



Modelos de Recuperación de la Información

Almacenamiento y Recuperación de la Información

Manuel Serrano

E.S. Informática CR – UCLM

Manuel.Serrano@uclm.es

<http://alarcos.inf-cr.uclm.es/doc/ari/>

1



Introducción

- 3 Modelos clásicos
 - Booleano
 - Vectorial
 - Probabilístico

- Normalmente nos basamos en términos para indexar y también para recuperar

2



Introducción

- Los términos son palabras claves que representan al documento
 - Manualmente (mejores, pero alguien tiene que elegirlos)
 - Automáticamente
- Los términos no tienen porque aparecer en el documento

3



Introducción

- Problema:
 - El enfoque es una simplificación
 - Sólo tenemos aspecto léxico
 - No tenemos
 - Sintaxis
 - Semántica
 - Pragmática
- Los 3 modelos clásicos usan esta simplificación
- Los documentos se representan por un conjunto de términos de indexación.

4



Introducción

- Modelo booleano
 - Los documentos son un conjunto de términos
 - Las preguntas son expresiones booleanas
- Modelo vectorial
- Modelo probabilístico

5



Introducción

- El modelo de representación de los documentos (D)
- Método de representación de las preguntas (P)
- Una función $S: D \times P \rightarrow \mathbb{R}$
 - Para cada par (documento, pregunta) asigna un valor real de similitud

6



Introducción

- Estas 3 características determinan el núcleo de un SARI, ignorando el modo de uso
- D y P son conjuntos de términos que analizaremos en los temas 3 y 4
- En este tema nos centraremos en la función

7



Archivos

- Tenemos un archivo con todos los documentos
- Una solución es recorrer el archivo buscando palabras → Ineficiente.
- Otra forma es tener un registro para cada documento con un 0 ó 1 por cada término

8



Archivos

	T1	T2	T3	T4	T5
D1	0	0	0	1	0
D2	0	1	0	1	0
D3	1	0	1	1	0
D4	0	1	1	0	1

Archivo Directo

- Para saber si un documento tiene un término miramos en la tabla

9



Archivos

	D1	D2	D3	D4	D5
T1	0	0	0	1	0
T2	0	1	0	1	0
T3	1	0	1	1	0
T4	0	1	1	0	1

Archivo Invertido

- Es más eficiente pues al buscar un término sólo miramos un registro

10



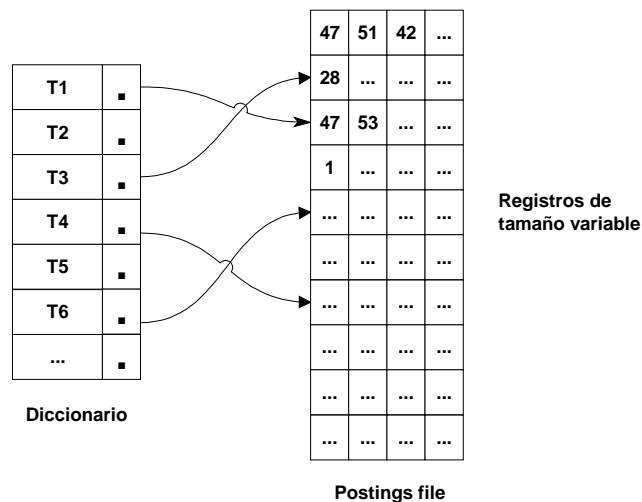
Archivos: Archivo Invertido

- El tamaño de los registros aumenta con el número de documentos
- La matriz está llena de 0's
- Solución: Partir el archivo en 2:
 - Diccionario
 - Archivo de almacenamiento de listas (Postings file)

11



Archivos: Archivo Invertido



12



Archivos: Diccionario

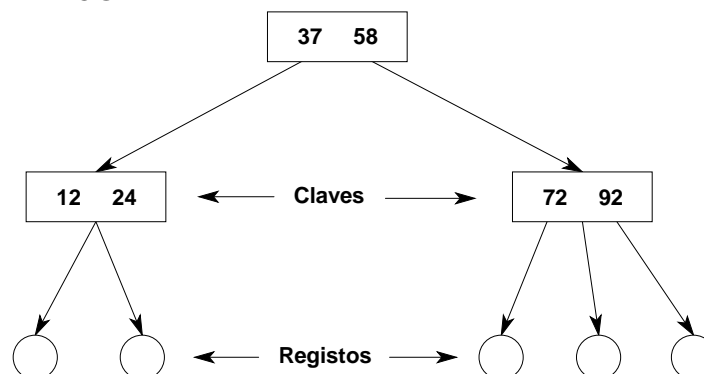
- Implementación:
 - Orden de aparición
 - A medida que vamos añadiendo documentos vamos metiendo nuevos nuevos términos
 - Ineficiente
 - Tabla ordenada
 - El problema es que tiene que ser dinámica
 - Árboles B
 - Eficientes para implementación en disco

