consultas.
Algunas son:
 - La **expansión** de una palabra al conjunto de sus sinónimos o el uso de **tesauros**.
 - Eliminar palabras que son muy frecuentes y no tienen significado (llamadas **Stopwords**).
- Las palabras que pueden usarse para machear con los términos de las consultas son las **palabras claves**.

Lenguajes de Consulta

Modelos de IR



Lenguajes de Consulta

Consultas basadas en Keywords



- Una **consulta** es la formulación de una necesidad de información de un usuario.
- Las **consultas basadas en Keywords** son populares, ya que son intuitivas, fáciles de expresar y permiten rankear rápidamente.
- Pero, la formulación de una necesidad de información puede ser más compleja. Es decir, puede contener operaciones que involucren **varias palabras**.

Lenguajes de Consulta

Consultas basadas en Keywords

Word Querying

- La **consulta** más elemental que se puede formular es una **palabra**.
- Algunos modelos además son capaces de ver la división interna de las palabras en letras.
- En estos casos, una palabra es una secuencia de letras enmarcada por separadores.
- La división del texto en palabras no es arbitraria, ya que las palabras llevan significado en lenguaje natural.

Lenguajes de Consulta

Consultas basadas en Keywords


Word Querying

- El **resultado** de estas consultas es un conjunto de documentos que contienen al menos una de las palabras de la consulta.
- Los resultados están rankeados según el grado de similaridad con respecto a la consulta.
- Para hacer el ranking, se usa el método **tf-idf**.
 - Term frequency (tf): cuenta el número de veces que una palabra aparece en un documento.
 - Inverse document frequency (idf): cuenta el número de documentos en los cuales aparece la palabra.

Lenguajes de Consulta

Consultas basadas en Keywords

Word Querying



- Otra posible interpretación de las consultas es la **forma conjuntiva**, usada por los motores de búsqueda de la Web.
- En este caso, un documento machea con una consulta solo si este contiene **todas** las palabras de la consulta.
- Esto es útil, cuando el número de resultados para una sola palabra es muy grande.

Lenguajes de Consulta

Consultas basadas en Keywords

Context Querying

- Muchos sistemas complementan las consultas con la posibilidad de buscar palabras en un contexto dado.
- Palabras que aparecen cerca de otras pueden ser señal de una probabilidad más alta de relevancia que si aparecen alejadas.
- Vamos a desear formar frases o palabras que esten próximas en el texto.
 - **Phrase**
 - Secuencia de single-word queries.
 - Una occurrencia de la phrase es una secuencia de palabras.
 - Se pueden rankear de manera análoga a las single words.
 - **Proximity**
 - Es una versión más relajada de la phrase query.
 - Se da una distancia máxima permitida entre single words o phrases.

Lenguajes de Consulta

Consultas basadas en Keywords

Boolean Querying

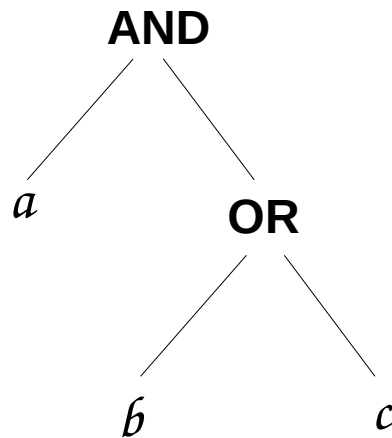
- La manera más simple y antigua de combinar palabras claves en las consultas es usando operadores booleanos.
- La sintaxis de una consulta booleana esta compuesta por:
 - **Atomos**: Consultas básicas que recuperan documentos.
 - **Operadores booleanos**: trabajan en sus operandos (conjuntos de documentos) y devuelven conjuntos de documentos.
- Este esquema es en general composicional.
 - se pueden componer los operadores sobre los resultados de otros operadores.

Lenguajes de Consulta

Consultas basadas en Keywords

Boolean Querying

- La sintaxis de una consulta define un árbol.
- Consideremos el siguiente ejemplo:



Va a recuperar todos los documentos que contengan la palabra *a* y, la palabra *b* o *c*.

Lenguajes de Consulta

Consultas basadas en Keywords

Boolean Querying

- Estos son los operadores más comunes:

Sean e_1 y e_2 dos consultas o sub-expresiones booleanas.

- e_1 **OR** e_2 : selecciona todos los documentos que satisfacen e_1 o e_2 .
- e_1 **AND** e_2 : selecciona todos los documentos que satisfacen e_1 y e_2 .
- e_1 **BUT** e_2 : selecciona todos los documentos que satisfacen e_1 pero no e_2 .
- **NOT** e_1 : selecciona todos los documentos que no satisfacen e_1 .

