

LENGUAJES NATURALES

TEMA. Extracción y Recuperación de Información

FJRP. LN, 2005

16 de enero de 2006

1. Introducción

Objetivos Generales:

- **Recuperación de Información (RI):** Determinar cuales son los documentos de una colección que satisfacen una “*necesidad de información*” de un usuario
→ RI trata de la representación, almacenamiento, organización y acceso a la información presente en colecciones de documentos textuales
- **Extracción de Información (EI):** Localizar las porciones de texto que contengan información relevante para una necesidades concretas de un usuario/s y proporcionar dicha información de forma adecuada para su proceso (de forma manual o automática)

2. Recuperación de Información

Idea Base (simplificación de partida)

- La semántica de los documentos y las necesidades de los usuarios pueden expresarse mediante un conjunto de términos de indexación (interpretación extrema del principio de composicionalidad)
- TÉRMINOS DE INDEXACIÓN: Palabra/conj. de palabras cuya semántica ayuda a identificar el tema/s principal de un documentos (Resumen el contenido del documento)
 - Generalmente suelen ser nombres (u otras palabras con contenido)
 - Pueden tener distinta importancia (asignación de pesos)
- OBJETIVO: Predecir que documentos son **relevantes** y cuales no. Es deseable además una indicación de su importancia (ranking)

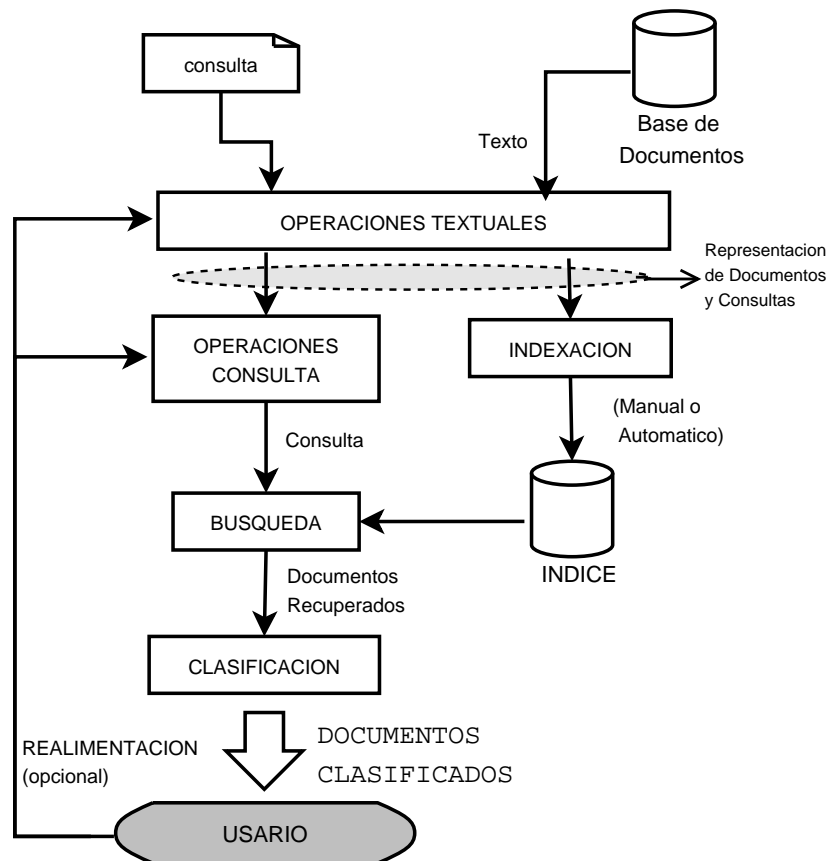
2.1. Tipos de tareas de R.I.