13



Archivos: Diccionario

■ Árbol B

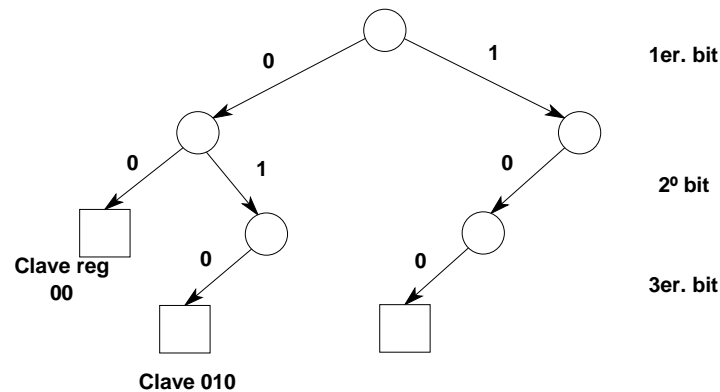


14



Archivos: Diccionario

■ Árboles digitales: TRIE



15



Archivos: Diccionario

■ Árboles digitales:

- Las claves se construyen con 0's y 1's
- Las claves están repartidas por árbol
- Es más eficiente que los árboles B
- Hay muchos tipos el de la figura anterior se llama TRIE
 - Traducción CUPE (reTRIEval → reCUPEración)
- Son muy útiles cuando se hacen búsquedas por prefijos

16



Archivos: Diccionario

- Árboles digitales:
 - Árbol Patricia
 - Optimización del TRIE
 - Los nodos que sólo tienen un descendiente se eliminan y se indica el nivel que se salta

17



Archivos: Diccionario

- Tablas de dispersión (Tablas Hash)
 - Es la más eficiente
 - mejor tiempo de acceso
 - Problemas:
 - Si es muy dinámica y está muy llena:
 - Colisiones
 - Método de resolución de colisiones
 - Es la forma más utilizada

18



Archivos: Diccionario

- Tablas de dispersión (Tablas Hash)
 - La clave se utiliza como argumento de la función que nos dice donde está
 - La función la mayoría de las veces acierta
 - El índice de aciertos decrece a medida que la tabla se va llenando

19



Archivos: Diccionario

- Tablas de dispersión (Tablas Hash)
 - Cuando hay colisiones
 - Utilizar la clave donde debería estar como clave para la nueva localización
 - Tener un área de desbordamiento
 - Las búsquedas son muy eficientes
 - Es lo que más hacemos

20



Archivos: Diccionario

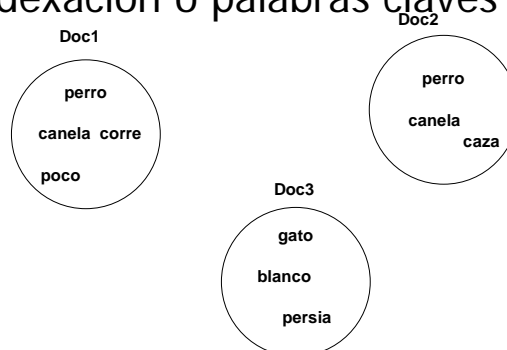
- Tablas de dispersión (Tablas Hash)
 - Las inserciones dependen de lo llena que esté.
 - Un orden de llenado del 60% - 70% está bien
 - Problemas:
 - No hay orden lexicográfico
 - Claves desordenadas
 - Nos sabemos cual es el siguiente a un término

21



Modelo Booleano

- El método de representación de los documentos es un conjunto de términos de indexación o palabras claves



22



Modelo Booleano

- Diccionario: Conjunto de todos los términos
 $T = \{t1, t2, t3, \dots\}$
- Documento: Conjunto de términos del diccionario donde tiene valor
 $D_i = \{t1, t2, t3, \dots\}$
 t_i = Verdad si es una palabra clave del doc