Lenguajes de Consulta

Consultas basadas en Keywords

Boolean Querying

- Con el sistema booleano clásico, no se pueden rankear los documentos recuperados.
 - Un documento satisface o no una consulta booleana.
- Esto es una limitación porque no permite el macheo parcial entre un documento y una consulta del usuario.
 - Para **solucionar esta limitación**, se relaja la condición.
 - Por ejemplo: Un documento que satisface parcialmente una condición **AND** será recuperado.

Lenguajes de Consulta

Consultas basadas en Keywords

Boolean Querying



- Se propuso un conjunto **fuzzy-booleano** de operadores.
- La idea es relajar el significado de los operadores **AND** y **OR**, así estos puedan recuperar más documentos.
- Los documentos son mejor rankeados cuando tienen más elementos en común con la consulta.

Lenguajes de Consulta

Consultas basadas en Keywords

Lenguaje Natural

- Empujando al modelo difuso aún mas, la distinción entre el **AND** y el **OR** puede borrarse completamente.
- En este caso, una consulta se transforma simplemente en una enumeración de consultas basadas en palabras y contextos.
- La negación (**NOT**) se puede manejar dejando que el usuario exprese algunas palabras que no desea.
 - Luego los documentos que contienen esas palabras, son penalizados a la hora de hacer el ranking.
- En este esquema, se han eliminado completamente las operaciones booleanas y se entra al campo de las **consultas en lenguaje natural**.

Lenguajes de Consulta

Pattern Matching

- Un **patrón** es un conjunto de características sintácticas que se debe encontrar en un segmento de texto.
- Un **segmento** machea con un patrón si satisface el patrón.
- Podemos buscar documentos que contengan segmentos que macheen con un patrón dado.
- Cada sistema permite especificar distintos **tipos de patrones**.
- En general, mientras más poderoso sea el conjunto de patrones permitidos, mejores consultas podrá formular el usuario.

Lenguajes de Consulta

Pattern Matching

- Los **Tipos de patrones** más usados son:
 - **Words**: una cadena que debe ser una palabra en el texto.
 - **Prefixes**: una cadena que debe formar el comienzo de una palabra del texto.
 - **Suffixes**: una cadena que debe formar la terminación de una palabra del texto.
 - **Substrings**: una cadena que debe aparecer dentro de una palabra del texto.
 - **Ranges**: una cadena que matchee cualquier cadena que este entre dos cadenas (orden lexicográfico).
 - **Allowing errors**: una palabra junto con un umbral de error.
 - **Regular expressions**: un patrón bastante general construido por cadenas simples.
 - **Extended patterns**: un lenguaje de consulta más fácil de usar para representar algunos casos comunes de expresiones regulares.

Lenguajes de Consulta

Pattern Matching

- Ejemplos:
 - ♦ el prefijo bio*.
 - ♦ el sufijo *logia.
 - ♦ Documentos que contengan valores en determinado rango:
 $1999 < x < 2014$

Lenguajes de Consulta

Structural Querying

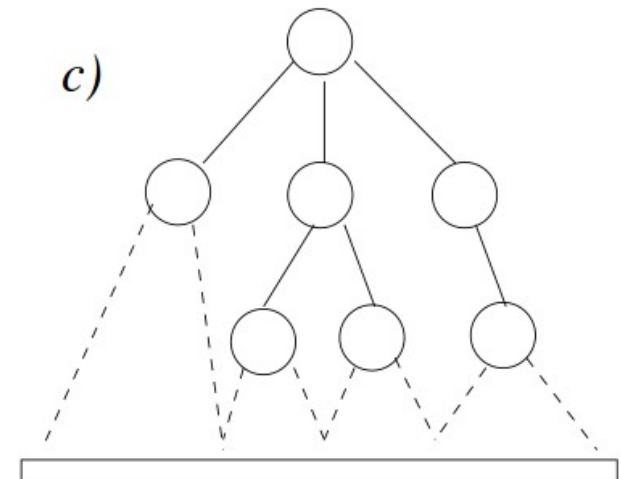
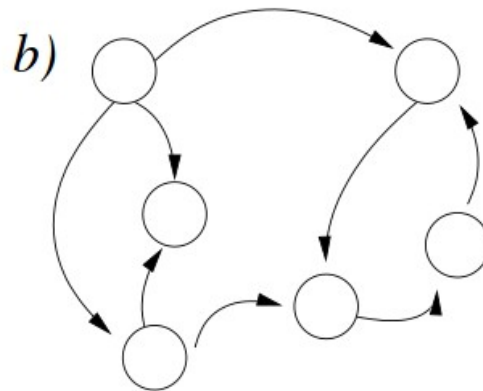
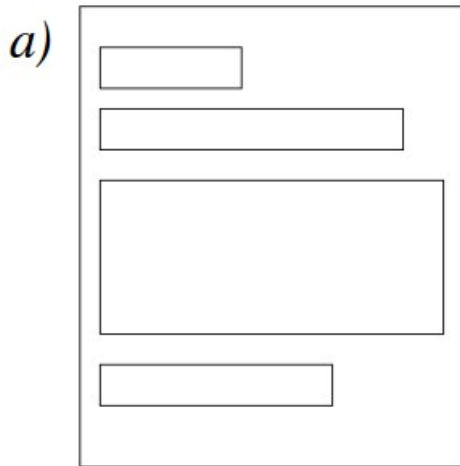
- Hasta ahora solo hemos considerado la colección de texto como un conjunto de documentos que pueden ser consultados con respecto a su contenido.
- Este modelo es incapaz de usar las nuevas ventajas que estan apareciendo, como la **estructura del texto**.
- Mezclar contenido y estructura nos permiten hacer consultas muy poderosas.
- Por lo que usando un lenguaje que soporte ambas, permite que la recuperación mejore notablemente.

