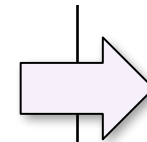


# Modelos de Búsqueda




# Respondiendo consultas

Valor de recuperación, "score"

$$f(q, d_1) \in \mathbb{R}$$



Ranking

$f(q, d_3)$    $d_3$   
∨  
 $f(q, d_1)$    $d_1$   
∨  
 $f(q, d_2)$    $d_2$   
∨  
⋮  
⋮




$f(d, q)$  estima en qué medida  $d$  es una buena respuesta a  $q$

$f$

**Función de ránking**

$q = w_1, w_2, w_3, \dots$

**Consulta**

$d_1$   $d_2$   $d_3$  ...  
   ...

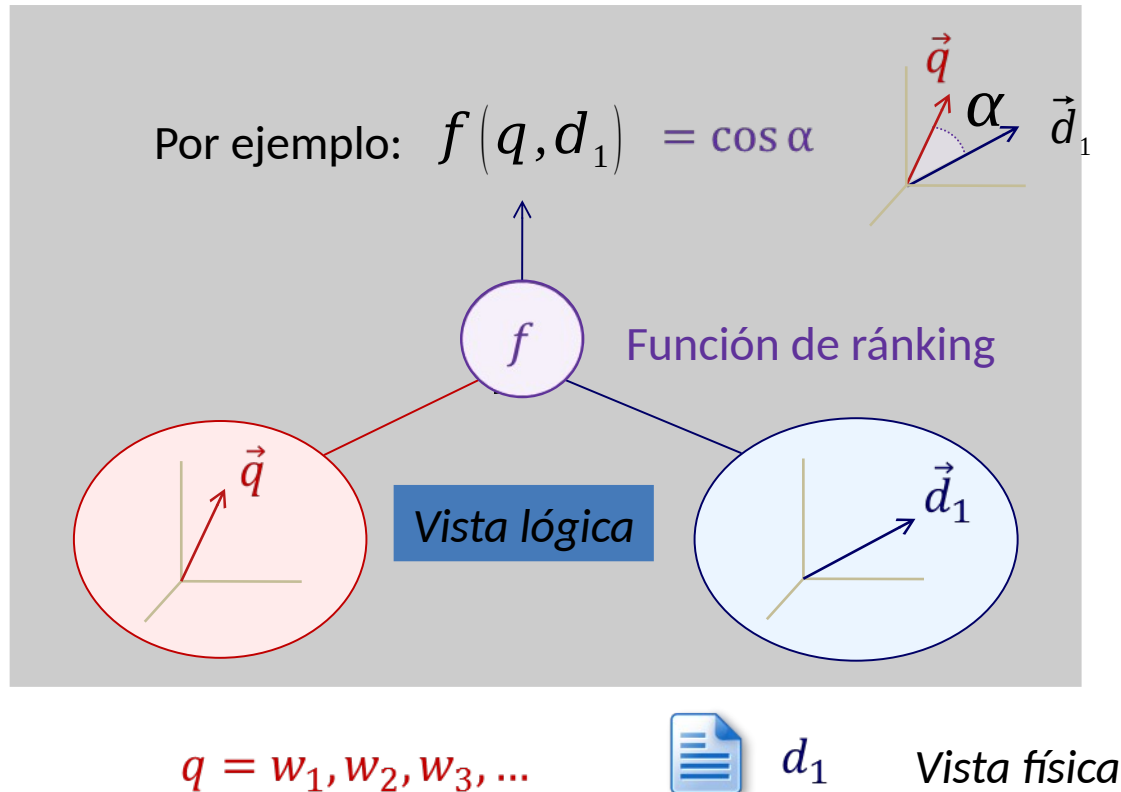
**Espacio de búsqueda**

Necesidad de información



Usuario  
final

# Modelo de RI



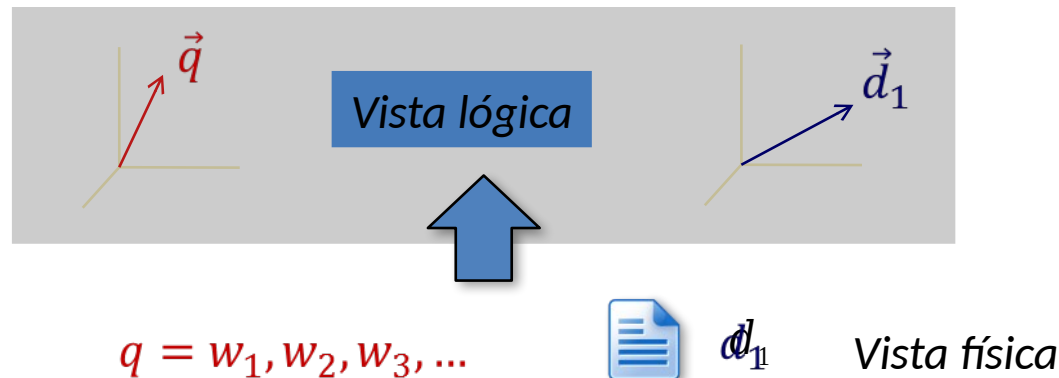
Un **modelo RI** es:

- ♦ Una **representación lógica** de documentos y consultas  
Incluyendo los métodos y cálculos para construirla
- ♦ Una **función de ránking** sobre dicha representación  
Puede ser muy elaborada de calcular

Induce  
el algoritmo  
de ránking



# Modelo RI



Construir la representación lógica de un espacio de gran escala es altamente costoso

- ♦ Se necesita **construir y actualizar offline**
- ♦ Y almacenar en una estructura eficiente que permita acceso rápido y concurrente en tiempo de consulta **○ Índices de búsqueda**

# Modelos de IR para texto

- ◆ Basados en *bag of words*
- ◆ Cada uno se caracteriza además por...
- ◆ Un framework: principios y fundamentos que rigen las relaciones entre palabras, documentos y consultas  