23



Modelo Booleano

- Las preguntas son expresiones booleanas cuyos componentes son términos de nuestro diccionario
- Operadores
 - O (\cup)
 - Y (\cap)
 - No ($-$)
 - (No suele implementarse, se suele implementar y_no)
- Ejemplo: (Perro o gato) y blanco

24



Modelo Booleano

- Función de similitud o semejanza
 - $Sem(d_i, p)$ es verdad si $p(d_i) = \text{verdad}$
 - $Sem(d_i, p)$ es falso si $p(d_i) = \text{falso}$
 - Ej:
 - $sem(d1, p) = (\text{perro o gato}) \text{ y blanco} = \text{falso}$
 - $Sem(d3, p) = (\text{perro o gato}) \text{ y blanco} = \text{verdad}$

25



Modelo Booleano

- Ventajas:
 - Más sencillo imposible
- Desventajas:
 - La función semejanza sólo tiene 2 valores
 - El lenguaje de consulta no es sencillo

26



Modelo Booleano

■ Algoritmo

- Nos permite calcular el valor de la función de semejanza
- 1ª aproximación: aplicar la función a todos los docs, pero esto no es eficiente
- Necesitamos una función que nos devuelva los id de los docs que tienen un término
 - fácil mirando el archivo invertido
- Luego mezclamos las listas

27



Modelo Booleano

■ Algoritmo:

- Entrada: 2 listas ordenadas ascendentemente
- Salida: 1 lista ordenada con la mezcla de las 2 listas de entrada
 - El orden puede ser el número de identificación de documento

28



Modelo Booleano

■ Algoritmo

```
MIENTRAS verdad
  SI ambas listas están vacías ENTONCES FIN
  SI_NO SI una lista de entrada está vacía
    ENTONCES
      transferir resto de elementos de la lista no vacía a salida
    FIN
  SI_NO tomar elemento de cabeza de L1(R1) y L2(R2)
    SI  $R1 < R2$ 
      ENTONCES transferir R1 a salida y eliminarlo de L1
    SI_NO transferir R2 a salida y eliminarlo de L2
  FIN_MIENTRAS
```

29



Modelo Booleano

■ T_i y T_j

- Mezclamos las 2 listas y es verdad para los términos que estén duplicado en la mezcla
- Ejemplo
 - $d1 = (t1, t3, t4)$ $d2 = (t1, t2, t4)$
 - $t1 = \{d1, d2\}$ $t2 = \{d2\}$ $t3 = \{d1\}$ $t4 = \{d1, d2\}$
 - $t1$ y $t4$: Mezcla = $\{d1, d1, d2, d2\} \rightarrow \{d1, d2\}$
 - $t1$ y $t3$: Mezcla = $\{d1, d1, d2\} \rightarrow \{d1\}$

30



Modelo Booleano

- T_i o T_j
 - Mezclamos las 2 listas y es verdad para los términos que estén 1 ó 2 veces
- No T_i
 - Los que no estén en la lista → Ineficiente
- T_i y no T_j
 - Hacemos T_i y T_j , Mezclamos(T_i , T_i y T_j) y quitamos los que aparecen más de una vez

31



Modelo Booleano

- Ejemplo
 - $T_1 = \{d_1, d_3\}$ $T_2 = \{d_1, d_2\}$ $T_3 = \{d_2, d_3, d_4\}$
 - $P = (T_1 \text{ o } T_2) \text{ y_no } T_3$
 - Mezcla(T_1, T_2) = $\{d_1, d_1, d_2, d_3\}$
 - $T_1 \text{ o } T_2 = \{d_1, d_2, d_3\}$ (Aparecen 1 o 2 veces)
 - Mezcla($[T_1 \text{ o } T_2], T_3$) = $\{d_1, d_2, d_2, d_3, d_3, d_4\}$
 - $(T_1 \text{ o } T_2) \text{ y } T_3 = \{d_2, d_3\}$ (Aparecen 2 veces)
 - Mezcla($[T_1 \text{ o } T_2], [(T_1 \text{ o } T_2) \text{ y } T_3]$) = $\{d_1, d_2, d_2, d_3, d_3\}$
 - $(T_1 \text{ o } T_2) \text{ y_no } T_3 = \{d_1\}$ (Quitando los duplicados)