Lenguajes de Consulta

Structural Querying

Los tres tipos principales de estructura son:

- a)* Fixed structure.
- b)* Hypertext structure.
- c)* Hierarchical structure.



Lenguajes de Consulta

Structural Querying

Fixed Structure



- La estructura permitida en los textos fue tradicionalmente bastante restrictiva.
- Los documentos tenían un conjunto de campos fijos, y cada campo tenía algo de texto adentro.
 - Algunos campos no estaban presentes en todos los documentos.
 - Algunos documentos podían tener texto sin clasificar en ningún campo.
 - No se permitía anidar o superponer.
- La recuperación permitida era: especificar que un patrón dado sea encontrado sólo en un campo determinado.

Lenguajes de Consulta

Structural Querying

Fixed Structure




- Cuando la estructura es muy rígida, el contenido de algunos campos pueden ser interpretados como números, fechas, etc.
- Esta idea nos lleva al **modelo relacional**, cada campo corresponde a una columna en la tabla de la base de datos.
- Hay muchas propuestas que extienden SQL para permitir recuperación full-text.

Lenguajes de Consulta

Structural Querying

Hypertext



- Los **Hypertexts** representan lo contrario con respecto al poder de la estructuración.
- La recuperación desde hypertext comenzó como una actividad meramente de **navegación**.
- Es decir, el usuario debía manualmente atravesar los nodos siguiendo los links para buscar lo que deseaba.
- Algunas herramientas permiten consultar hypertexts basandose en su contenido y estructura.

Lenguajes de Consulta

Structural Querying

Hierarchical

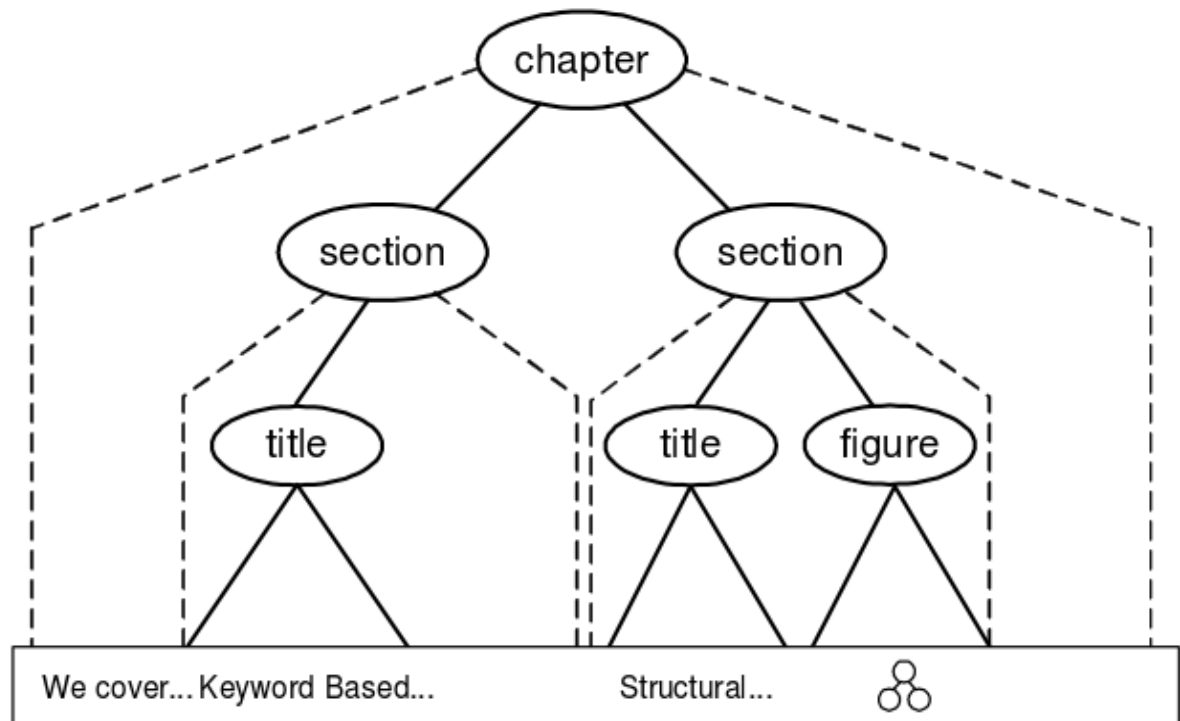
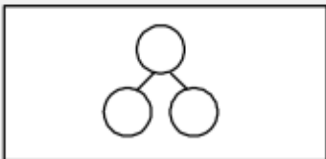
- Un modelo intermedio que está entre los dos anteriores es: **hierarchical structure**
- Un ejemplo: la página de un libro.