Una **función de recuperación** que define el ránking  
Típicamente cada modelo incluye una forma de definir
- ◆ un **peso** para cada palabra en cada documento  
Se puede ver como una matriz término / documento

# Modelo de Recuperación = definición medible de “relevancia”

$S(\text{"world cup schedule"}, d)$

$s(\text{"world"}, d)$

$s(\text{"cup"}, d)$

$s(\text{"schedule"}, d)$

Cuántas veces aparece “schedule” en  $d$ ?

**Term Frequency (TF):**  $c(\text{"schedule"}, d)$

Qué tamaño tiene  $d$ ?

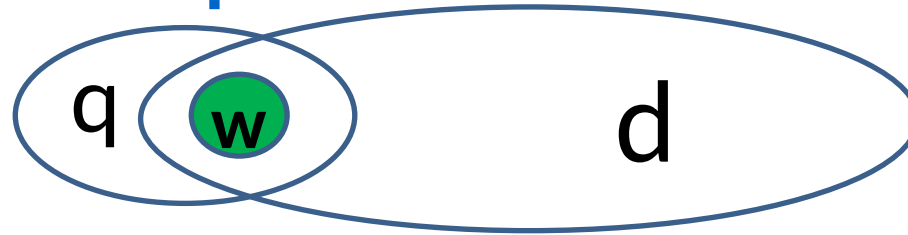
**Document length:**  $|d|$

Es frecuente “schedule” en nuestra colección  $C$ ?

**Document Frequency:**  $df(\text{"schedule"})$

$P(\text{"schedule"} | C)$

# En general, los modelos consideran una bolsa de palabras



Suma sobre  
términos comunes

$$s(q, d) = f \left( \sum_{w \in q \cap d} \text{weight}(w, q, d), a(q, d) \right)$$

$$g[c(w, q), c(w, d), |d|, \text{df}(w)]$$

Term Frequency (TF)

Document length

$$p(w | C)$$

Inverse  
Document  
Frequency  
(IDF)

# Modelos de IR

*¿Cómo se determina qué documentos son probablemente relevantes para una consulta?*



# Modelos de IR en texto

**Boolean:** [Lancaster et al. 1973]



**Vector Space Models:** [Salton et al. 1975], [Singhal et al. 1996],



...