32



Modelo Vectorial

- Es el más usado
- Permite dar graduación a la pertenencia de un documento a una pregunta
- Los docs están representados por un pto en el espacio vectorial que construimos
- La pregunta es otro punto en el mismo espacio vectorial
- Diferencia con el booleano → El método de representación es el mismo para las preguntas y los documentos

33



Modelo Vectorial

- El espacio vectorial tiene tantas dimensiones como términos de indexación tiene el diccionario
- Cada elemento del vector indica el grado de importancia de los términos en el documento (\mathbb{R}^+)
 - $d_i = (w_{i1}, w_{i2}, w_{i3}, \dots, w_{in})$
 - n : nº de términos distintos de la colección
 - $w_{ij} \in \mathbb{R}^+$

34



Modelo Vectorial

■ Ejemplo

- Diccionario (t1=perro, t2=gato, t3=azul, t4=verde, t5=pequeño)
- $d1 = (\text{perro}, \text{azul}, \text{pequeño}) = (1, 0, 1, 0, 1)$
- $d2 = (\text{gato}, \text{verde}) = (0, 1, 0, 1, 0)$
 - Asumiendo peso = 1

35



Modelo Vectorial

■ Preguntas

- Igual que los documentos
- $P = (wp_1, wp_2, wp_3, \dots, wp_n)$
 - n : nº de términos distintos de la colección
 - $wp_j \in \mathbb{R}^+$

36



Modelo Vectorial

- Función de semejanza
 - Tiene que ordenar los docs dependiendo de la proximidad con la pregunta
 - Una 1ª aproximación es el vector diferencia
 - No se usa
 - Los docs extensos tienen más términos y están más alejados
 - Las preguntas tienen pocos términos
 - Se penalizarían los docs largos

37



Modelo Vectorial

- Función de semejanza
 - Producto interno (Producto escalar)
 - Favorece los documentos largos pues al tener más términos suman más

$$sem(p, di) = \sum_{j=1}^n w_{pj} \times w_{ij}$$

38



Modelo Vectorial

- Función de semejanza
 - Función Coseno
 - Normaliza los vectores respecto a su longitud
 - Si p y di son ortogonales la relevancia es 0
 - Si son paralelos es muy relevante

$$sem(p, di) = \frac{\sum_{j=1}^n wpj \times wij}{\sqrt{\sum_{j=1}^n wpj^2} \sqrt{\sum_{j=1}^n wij^2}} = \cos(\alpha)$$

39



Modelo Vectorial

- Calculo automático de pesos
 - wij: peso del término j en el doc i
 - ftij: frecuencia del término j en el doc i
 - n: n° de términos
 - fdj: n° de docs que tienen el término j
 - d: n° de docs
 - fidj = log(d/fdj) (Frecuencia inversa)
 - Mínimo=0 fdj=d (tj aparece en todos los docs)
 - Máximo=log(d) fdj=1 (tj sólo aparece en 1 doc)

40



Modelo Vectorial

- Cálculo automático de pesos
 - $w_{ij} = f_{tij} \cdot f_{idj}$
 - Frecuencia del término en el doc X
frecuencia inversa
 - Lo importante que es el término en el doc X
lo importante que es el término en la colección
 - Para las preguntas calculamos w_{pj} igual
que w_{ij}

41



Modelo Vectorial

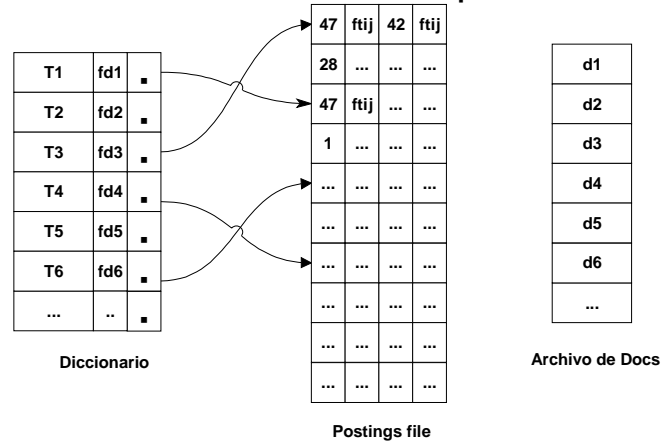
- Cálculo automático de pesos
 - Implementación
 - En los docs se almacena la frec de aparición de cada término
 - La frec inversa se deja como característica de los términos (depende de la colección)
 - Necesitamos d (nº de docs) pero es un entero que depende de la colección
 - Lo más eficaz es almacenar todo en el archivo invertido