Chapter 6

We cover in this chapter the different kind of...

6.1 Keyword Based...

6.3 Structured Queries

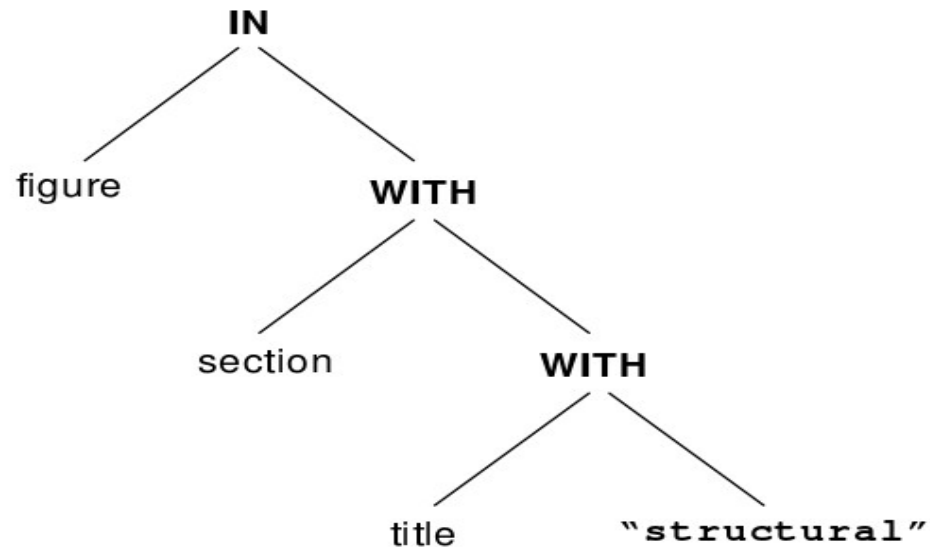


Lenguajes de Consulta

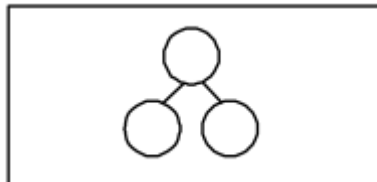
Structural Querying

Hierarchical

- Una consulta sobre la estructura podría ser:



- Va a devolver:



Lenguajes de Consulta

Structural Querying

Hierarchical

- Algunos modelos jerárquicos son:
 - PAT Expressions.
 - Overlapped Lists.
 - List of References.
 - Proximal Nodes.
 - Tree Matching.

Lenguajes de Consulta

Protocolos

- A veces, los lenguajes de consultas son usados por aplicaciones para consultar bases de datos de texto.
- Como no son para el uso humano, se los llaman **protocolos**.
- Los más importantes son:
 - Z39.50
 - Wide Area Information Service (WAIS)
- El objetivo principal de los siguientes protocolos es proveer “disc interchangeability”.
- Podemos mencionar tres:
 - Common Command Language (CCL)
 - Compact Disk Read only Data exchange (CD-RDx)
 - Structured Full-text Query Language (SFQL)
- SFQL esta basado en SQL.

Lenguajes de Consulta

Protocolos



- Por ejemplo, una consulta en SFQL es:

```
Select abstract from journal.papers  
where title contains "text search"
```

- El lenguaje soporta operadores lógicos y booleanos, tesauros, operaciones de proximidad y algunos detalles más.