**Classic Probabilistic Models:** [Maron & Kuhn 1960], [Harter 1975], [Robertson & Sparck Jones 1976], [van Rijsbergen 1977], [Robertson 1977], [Robertson et al. 1981], [BM25](#) [Robertson & Walker 1994], ...



**Language Models:** [Ponte & Croft 1998], [Hiemstra & Kraaij 1998], [Zhai & Lafferty 2001], [Lavrenko & Croft 2001], [Kurland & Lee 2004], ...

**Non-Classic Logic Models:** [van Rijsbergen 1986], [Wong & Yao 1995], ...

**Divergence from Randomness:** [Amati & van Rijsbergen 2002], [He & Ounis 2005], ...

**Learning to Rank:** [Fuhr 1989], [Gey 1994],

# Modelo booleano

- ♦ Los pesos de los términos son binarios: 1 si aparecen y 0 en otro caso
  - Se ignora la frecuencia de aparición
  - Los documentos se representan como **conjuntos** de términos
  - Las respuestas son **exactas** (tal como se ha definido la tarea)
- ♦ Se puede utilizar and, or y not en las consultas
- ♦ Se devuelven los documentos que cumplen la condición expresada en la consulta
  - Poner la consulta en forma normal disyuntiva
  - Formar la unión de los documentos que cumplen cada componente conjuntiva

# Modelo booleano: función de ránking

Dada  $q = q_1 \vee q_2 \vee \cdots \vee q_n$

$$f(d, q) = \begin{cases} 1 & \text{si } \exists i : d = q_i \\ 0 & \text{en otro caso} \end{cases}$$

# Ejemplo

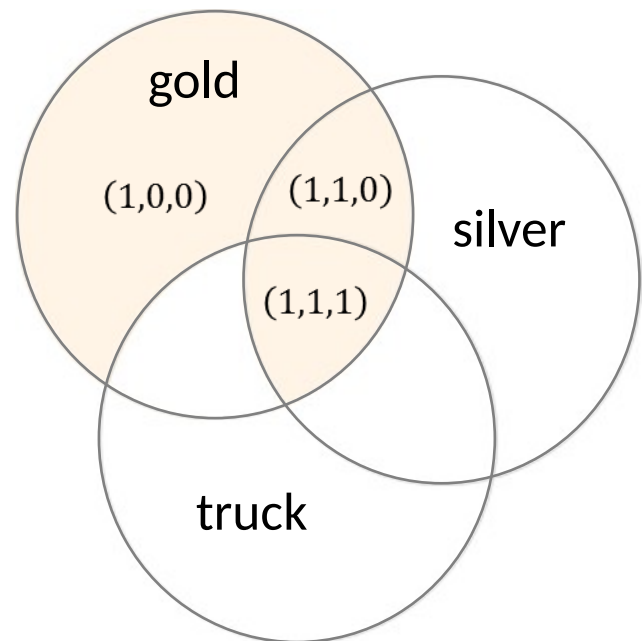
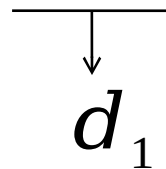
$$q = \text{gold} \wedge (\text{silver} \vee \neg \text{truck})$$

$$d_1 = \text{"Shipment of gold damaged in a fire"} \rightarrow (1,0,0)$$

$$d_2 = \text{"Delivery of silver arrived in a silver truck"} \rightarrow (0,1,1)$$

$$d_3 = \text{"Shipment of gold arrived in a truck"} \rightarrow (1,0,1)$$

$$q = (1,1,0) \vee (1,1,1) \vee (1,0,0)$$



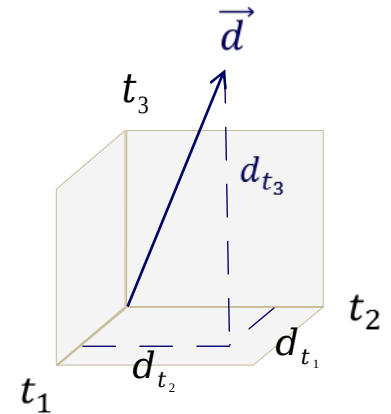
# Modelo booleano

En sí mismo, un modelo muy limitado

- ♦ **No hay ránking** , no escala bien con el número de docs  
Devuelve demasiados documentos o demasiado pocos
- ♦ Las consultas booleanas resultan difíciles a los usuarios
- ♦ Sin embargo fue el modelo primario durante tres décadas
- ♦ Se sigue usando en funcionalidades de búsqueda sencilla
- ♦ (email, escritorio, bibliotecas...) y como primer filtro
- ♦ de un segundo algoritmo
- ♦ Mejores soluciones: tener en cuenta la **frecuencia** de los términos