42



Modelo Vectorial

■ Calculo automático de pesos



43



Modelo Vectorial

■ Cálculo de la semejanza

- El método típico para el cálculo de la semejanza es extraer los documentos que tienen alguno de los términos de la pregunta
- Así solo calculamos la semejanza para unos pocos documentos

44



Modelo Probabilístico

- La base de cálculo es la probabilidad de un documento de ser relevante a una pregunta dada
- La función de semejanza es la probabilidad de que un doc sea relevante

$$\text{Sem}(p, d_i) = P(R|d_i)$$

45



Modelo Probabilístico

- Utilizaremos el modelo probabilístico de independencia de términos binarios
 - La probabilidad de los términos es independiente (un término es independiente de los otros)
 - Los pesos asignados a los términos son binarios

46



Modelo Probabilístico

- Representación igual al modelo booleano
 - $di = \{wi1, wi2, \dots, win\}$
 - $wij = \{1 \text{ si } tj \text{ es término de } di, 0 \text{ si no}\}$
- Las preguntas son un subconjunto de términos

47



Modelo Probabilístico

- Para la función de semejanza es mejor coger como relevante aquellos docs en los que su probabilidad de ser relevante es mayor que la de no serlo

$$P(R | di) > P(\bar{R} | di)$$

$$sem(p, di) = \frac{P(R | di)}{P(\bar{R} | di)} \rightarrow \begin{cases} > 1 & \text{relevante} \\ < 1 & \text{no relevante} \end{cases}$$

48



Modelo Probabilístico

- Aplicando el Teorema de Bayes

$$sem(p, di) \approx \frac{P(di | R)}{P(di | \overline{R})}$$

- $P(di|R)$ es la probabilidad de que dado el conjunto de relevantes di esté dentro

49



Modelo Probabilístico

- Simplificando...

$$sem(p, di) = \sum_{j=1}^m w_{pj} w_{ij} \log \frac{p_j(1-q_j)}{q_j(1-p_j)}$$

- p_j es la probabilidad de que t_j esté en el conjunto de docs relevantes
- q_j es la probabilidad de que t_j esté en el conjunto de docs no relevantes

50



Modelo Probabilístico

- Si conociéramos R:

		Relevancia		
		SI	NO	
Término j	SI	r	n-r	n
	NO	R-r	N-R-n+r	N-n
		R	N-R	N

- N = N° docs colección
- R = N° docs relevantes
- r = N° docs relevantes que tienen t_j
- n = N° docs colección que tiene t_j

51



Modelo Probabilístico

- Si conociéramos R:
 - p_j : probabilidad de que cogiendo un doc relevante tenga t_j
$$p_j = (r/R)$$
 - q_j : probabilidad de que en los docs no relevantes no esté t_j
$$q_j = (n-r)/(N-R)$$

52



Modelo Probabilístico

- Si conociéramos R:
 - Substituimos p_j y q_j

$$sem(p, di) \approx \sum \log \frac{\frac{r}{R-r}}{\frac{n-r}{N-R-n+r}}$$

53



Modelo Probabilístico

- Como no conocemos R
 - La prob de que un término esté en el conj de docs relevantes es la misma para todos los términos. A priori no hay un término más relevante que otro
 $p_j = 0.5$
 - Para q_j se usa la frec inversa del término en la colección

$$q_j = fdj/N$$

fdj : N° de docs con el término t_j
 N : N° de docs de la colección

54



Modelo Probabilístico

- Como no conocemos R:
 - Usamos $p_j = 0.5$ y $q_j = f_{dj}/N$
 - Recuperamos los N primeros y consideramos que son relevantes
 - Preguntamos al usuario y recalculamos p_j y q_j

55



Modelos de navegación

- La necesidad de información no se expresa por una pregunta
- En un sistema real tenemos una mezcla:
 - Hago una pregunta
 - Exploro los resultados
 - Reformulo la pregunta...
- El interfaz de usuario es más importante