- AD HOC RETRIEVAL: Tarea más básica, ciclo consulta→docs. relevantes
 - El conjunto de documentos de la colección es relativamente fijo
 - Las consultas (queries) varían con cada usuario
 - Importancia del ranking de los docs. devueltos
 - Ejemplo: Buscador Internet
- CATEGORIZACIÓN DE DOCUMENTOS: Asignación de un nuevo documento a una clase de documentos preexistente

- Las clases de docs. se conocen a priori y están caracterizadas por una serie de términos (elegidos y ponderados manualmente o mediante técnicas de aprendizaje automático)
 - Las consultas (profiles) asociadas a cada tema no suelen variar
 - Importancia del método de selección de términos para caracterizar las clases
 - *Routing*: El destinatario del doc. es un usuario/conj. usuarios con interés en la categoría a la que pertenece dicho doc. (Ej.: Sistema de suscripción de noticias)
 - *Filtering*: El objetivo es aceptar o rechazar un doc. concreto (Ej.: filtro de correo)
- DOCUMENT CLUSTERING: Creación de un conjunto razonable de clases (clusters, grupos) a partir de los docs. presentes en la colección de partida.
 - Se trata de identificar los conjuntos de términos que permitan crear una clasificación aceptable (el nº de clases a crear es desconocido a priori)
 - Se busca maximizar la similaridad entre los docs. pertenecientes a un mismo cluster y minimizar la similaridad con docs. de otros clusters.
- OTRAS TAREAS:
 - *Text segmentation*: División de docs. grandes y complejos en segmentos semánticamente relevantes
 - *Text summarization*: Creación de versiones resumidas con los aspectos relevantes del doc. actual
 - *Question answering*: Búsqueda de respuestas a preguntas en lenguaje natural sobre colecciones de docs. tex-

tuales. (localización del doc. relevante + identificación de la respuesta)

2.2. Modelo general de Sistemas R.I.



Ideas básicas:

- Uso de representaciones “compatibles” para docs. y para consultas, junto con algún tipo de mecanismo de emparejamiento para relacionarlas.
- Las operaciones realizadas sobre ambos (docs. y consultas) deben ser similares.

Operaciones textuales

- Objetivo: Normalizar y conseguir mejores términos de indexación.

- Técnicas básicas:
 - *Stemming*: Creación de una forma normalizada de los términos de indexación a partir de las palabras (formas) presentes en docs. y consultas.
 - Eliminación de sufijos y prefijos
 - Uso de reglas heurísticas, no necesariamente con base léxica.
 - Aceptable en lenguajes con morfología sencilla (inglés).
 - Uso de *Stop words*: Eliminación de palabras/términos muy frecuentes y/o poco significativos desde el punto de vista semántico
 - Omitir palabras que carezcan de poder discriminatorio.

Implementación de los índices

- La creación de *índices* para los términos de indexación es útil si la colección es grande y semi-estática (no volátil). Si no es así una posibilidad es usar búsquedas directamente sobre los textos (expr. regulares)
- Técnica básica: uso de *ficheros invertidos*
- Ficheros invertidos: Relacionan cada términos de indexación con la lista de docs. en las cuales aparece (opcionalmente incorporan info. adicional [posición, importancia/frecuencia relativa dentro de cada doc., etc])
- 2 componentes:
 - *Vocabulario/diccionario*: conjunto de términos diferente presentes en la colección
 - *Ocurrencias*: lista de docs. donde aparece cada término, junto con info. adicional

- Los diccionarios suelen ser razonablemente pequeños (uso stop-words) como para poder caber en memoria principal.
- Uso de técnicas de *hashing* para agilizar las búsquedas en los diccionarios. También uso de *tries* (árboles de letras) o árboles *B*.
- Otras implementaciones/estructuras de datos (*Suffix trees*, *Suffix arrays*).
 - Ven el texto como un único string de caracteres
 - Manejan sufijos (todos los posibles) extraídos de ese string, no palabras, organizándolos en forma de árboles o arrays.