ESTRATEGIAS DE BÚSQUEDA

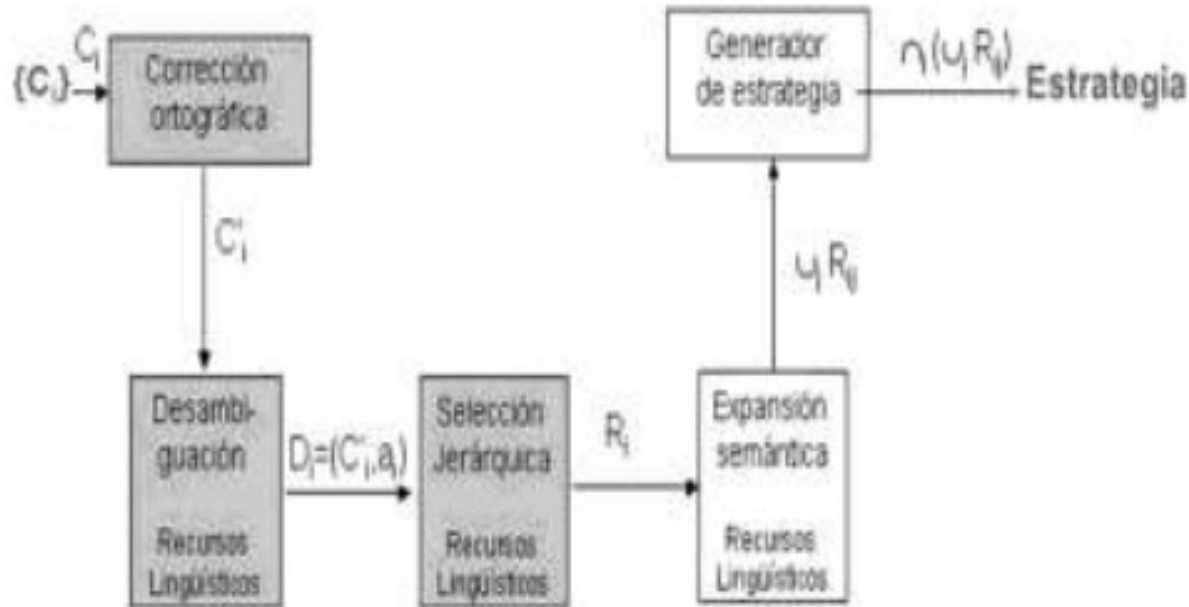
¿Qué entendemos por Estrategias de Búsqueda?



Una **estrategia de búsqueda** es una expresión lógica compuesta por distintos **conceptos** combinados con los conectores lógicos clásicos.

Expansión semántica de la consulta

- La **expansión semántica** de la consulta consiste en incorporar a la búsqueda términos que sean conceptualmente equivalentes (sinónimos, términos relacionados).
- A continuación, vemos una arquitectura de la expansión semántica de una consulta:



Expansión semántica de la consulta

Para una búsqueda que involucra los conceptos:

$$C_1, C_2, \dots, (\neg C_h), \dots, C_n$$

se obtiene la siguiente estrategia:

$$(R_{11} \text{ OR } R_{12} \text{ OR } \dots \text{ OR } R_{1r}) \text{ AND } \dots \text{ AND } \\ (\text{NOT } (R_{h1} \text{ OR } R_{h2} \text{ OR } \dots \text{ OR } R_{hr})) \dots \text{ AND } (R_{n1} \text{ OR } R_{n2} \text{ OR } \dots \text{ OR } R_{nr})$$

donde:

$(R_{11} \text{ OR } R_{12} \text{ OR } \dots \text{ OR } R_{1r})$ es la expansión del concepto C_1 .

\dots
 $(\text{NOT } (R_{h1} \text{ OR } R_{h2} \text{ OR } \dots \text{ OR } R_{hr}))$ es la negación de la expansión del concepto C_h .

\dots
 $(R_{n1} \text{ OR } R_{n2} \text{ OR } \dots \text{ OR } R_{nr})$ es la expansión del concepto C_n .

y el valor de r depende de cada concepto, pues todos los conceptos pueden tener diferentes cantidad de expansiones.

Expansión semántica de la consulta

Veamos un ejemplo:

El interés del usuario es saber la:

“relación de la aspirina en el tratamiento del cáncer de pulmón”.

Los conceptos que ingresa son: *aspirina - tratamiento - cáncer de pulmón.*

La estrategia de búsqueda provista por el generador de estrategia es:

*(lung neoplasm **OR** lung cancer **OR** cancer de pulmon **OR** carcinoma of the lungs)*
AND
*(aspirina **OR** aspirin **OR** acido acetil salicilico)*
AND
*(tratamiento **OR** treatment)*

Expansión semántica de la consulta

Recursos lingüísticos para la expansión



- Los **recursos lingüísticos** ayudan para la preparación de estrategias de búsqueda.
- Se utilizan para:
 - Desambiguar los conceptos.
 - Permitir la selección de conceptos jerárquicamente relacionados.
 - Expandir semántica y multilingualmente cada concepto.
 - Mejorar la recuperación de información.
- Los **recursos lingüísticos** pueden ser de tipo general o especializados en algún dominio del conocimiento.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

- Los **recursos lingüísticos** pueden usarse también para:
 - Clasificación de la información: Reconocer conceptos similares.
 - Por ejemplo, clasificación de las páginas Web a través de un buscador.
 - Integración de la información: Permite unificar conceptos expresados con distinta terminología y reconocer coincidencia de autores o instituciones que pueden estar expresadas de distinta manera.
 - Por ejemplo, reconocer que dos documentos provienen de la misma institución. “UNR” - “Universidad Nacional de Rosario”.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Diccionarios