56



Modelos de navegación

- Navegación directa
 - Los docs están normalmente almacenados en una lista sin un criterio de ordenación útil o agrupados por su proximidad semántica
 - La carga de exploración corre por parte del usuario
 - Para agruparlos se suele utilizar una función de semejanza entre los docs
 - El criterio de semejanza es muy difuso → que estén próximos

57



Modelos de navegación

- Navegación guiada por una estructura
 - Los docs están organizados por **UN** criterio
 - Este criterio genera una estructura jerárquica
 - Los temas de la raíz son genéricos y las hojas específicos
 - Se usa en bibliotecas y centros de documentación

58



Modelos de navegación

- Navegación guiada por hipertexto
 - Los docs están organizados por un sistema multicriterio
 - Obtenemos un grafo
 - Los docs son nodos cualesquiera del grafo
 - Es el sistema de moda

59



Modelos Avanzados

- Son extensiones del modelo booleano
- El modelo booleano no permite dar orden al conjunto de docs relevantes
- El modelo booleano sólo permite decir si son relevantes o no
- En las extensiones se permite dar un orden

60



Conjuntos Difusos

- Se basa en que la representación de un doc por un conjunto de término no sea categórica
- Se difumina la pertenencia para que no sea binaria
- $F(D,t)=0 \text{ ó } 1 \Rightarrow f(D,t) \in [0,1]$
(Función de pertenencia)

61



Conjuntos Difusos

Fórmula lógica	Fórmula de evaluación
$f(di,tj \text{ y } tk)$	$F(di,tj) * F(di,tk)$ $\equiv \min(F(di,tj), F(di,tk))$
$f(di,tj \text{ o } tk)$	$F(di,tj) + F(di,tk) - (F(di,tj) * F(di,tk))$ $\equiv \max(F(di,tj), F(di,tk))$
$f(di,tj \text{ y_no } tk)$	$F(di,tj) * (1 - F(di,tk))$

62



Conjuntos Difusos

- Cálculo de pesos
 - Generalmente se hace como en el modelo vectorial
 - $w_{ij} = f_{dij} * f_{idj}$
- Cuando las funciones sólo toman 0 ó 1 degenera en el modelo booleano

63



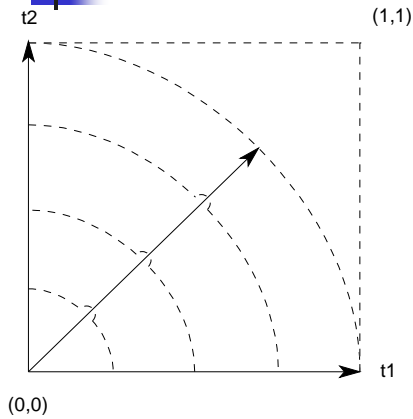
Conjuntos Difusos

- Desventajas:
 - Es menos flexible de lo que parece
 - No podemos dar pesos a las preguntas
 - La ordenación puede estar desfigurada pues sólo algunos términos influyen en la semejanza
 - Ej: $d1 = \{(t3, 0.8)\}$ $d2 = \{(t1, 0.7), (t2, 0.7), (t3, 0.8)\}$
 - $P = t1$ o $t2$ o $t3$
 - A los dos les da 0.8 (Máximo) aun cuando el $d2$ sería más relevante

64



Booleano Extendido



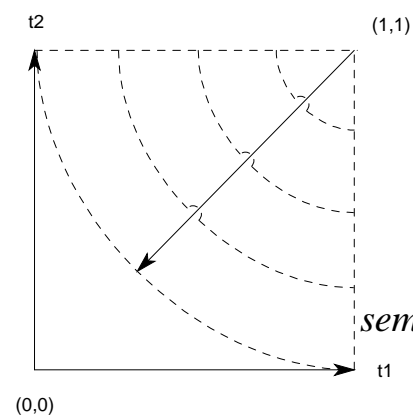
- Introduce el concepto de distancia
- $P = (t_1 \text{ o } t_2)$
- Cuanto más cerca del 0 menos representa al doc

$$sem(d, t_1 \circ t_2) = \sqrt{\frac{t_1^2 + t_2^2}{2}}$$

65



Booleano Extendido



- $P = (t_1 \text{ y } t_2)$
- Cuanto más cerca del 0 más representa al doc

$$sem(d, t_1 y t_2) = 1 - \sqrt{\frac{(1-t_1)^2 + (1-t_2)^2}{2}}$$

66



Booleano Extendido

- Aquí todos los términos contribuyen
- Introducimos los pesos

$$P=(w_{pj},w_{pk}) \quad di=(w_{ij},w_{ik})$$

$$sem(d, t_j \circ t_k) = \sqrt{\frac{w_{pj}^2 w_{ij}^2 + w_{pk}^2 w_{ik}^2}{w_{pj}^2 + w_{pk}^2}}$$

$$sem(d, t_j y t_k) = 1 - \sqrt{\frac{w_{pj}^2 (1 - w_{ij})^2 + w_{pk}^2 (1 - w_{ik})^2}{w_{pj}^2 + w_{pk}^2}}$$