2.3. Modelos clásicos de R.I.

2.3.1. Modelo Booleano

- Modelo más simple y más usado.
- Basado en teoría de conjuntos y álgebra booleana
- Los *documentos* son conjuntos de términos (*bag of words*)
- Las *consultas* son expresiones booleanas (and, or, not + paréntesis) referidas a términos de indexación
- Asociación de conjuntos de docs. a cada término en la consulta y uso de uniones e intersecciones sobre esos conjuntos.
- PROBLEMAS
 - Rigidez: uso de criterio de decisión binario (no hay gradación)
 - Un doc. es relevante o no relevante, no hay correspondencia parcial con la consulta.

- Se recuperan demasiados docs. o muy pocos.
- No existe ranking en los docs. recuperados.
- No es trivial transformar una necesidad de info. del usuario en una consulta booleana.
 - Ej.: ‘‘veterinarios que atiendan a perros y gatos’’
 \neq veterinario AND perro AND gato

■ EXTENSIONES

- Uso de restricciones de proximidad entre términos
- Modelo booleano extendido: permite ponderación de términos de consulta y proporciona un ranking
- Modelos *fuzzy-logic*: basados en operaciones de la lógica difusa

2.3.2. Modelo Vectorial (Salton, 1971)

- Documentos y consultas se representan como vectores con las ponderaciones de los términos.
- Asignación de pesos no binarios a los términos de indexación, que representan su importancia en el vector (doc. o consulta)

$$\begin{array}{ll} \text{Documento} & \vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{nj}) \\ \text{Query} & \vec{q}_k = (w_{1k}, w_{2k}, \dots, w_{nk}) \end{array}$$

n : nº de términos distintos en toda la colección
 w_{ij} : peso del término i en el vector j (doc./query)

- NOTA: modelo booleano puede verse como un vectorial donde los pesos son 0 o 1.
- Uso de los pesos para calcular el grado de similaridad entre el vector que representa al doc. y el vector que representa la consulta.

- Se calcula una *distancia* entre los vectores de acuerdo a una métrica
- Distintas medidas de similaridad posibles (producto escalar, coeficiente Dice, coeficiente Jaccard)
 - El más usado es el *coseno* del ángulo entre los 2 vectores
 - Exige pesos con valores en $[0, 1]$, la similaridad también en $[0, 1]$

$$\text{sim}(\vec{d}_j, \vec{q}_k) = \frac{\vec{d}_j \bullet \vec{q}_k}{|\vec{d}_j| \times |\vec{q}_k|} = \frac{\sum_{i=0}^{i=n} w_{ij} \times w_{ik}}{\sqrt{\sum_{i=0}^{i=n} w_{ij}^2} \times \sqrt{\sum_{i=0}^{i=n} w_{ik}^2}}$$

- Selección de los docs. más próximos de acuerdo a la métrica
- Permite correspondencia parcial y devolver un ranking de docs. relevantes más preciso.
- **ASIGNACIÓN DE PESOS**
 - Aspecto crítico en la efectividad del modelo.
 - Uso de la frecuencia relativa de los términos (en el documento y en la consulta)

tf_{ij} : frecuencia del término i en el documento j

df_i : n° de docs. en los que aparece el término i

$idf_i = \log\left(\frac{N}{df_i}\right)$: *inverse document frequency*

con $N = n^\circ$ total de docs en la colección

PESO del término i en el documento j : $\boxed{\mathbf{w}_{ij} = \mathbf{tf}_{ij} \times \mathbf{idf}_i}$

- NOTA: Para las consultas se suelen usar fórmulas distintas ($tf - idf$ no tiene sentido).
- **Extensiones**
 - Uso de *relevance feedback*

- *Latent semantic indexing:*
 - Descomposición de vectores, reduciendo su dimensión
 - Describir regularidades en la matriz términos-documentos

2.3.3. Modelo Probabilístico

- **Idea base:** Toda consulta tendrá un conjunto ideal de docs. relevantes en la colección (conj. R)
- Para una consulta q_k se tratará de estimar la probabilidad de que un documento d_j sea relevante, $P(R|q_k, d_j)$, o irrelevante, $P(\neg R|q_k, d_j)$.
- Un doc. se devolverá se la probabilidad de ser relevante supera a la de ser irrelevante en un valor umbral.
- Se usa la regla de Bayes y una serie de simplificaciones de forma que esa condición se pueda calcular a partir de probabilidades simples calculadas en base a las probabilidades de los términos de indexación.