# Modelo vectorial (VSM)

- ♦ Se representan documentos y consultas en un espacio vectorial  $R^V$ , donde  $V$  es el vocabulario (número de términos)
- ♦ La coordenada de los vectores documento para cada término son pesos que se calculan con una fórmula basada en frecuencias
  - ¿Cómo definir una ponderación representativa?
  - Que por un lado cuantifique cuán representativo es cada término en el documento
  - Que por otro matice entre términos muy comunes y otros más específicos (y por tanto significativos)

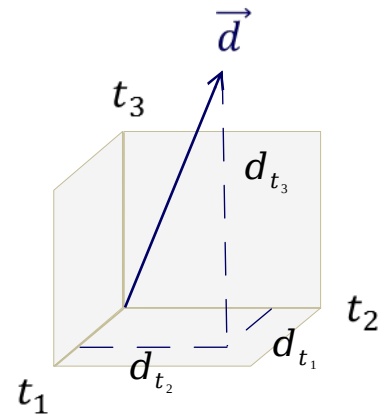


# Modelo vectorial: esquema *tf-idf*

- ♦ El esquema típico de ponderación es *tf-idf*

$$d_t = tf(t, d) \cdot idf(t)$$

- *tf* mide la “importancia” de los términos en los documentos
- *idf* mide el poder de discriminación del término
- ♦ Existen diversas variantes para concretar las funciones *tf* e *idf*, en todas ellas:
  - *tf*(*t*, *d*) es creciente respecto a la frecuencia de *t* en *d*
  - *idf*(*t*) mide la especificidad de *t* por su frecuencia en la colección



# El esquema *tf-idf*

◆

$$tf(t, d) = \begin{cases} 1 + \log_2 \text{frec}(t, d) & \text{si } \text{frec}(t, d) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$idf(t) = \log \frac{|\mathcal{D}|}{|\mathcal{D}_t|}$$

-----

la colección de documentos (espacio de búsqueda)  
documentos que contienen el término

- ◆ *tf* tiene que ver con la probabilidad del término en el documento
- ◆ E *idf* con la probabilidad en la colección



# El esquema *tf-idf* (cont)

◆ Otras variantes:

$$tf(t, d) = \frac{frec(t, d)}{\max_{t' \in \mathcal{V}} frec(t', d)}$$

- ◆ Pro: evita ventaja para documentos largos
- ◆ Contra: sensible a outliers

$$tf(t, d) = \lambda + (1 - \lambda) \frac{frec(t, d)}{\max_{t' \in \mathcal{V}} frec(t', d)} \quad \text{p.e. } \lambda = 0.5$$