67



Booleano Extendido

- Podemos dar peso a los operandos (p)

$$sem(d, t_j \circ^p t_k) = \sqrt[p]{\frac{w_{pj}^p w_{ij}^p + w_{pk}^p w_{ik}^p}{w_{pj}^p + w_{pk}^p}}$$

$$sem(d, t_j y^p t_k) = 1 - \sqrt[p]{\frac{w_{pj}^p (1 - w_{ij})^p + w_{pk}^p (1 - w_{ik})^p}{w_{pj}^p + w_{pk}^p}}$$

$$sem(di, t_j y_{no}^p t_k) = w_{pj} \cdot w_{ij} \cdot w_{pk} \cdot (1 - w_{ik})$$

68



Booleano Extendido

- Si $p=\infty$

$$sem(d, t_j \circ^\infty t_k) = \lim_{p \rightarrow \infty} \left(\sqrt[p]{\frac{w_{pj}^p w_{ij}^p + w_{pk}^p w_{ik}^p}{w_{pj}^p + w_{pk}^p}} \right)$$

$$sem(d, t_j \circ^\infty t_k) = \frac{\max(w_{pj} w_{ij}, w_{pk} w_{ik})}{\max(w_{pj}, w_{pk})}$$

$$sem(d, t_j \circ^\infty t_k) = 1 - \frac{\max(w_{pj}(1 - w_{ij}) + w_{pk}(1 - w_{ik}))}{\max(w_{pj} + w_{pk})}$$

69



Booleano Extendido

- Si $p=\infty$ y $w_{pj}=w_{pk}=1$
 - $Sem(d_i, t_j \text{ o } t_k) = \max(w_{ij}, w_{ik})$
 - $Sem(d_i, t_j \text{ y } t_k) = 1 - \min(w_{ij}, w_{ik})$
- Es lo mismo que el modelo de conjuntos difusos
- El modelo booleano extendido engloba al modelo de conjuntos difusos

70



Booleano Extendido

- Si $p=1$

$$sem(d, t_j \circ^1 t_k) = \frac{w_{pj} w_{ij} + w_{pk} w_{ik}}{w_{pj} + w_{pk}}$$

$$sem(d, t_j y^1 t_k) = 1 - \frac{w_{pj}(1 - w_{ij}) + w_{pk}(1 - w_{ik})}{w_{pj} + w_{pk}}$$

$$sem(d, t_j y^1 t_k) = \frac{w_{pj} w_{ij} + w_{pk} w_{ik}}{w_{pj} + w_{pk}}$$

71



Booleano Extendido

- Si $p=1$

- No hay diferencia entre "y" y "o"
- La ordenación es igual a la del modelo vectorial
- El modelo booleano extendido engloba también al modelo vectorial

- Si $1 < p < \infty$

- Situaciones intermedias
- Con $p \rightarrow \infty$ nos acercamos al booleano estándar
- Con $p \rightarrow 1$ nos acercamos al modelo vectorial

72



Booleano Extendido

$$P = ((A, a)y^2(B, b))o^\infty(C, c)$$

- Pregunta: Quiero que el "y" se asemeje al modelo vectorial y el "o" al modelo booleano

73



Booleano Extendido

- Ventajas
 - Modelo generalista
 - Engloba a muchos otros
- Desventajas
 - Las leyes booleanas (asociativa, distributiva,...) no se cumplen
 - Ej: $\text{sem}((A \text{ y } B) \text{ y } C) \neq \text{sem}(A \text{ y } (B \text{ y } C))$
 - Coste computacional es muy alto (Normalmente solo se permite $p=1, 2 \text{ e } \infty$)
 - Los usuarios no se sienten a gusto con la formulación (Por defecto $p=2$)

74