Por ejemplo: $P(R|t_j, q_k)$: probabilidad de “relevancia” de un término de indexación (palabra) t_i para la consulta q_k .

2.4. Evaluación de los sistemas de R.I.

- Difícil evaluar efectividad.
 - La relevancia de un doc. respecto a una consulta es un juicio humano.
 - Los criterios de efectividad pueden diferir en distintos dominios/colecciones.
- En general, basada en el uso de una colección de docs. de referencia con consultas de evaluación, acompañada cada

una de ellas por su conjunto de docs. relevantes dentro de esa colección. (Ej.: Colecciones de las competiciones TREC o CLEF)

- Medidas utilizadas

- *Precisión*: fracción de documentos devueltos que son relevante para la consulta.

$$precision = \frac{|R_a|}{|A|}$$

- *Cobertura (recall)*: fracción de los documentos relevantes que fue devuelta.

$$recall = \frac{|Ra|}{|R|}$$

- Precisión y cobertura medidas contrapuestas.

- cobertura del 100 % si devuelve todos los docs. de la colección
- precisión del 100 % si no devuelve nada (o sólo 1 doc. relevante)
- Se suelen representar en gráficos precisión s. recall (precisión con recall de 10 %, 20 %, 20 % , ...)

2.5. Técnicas de mejora de las consultas

Relevance feedback (realimentación)

- Método iterativo (e interactivo) de mejora de las consultas.
- Reformulación de la consulta original en base a los docs, que el usuario considere relevantes en sucesivos ciclos consulta→respuesta.
- Se añaden nuevos términos de consulta o se ajustan los pesos de los ya existentes en base a los términos presentes en los docs. que el usuario haya considerado relevantes en la iteración anterior.
- Múltiples técnicas: probabilísticas, basadas en aprendiz. automático (redes neuronales, algoritmos genéticos, etc,...)
 - En modelos vectoriales se usan métodos (algoritmo de Roccio) basados en el cálculo del “vector medio” de los docs, considerados relevantes, que es sumado al vector de la query original para crear el nuevo vector consulta. También se puede restar el vector medio de los docs. no relevantes.

Query expansion

- IDEA BASE: Añadir a la consulta original nuevos términos relacionados con los originales.
 - Aumenta cobertura, puede reducir precisión.
- Se basan en la definición de listas de términos relacionados (*tesauros*).
- Esas listas pueden estar basadas en:
 - relaciones semánticas: sinonimia, hiperonimia, etc,...
 - correlación estadísticas: palabras que suelen aparecer próximas, etc,...
- Generación automática de tesauros está muy relacionada con los problemas de *clustering*

3. Aplicación del Procesam. del Lenguaje Natural en R.I.

3.1. PLN en indexación

■ Indexación de palabras

- Stemming morfológico, no heurístico.
 - Uso de lemas para normalizar términos de indexación.
 - Base léxica más sólida que stemming clásico basado en reglas.
 - Poca mejora en lenguajes con poca riqueza morfológica (inglés), útil en idiomas con morfologías complejas (lenguas romances).
- Filtrado de stop-words en base a su categoría sintáctica.
- Filtrado de significados (*Word Sense Disambiguation*)

■ Indexación de conceptos/acepciones

- Incluir información sobre el sentido de la palabra al indexar.
- Explotar las relaciones semánticas entre palabras (sinonimia, hiperonimia)
- Uso de recursos como *WordNet* (normalización e indexación de symsets) o de técnicas de generación de *clusters* de palabras.
- INCONV.: Aproximación muy dependiente de la efectividad en la desambiguación de sentidos.
 - Especialmente en las consultas (poco contexto para poder identificar el sentido correcto)

■ Indexación de “frases”