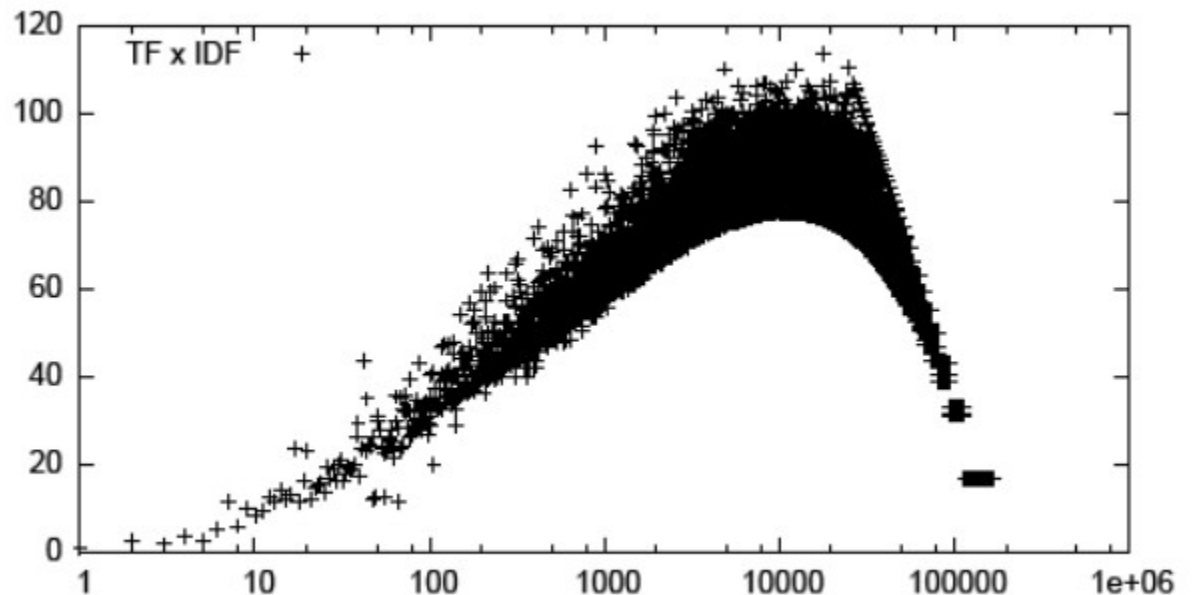
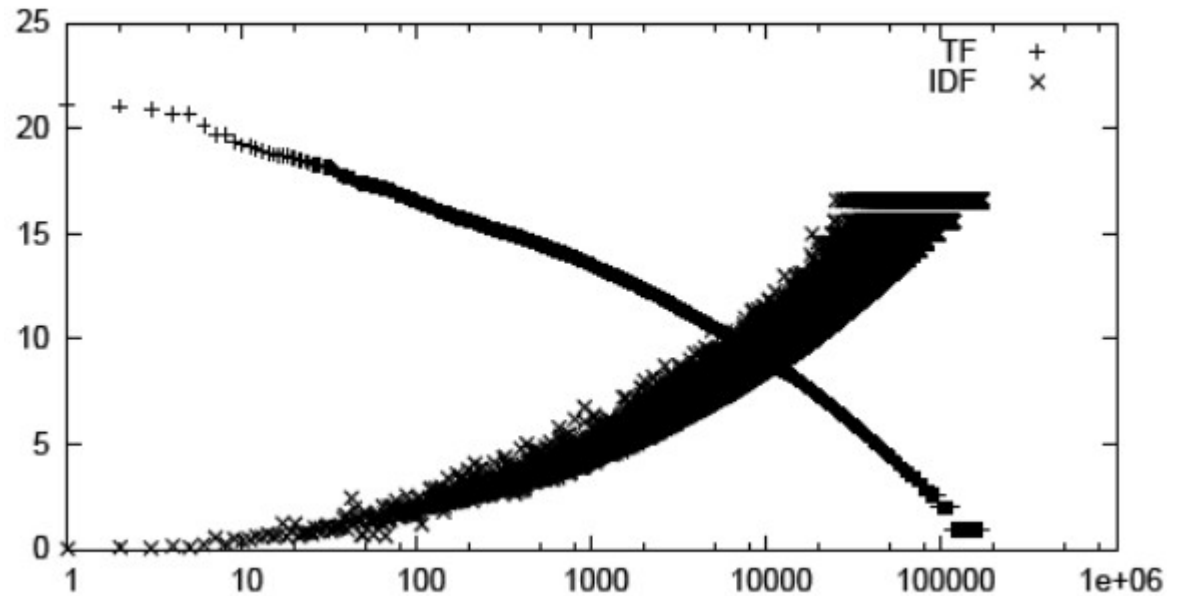
$$idf(t) = \log \left( 1 + \frac{|\mathcal{D}|}{1 + |\mathcal{D}_t|} \right)$$

...y unas cuantas más (tuning)

# TF vs. IDF

Plots colección Wall Street  
Journal  
Comportamiento

- ♦ power law
- ♦ Tf e idf se contrarrestan
- ♦ Idf intermedios son los
- ♦ más interesantes