- Un **diccionario** indica las distintas acepciones de un término.
 - Permite su expansión con sinónimos.
 - Algunos permiten además la expansión con merónimos, hipónimos e hiperónimos.
- Sinonimia: relación entre términos con un mismo significado. Ej: cancer, neoplasma.
- Meronimia: relación semántica entre un término que denota una parte y el que denota el correspondiente todo. Ej: brazo es merónimo de cuerpo.
- Hiponimia: relación de subordinación entre términos, es decir un término es un hipónimo de otro si su significado está incluido en el primero. Ej: gorrión es un hipónimo de pájaro.
- Hiperonimia: relación de superordenación entre términos, es decir un término es un hiperónimo de otro si su significado incluye al del segundo. Ej: animal es un hiperónimo de pájaro.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Diccionarios Multilinguales



- Para aumentar el número de documentos a recuperar se puede ampliar cada concepto en los idiomas deseados por los usuarios mediante el uso de diccionarios multilinguales generales y especializados que permiten traducir un concepto a otros idiomas.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Diccionarios - Diccionarios Multilinguales

- Algunos de los diccionarios disponibles y más usados son:
 - WordNet.
 - El diccionario de la Real Academia Española.
 - ForeingWord.com (ML).
 - Diccionarios.com (ML).
 - Wordreference.com (ML).

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Tesauros

- Un **tesauro** es un instrumento de control terminológico utilizado para traducir a un lenguaje más estricto el idioma natural utilizado en los documentos.
- Por su estructura, es un vocabulario controlado y dinámico de términos relacionados semántica y genéricamente, los cuales cubren un dominio específico del conocimiento.
- Un **tesauro** esta estructurado formalmente para hacer explicitas las relaciones entre conceptos. Estas relaciones pueden ser:
 - Jerárquicas: indican términos más amplios o más específicos de cada concepto.
 - De afinidad: términos relacionados conceptualmente, pero que no estan ni jerárquica ni preferencialmente relacionados.
 - Preferenciales: indican cual es el término preferido entre un grupo de sinónimos.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Tesauros



- En las **bases de datos documentales** se utilizan **palabras claves** para describir el contenido de un documento.
 - Estas **palabras claves**, pueden estar formadas por un término o por una frase que se ligan de un diccionario de términos controlados para el sistema, es decir un **tesauro**.
- Entonces, el **tesauro** representa una herramienta documental que permite la conversión del **lenguaje natural** de un documento al **lenguaje controlado** documental.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Tesauros

- Descriptor (o término preferente): es una palabra o grupo de palabras incluidas en un tesauro y escogidas de entre un conjunto de términos equivalentes para representar sin ambigüedad un concepto contenido en un documento o en una petición de búsqueda documental.
- No descriptor (o término no preferente o término equivalente): es un sinónimo de un descriptor presente en el tesauro. No puede ser utilizado para indizar documentos ni para formular consultas, pero reenvía al descriptor aceptado.
 - Es, por lo tanto, un punto de acceso que facilita el paso del lenguaje natural al lenguaje del sistema, permitiendo la elección de los descriptores pertinentes.

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Tesauros

- Se utilizan entre términos considerados equivalentes. Permiten evitar ambigüedades terminológicas y la sinonimia, la homonimia, la antonimia y la polisemia. Se producen entre descriptores y no descriptores.
 - Relación de Equivalencia **USE** (Use)
 - = Relación de Equivalencia **UP, UF** (Usado Por, Used For)
- Establecen relaciones de generalidad o especificidad entre descriptores. Los descriptores genéricos representan un concepto que engloba a otros más específicos.
 - < Relación Jerárquica **TG, BT** (Término Genérico, Broader Term)
 - > Relación Jerárquica **TE, NT** (Término Específico, Narrower Term)
- Relación entre dos descriptores que designan conceptos afines, pero sin una relación semántica o jerárquica.
 - Relación Asociativa **TR, RT** (Término Relacionado, Related Term)

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Tesauros

- A continuación se mencionan algunos **tesauros**:
 - Tesauros del CINDOC (Centro de Información y Documentación Científica)
http://thes.cindoc.csic.es/index_esp.php
 - Tesoro de la UNESCO
<http://databases.unesco.org/thessp/>
 - Eurovoc Thesaurus
<http://europa.eu/eurovoc/>

Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Ontologías



- Las **ontologías** permiten representar conocimiento y capturar semántica.
- Una definición conocida es: *“especificación explícita y formal sobre una conceptualización compartida”*.
- De la definición, se desprende que:
 - Las **ontologías** definen conceptos y relaciones de algún dominio, de forma compartida, y esta conceptualización se debe representar de forma formal, legible y que las computadoras la puedan usar.


Expansión semántica de la consulta

Recursos lingüísticos para la expansión

Ontologías

- Los **conceptos** son las ideas básicas que se intenta formalizar.
- **Relaciones** que representan la interacción entre los conceptos del dominio. De aquí, se deduce la taxonomía del dominio (subclase-de, is-a).
- Las **instancias** se utilizan para representar objetos determinados de un concepto.
- Los **axiomas** que son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología.
 - Estos **axiomas**, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía.