- Agrupación de palabras que forman una unidad con “significado” en sí misma.
 - identificación de términos de indexación multipalabra
- No requiere un análisis sintáctico en profundidad:
 - Posibles técnicas:
 - ◇ Estadísticas: coocurrencia/cercanía de grupos de palabras.
 - ◇ Análisis sintáctico superficial: identificación de palabras con dependencias sintácticas simples.
 - Principalmente interesa identificar grupos nominales (núcleo+modificadores)
- Puede haber una fase posterior de normalización (mejora recall)

$$\left. \begin{array}{l} \text{''aroma floral''} \\ \text{''aroma de flores''} \end{array} \right\} \text{aroma+flor}$$

- PROBLEMA: Tratamiento de la ambigüedad léxica y sintáctica.

$$\text{''empanada de berberechos de Tui''} \left\{ \begin{array}{l} \text{empanada+berberecho, berberecho+Tui} \\ \text{óempanada+berberecho, empanada+Tui} \end{array} \right.$$

- **Indexación de estructuras sintácticas/semánticas de mayor nivel**

3.2. PLN en consulta

- Interfaces de consulta en lenguaje natural
- Expansión automática de las consultas.
 - Uso de técnicas de construcción automática de tesauros
 - Explotación de la semántica léxica (WorNet, expansión de sinónimos/hiperónimos)

- Relevance feedback con base lingüística
 - Uso de técnicas de sumarización y Word Sense Disambiguation.

3.3. Cross Language Information Retrieval (CLIR)

- Principal área donde PLN ha tenido éxito.
- Recuperación de documentos sin tener en cuenta el lenguaje en el que los docs. o las consultas fueron escritos.
- Se necesita un grado más alto de análisis del lenguaje que en RI convencional.
- Técnicas generales:
 - Traducir documentos
 - Traducir consultas
 - Proyectar ambos a un espacio de indexado neutral (Por ej.: usar un índice de conceptos)
- Limitaciones: los textos de las consultas son muy cortos (poco contexto) y la fiabilidad de la traducciones e menor.

3.4. Futuro del PLN y la RI

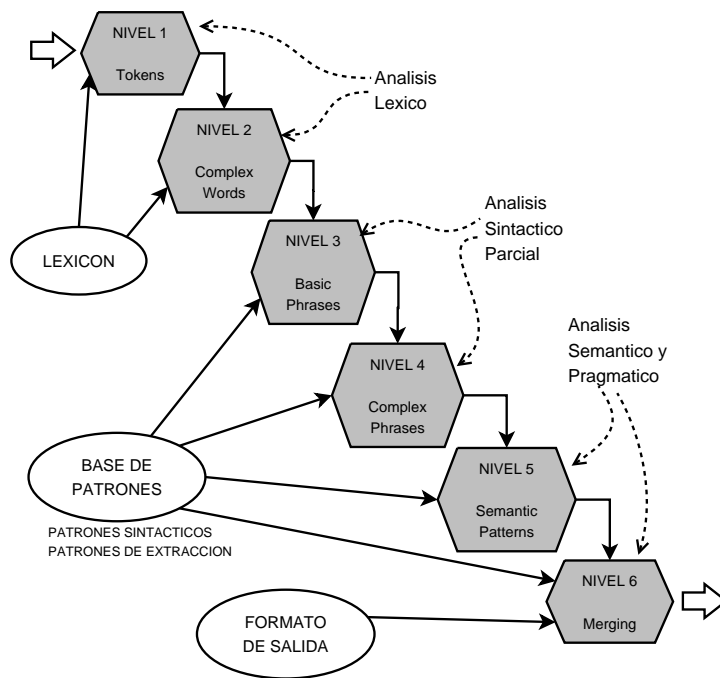
- Búsqueda conceptual (semántica).
- Respuesta a preguntas concretas (no sólo búsqueda de docs. relevantes)
- Resumen automático
- Fusión de la información recopilada por diferente métodos.
- C.L.I.R.