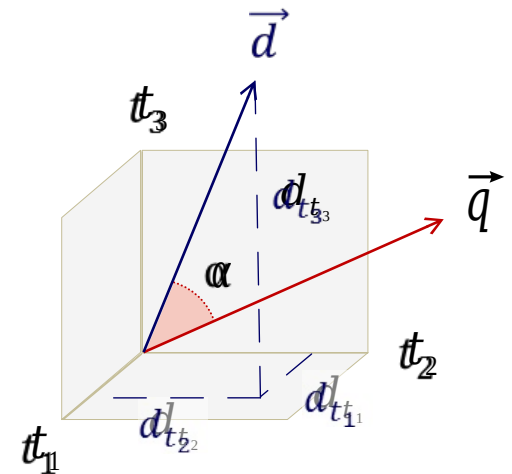
# Modelo vectorial: función de ránking

Finalmente...

- ◆ Construimos  $\vec{q}$ 
  - También por *tf-idf*, aunque no necesariamente con la misma variante
- ◆ Comparamos los vectores  $\vec{d}$  y  $\vec{q}$  en similitud por ángulo

$$f(d, q) = \text{sim}(d, q) = \text{angulo}(\vec{d}, \vec{q}) \propto \cos(\vec{d}, \vec{q})$$

$$\cos(\vec{d}, \vec{q}) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}| |\vec{q}|} = \frac{\sum_t d_t q_t}{\sqrt{\sum_t d_t^2} \sqrt{\sum_t q_t^2}} \in [0, 1]$$



# Ejemplo 2

$$1 + \log_2 \text{frec}(t, d)$$

$$\log \frac{|\mathcal{D}|}{|\mathcal{D}_t|}$$

	frec(t, d)			
	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>
arbol	4			
hoja		4	2	
olivo			1	1
raiz			4	1
rama	1	4	2	1
savia	4		1	

arbol	4			
hoja		4	2	
olivo			1	1
raiz			4	1
rama	1	4	2	1
savia	4		1	

	tf(t, d)			
	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>
arbol	3	0	0	0
hoja	0	3	2	0
olivo	0	0	1	1
raiz	0	0	3	1
rama	0	0	0	0
savia	3	0	1	0

	idf(t)			
	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>
arbol	2			
hoja	1			
olivo	1			
raiz	1			
rama	0			
savia	1			

6	0	0	0	1
0	3	2	0	1
0	0	1	1	1
0	0	3	1	
0	0	0	0	
3	0	1	0	

$q$  = "hoja arbol olivo"