Una ontología interesante es DBpedia y se puede ver en dbpedia.org



INDEXADO Y BÚSQUEDA

Indexado y Búsqueda

- Aunque la **eficiencia** parezca un tema secundario comparado con la **efectividad**, siempre se debe tener en cuenta a la hora del diseño de un SRI.
- **Eficiencia en los SRI**: procesar consultas de usuarios con mínimos requerimientos computacionales.
- Cuando estamos en aplicaciones de gran tamaño, la **eficiencia** se vuelve más y más importante.
- Por ejemplo, los motores de búsqueda en la Web indexan teras de información y responden a miles de consultas por segundo.

Indexado y Búsqueda

- **Indice**: Una estructura de datos construida desde el texto para acelerar las búsquedas.
- En el contexto de un SRI, la **eficiencia** se puede medir por:
 - Indexing time: Tiempo necesitado para construir el índice.
 - Indexing space: Espacio usado durante la construcción del índice.
 - Index storage: Espacio requerido para almacenar el índice.
 - Query latency: Intervalo de tiempo entre la llegada de una consulta y la generación de la respuesta.
 - Query throughput: Número promedio de consultas procesadas por segundo.

Indexado y Búsqueda



- Cuando un texto se actualiza, cualquier índice construido desde ese texto debe ser actualizado.
- La tecnología actual sobre indexado no esta bien preparada para soportar cambios frecuentes sobre la colección de textos.
- Semi-static collections: Colecciones que son actualizadas cada intervalos razonables de tiempo (diariamente).
- La mayoría de las colecciones de textos, incluyendo la Web, son semi-static.

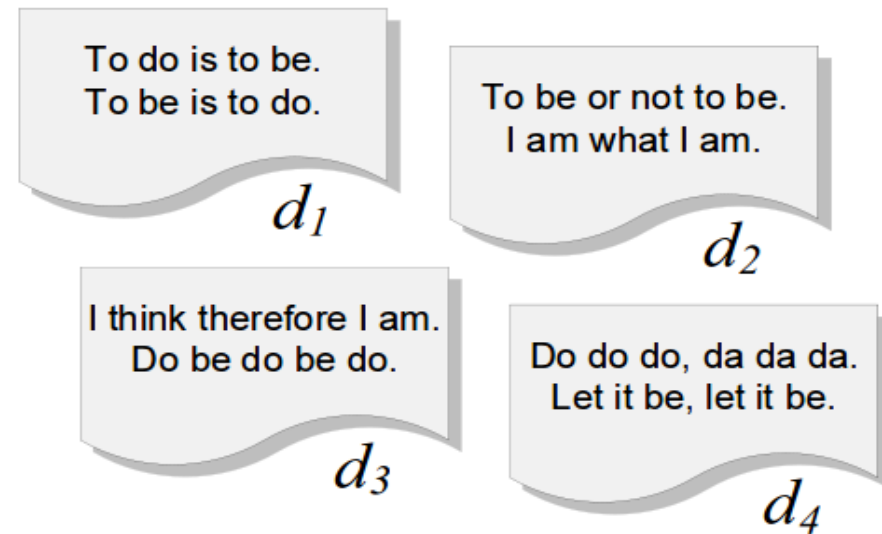
Indexado y Búsqueda

- **Indice Invertido**: un mecanismo orientado a palabras para indexar una colección de texto para acelerar las tareas de búsquedas.
- La estructura de indice invertido está compuesta por dos elementos:
 - **El vocabulario** (Conjunto de todas las palabras diferentes en el texto).
 - **Las Ocurrencias**.
- Por cada palabra en el **vocabulario**, el índice almacena los documentos que contiene esa palabra (**Indice invertido**).
- Para obtener el **vocabulario** de un conjunto de documentos, se preprocesan los documentos.
 - Tokenization.
 - Stemming.
 - Lemmatization.
 - Remoción de Stopwords.

Indexado y Búsqueda

La manera más sencilla sería, por ejemplo:

Vocabulary	n_i	d_1	d_2	d_3	d_4
to	2	4	2	-	-
do	3	2	-	3	3
is	1	2	-	-	-
be	4	2	2	2	2
or	1	-	1	-	-
not	1	-	1	-	-
I	2	-	2	2	-
am	2	-	2	1	-
what	1	-	1	-	-
think	1	-	-	1	-
therefore	1	-	-	1	-
da	1	-	-	-	3
let	1	-	-	-	2
it	1	-	-	-	2



Pero requiere mucho espacio.

Indexado y Búsqueda

La SOLUCIÓN es asociar una lista de documentos con cada palabra.