4. Extracción de Información

- **Objetivo:** Localizar porciones de texto con información relevante (respeto a una necesidad de info. predefinida) y presentarla de forma adecuada para su procesamiento.
- Se extrae información sobre ENTIDADES, RELACIONES y EVENTOS a partir de documentos textuales en un dominio restringido.
- Ejemplos:
 - Info. bursátil (asociaciones entre empresas [adquisiciones, ventas], tendencias, etc,...)
 - Informes meteorológicos
 - Anuncios y ofertas de empleo (para rellenar las tablas de una BD de ofertas)
- Características principales
 - La info. requerida puede representarse mediante “plantillas” fijas y relativamente simples.
Existirán “huecos” que se deberán rellenar a partir del texto localizado en los docs. (Ej: MUC6)
 - Sólo una parte pequeña del texto original es necesaria para completar la plantilla (el resto puede ignorarse)
- La salida de un sistema de E.I. será:
 - una única plantilla con un cierto n^o de elementos rellenos
 - una jerarquía compleja de plantillas y relaciones entre plantillas

4.1. Arquitectura típica de un sist. E.I.

- Arquitectura en cascada. Generalmente precedida de un paso de “recuperación de información” para identificar los documentos que se deberán procesar.
- Organizado como una secuencia de traductores que en cada pasada añaden estructura al texto y eliminan info. irrelevante, aplicando reglas de extracción adquiridas de forma manual o automática.
- Uso frecuente de técnicas de estado finito (autómatas y traductores finitos)
 - Eficiencia (complej. lineal) + simplicidad de especificación.
 - Incluyen ampliaciones (semántica, etc,...) para aproximar formalismos más potentes (gram. independ. del contexto)
- Ejemplo: Niveles de procesamiento en el sistema FASTUS (Hobbs&Applet, 1997)



Nivel 1: *Tokens*

- División del texto en palabras y etiquetado inicial

Nivel 2: *Complex Words*

- Reconocimiento de términos multi-palabra: números/expr. numéricas, fechas, nombre propios (IMPORTANTE), siglas, ...
- Uso de procesadores específicos (traductores finitos) para cada tipo.

Nivel 3: *Basic Phrases*

- Segmentación de sentencias en grupos básicos.
 - Grupos nominales y verbales
 - Partículas conectivas
 - Indicadores de localización o tiempo

Nivel 4: *Complex Phrases*

- Combinación de grupos básicos para formar grupos nominales y verbales complejos.

- Se ignoran las partes del texto no clasificadas.

Nota: En FASTUS los niveles 3 y 4 realizan un análisis sintáctico parcial.

Se usan cascadas de traductores finitos, generados a partir de metareglas.

No soporta grupos recursivos, se simulan (mediante repetición acotada) sólo en el caso que sean interesantes.

Nivel 5: *Semantic Patterns*

- Identifica entidades y eventos en los grupos creados en nivel 4.
- Inserta los objetos y relaciones reconocidos en las plantillas.
- El reconocimiento de entidades y elementos se hace mediante autómatas generados a partir de “*patrones de extracción*”.
- Los *patrones de extracción* son creados manualmente.
 - Específicos de cada dominio de aplicación. en general no transportables.
 - Similares a reglas de las gramáticas semánticas.

Nivel 6: *Merging*

- Creación de una estructura jerárquica a partir de las plantillas del nivel 5
- El algoritmo de mezcla decide si 2 plantillas parciales son consistentes y describen el mismo evento y las mezcla.
- Son necesarios procesos de resolución de referencias (análisis pragmático) para decidir si dos descripciones/referencias se refieren a la misma entidad,

- PRINCIPAL PROBLEMA: Portabilidad

- Gran dependencia del dominio.
- Cambio de dominio \Rightarrow $\left\{ \begin{array}{l} \text{cambiar lexicón (diccionario)} \\ \text{cambiar base de patrones de extracción} \\ \text{cambiar estructura de salida} \end{array} \right.$
- Necesidad de un proceso de afinado (*tunning*) manual o semi-automático.