$$\frac{d \cdot q}{|d| |q|}$$

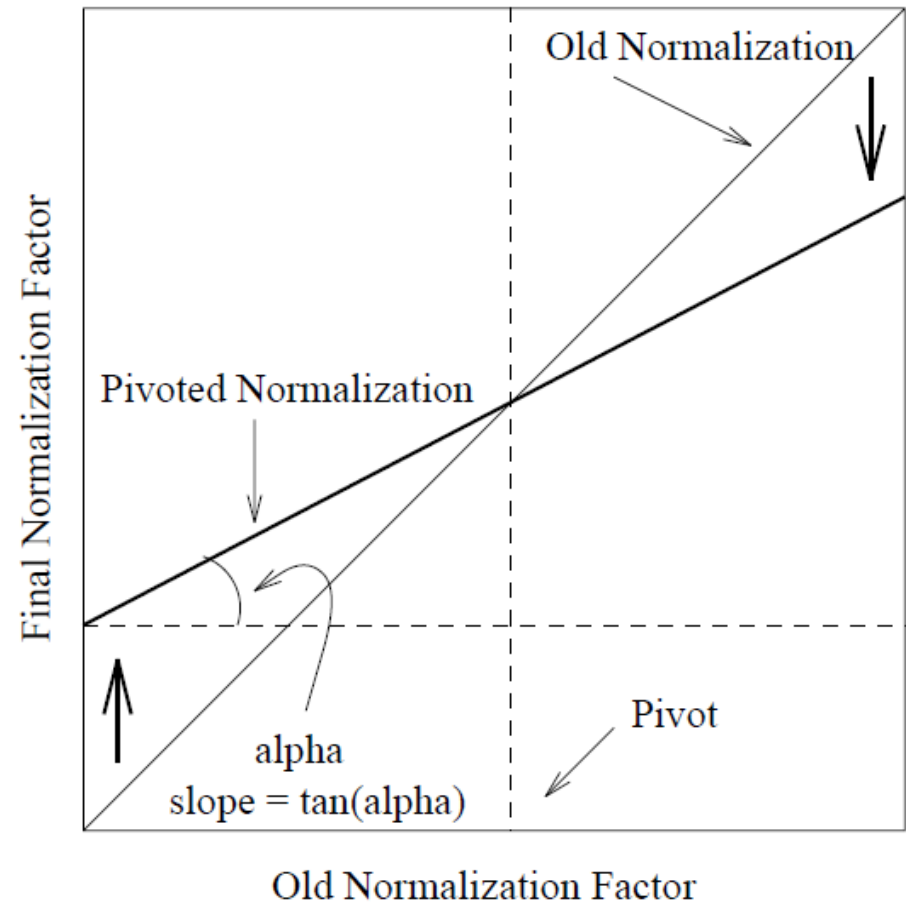
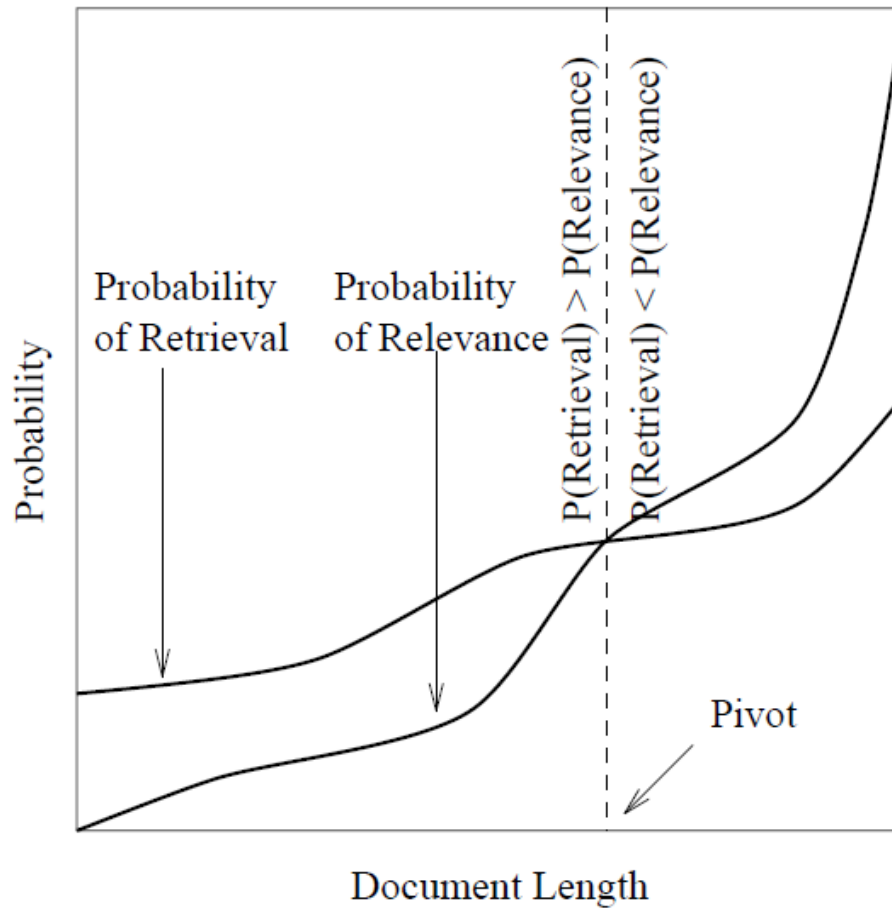
3				
0.52	0.58	0.45	0.41	