Vocabulary	n_i	Occurrences as inverted lists
to	2	[1,4],[2,2]
do	3	[1,2],[3,3],[4,3]
is	1	[1,2]
be	4	[1,2],[2,2],[3,2],[4,2]
or	1	[2,1]
not	1	[2,1]
I	2	[2,2],[3,2]
am	2	[2,2],[3,1]
what	1	[2,1]
think	1	[3,1]
therefore	1	[3,1]
da	1	[4,3]
let	1	[4,2]
it	1	[4,2]

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

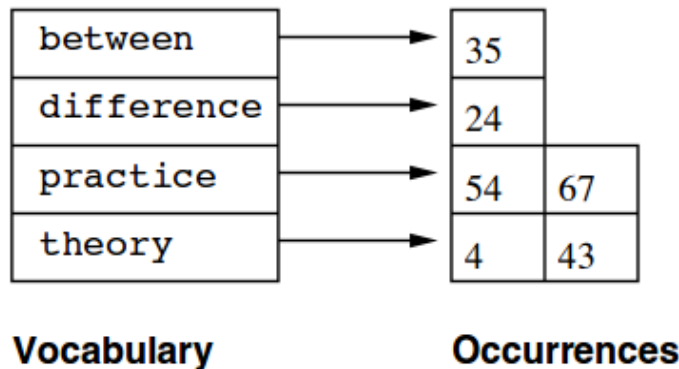
d_4

Indexado y Búsqueda

- El índice básico (anterior) no sirve para responder consultas con frases o consultas con proximidad.
- Por lo tanto, necesitamos agregar las posiciones de cada palabra en cada documento en el índice (full inverted index).

1 4 12 18 21 24 35 43 50 54 64 67 77 83
In theory, there is no difference between theory and practice. In practice, there is.

Text



Indexado y Búsqueda

- En el caso de múltiples documentos, necesitamos almacenar una lista de ocurrencia por término en cada par término-documento.

Vocabulary	n_i
to	2
do	3
is	1
be	4
or	1
not	1
I	2
am	2
what	1
think	1
therefore	1
da	1
let	1
it	1

Occurrences as full inverted lists

[1,4,[1,4,6,9]],[2,2,[1,5]]

[1,2,[2,10]],[3,3,[6,8,10]],[4,3,[1,2,3]]

[1,2,[3,8]]

[1,2,[5,7]],[2,2,[2,6]],[3,2,[7,9]],[4,2,[9,12]]

[2,1,[3]]

[2,1,[4]]

[2,2,[7,10]],[3,2,[1,4]]

[2,2,[8,11]],[3,1,[5]]

[2,1,[9]]

[3,1,[2]]

[3,1,[3]]

[4,3,[4,5,6]]

[4,2,[7,10]]

[4,2,[8,11]]

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

Indexado y Búsqueda

El **proceso de búsqueda** en los archivos invertidos esta dividido en tres etapas:

1_ Se buscan las palabras de la consulta en el vocabulario.

Si la consulta esta compuesta por varias palabras, como una frase, se la divide en los términos que la forman y se procede de esta forma separadamente para cada palabra.

2_ Se recuperan los documentos en que ocurren dichas palabras.

- Si la consulta estaba compuesta por una palabra el proceso termina y se muestran los resultados ordenados por un ranking.
- Si la consulta estaba compuesta por varias palabras, en la tercera etapa se termina de resolver la consulta procesando los datos obtenidos.

3_

- Si la consulta era una frase, se busca entre los documentos recuperados en el paso dos, aquellos en que las palabras que componen la frase aparezcan en forma consecutiva. Esto se verifica usando la información sobre las posiciones de las palabras.
- En el caso de que aparezcan operadores booleanos **AND**, **OR**, **NOT** se realizará la intersección, union ,complemento, respectivamente, de los conjuntos de documentos recuperados.

Indexado y Búsqueda



- Ventajas de los Archivos invertidos
 - Poseen un formalismo simple y eficiente.
 - Capacidad de manejar pesos no binarios y así medir la similitud entre un documento y una consulta de manera gradual.
- Desventajas de los Archivos invertidos
 - Asumen a los textos como secuencias de palabras, limitando las búsquedas.
 - Consultas complejas (como frases) muy costosas de resolver.

Indexado y Búsqueda



- Otra Técnica:
 - **Vectores de Sufijos**
 - Es posible buscar prefijos, palabras y frases directamente en los árboles de sufijos.
 - La búsqueda se realiza comparando la cadena buscada con las entradas hasta hallar una que la contenga como subcadena en su inicio.
 - Permiten resolver consultas complejas (como consultas de frases) de manera más eficiente.
 - Su construcción es un proceso muy costoso.

Bibliografía



- Bender, C. M., Deco, C., Tópicos avanzados de Bases de datos.
- Baeza-Yates, R., Ribeiro-Neto, B- (eds.), Modern Information Retrieval. New York. ACM Press, 1999.
- Salton, G. Introduction to Modern Information Retrieval. New York: McGraw-Hill, 1983.
- Van Rijsbergen, C. J. Information Retrieval. Butterworths, 1979.