# Normalización por pivote

- ❖ Propuesta por Amit Singhal (hoy senior VP Google) y otros en 1996
- ◆ Hoy día incorporada de forma habitual a las implementaciones del modelo vectorial
- ◆ La normalización del coseno (norma  $L_2$ ) es demasiado severa en general
- ◆ Singhal et al muestran que los documentos largos tienden a ser más relevantes y proponen una normalización más suave

$$norm(d) = f(|d|) \leq |d|$$

# Normalización por pivote



# Normalización por pivote

- ♦ Rotación de pendiente  $m$  con centro en el punto  $p$  (pivote)

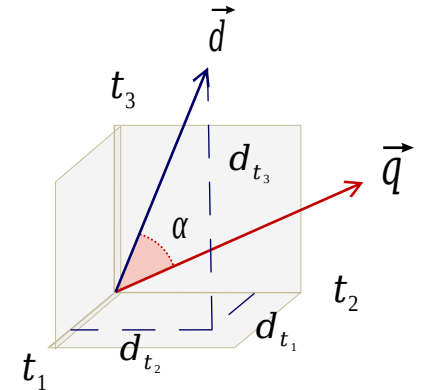
$$\text{norm}(d) = f(|d|) = \frac{m}{(1-m)p} |d| + 1, \quad m \in [0,1]$$

- ♦ Se suele tomar  $p \sim \text{avg}_d |d|$
- ♦ El parámetro  $m$  se optimiza con datos de entrenamiento
- ♦ Más detalles en: A. Singhal, C. Buckley, M. Mitra. Pivoted Document Length Normalization. SIGIR 1996, pp. 21-29

# Modelo vectorial por coseno

## Vector de consulta

- ♦ Se podría hacer tf binario
  - Salvo que se considere significativa la repetición de términos
  - P.e. aplicaciones donde la consulta es un documento
- ♦ idf penaliza doblemente los términos muy comunes (se puede omitir)
- ♦ Se puede omitir la norma de la consulta en el denominador, no cambia el ranking



## Normalización longitud de documento

El módulo del documento en el denominador representa

- ♦ una normalización para evitar el sesgo a documentos largos
- ♦ Se puede normalizar por otras funciones de longitud

Tamaño en bytes

Nº de palabras

Pivoted normalization

...

¿Sería también adecuada la distancia euclídea como alternativa al coseno?



# Modelo vectorial

- ♦ Primeras versiones de los años 50
  - Sigue siendo muy utilizado hoy día
- ♦ También se utiliza en clustering, clasificación, y otras aplicaciones con documentos de texto
  - También en espacios donde las coordenadas son otro tipo de características (tags, etc.)
- ♦ Aproximación geométrica a la estimación de relevancia
- ♦ Similitud gradual a la consulta, mejora la calidad del ránking
- ♦ Un documento puede ser recuperado aunque no contenga todos los términos de la consulta
- ♦ Incorpora normalización por longitud de forma natural

## Modelos estado del arte ...

- Modelo de Espacio vectorial

Pivoted length normalization (PIV) [Singhal et al. 1996]

$$\sum_{w \in q \cap d} \frac{1 + \ln(1 + \ln(c(w, d)))}{(1 - s) + s \frac{|d|}{avdl}} \cdot c(w, q) \cdot \ln \frac{N + 1}{df(w)}$$

- Modelo Probabilístico

BM25 [Robertson & Walker 1994]

$$\sum_{w \in q \cap d} \ln \frac{N - df(w) + 0.5}{df(w) + 0.5} \cdot \frac{(k_1 + 1) \times c(w, d)}{k_1((1 - b) + b \frac{|d|}{avdl}) + c(w, d)} \cdot \frac{(k_3 + 1) \times c(w, q)}{k_3 + c(w, q)}$$

## Modelos estado del arte ...

- Basados en Modelado de Lenguaje
  - ↪ Query likelihood with Dirichlet prior (DIR) [Ponte & Croft 1998], [Zhai & Lafferty 2001]

$$\sum_{w \in q \cap d} c(w, q) \times \ln \left( 1 + \frac{c(w, d)}{\mu \cdot p(w|C)} \right) + |q| \cdot \ln \frac{\mu}{\mu + |d|}$$

- Modelo Probabilístico
  - ↪ Divergence from randomness (PL2) [Amati & van Rijsbergen 2002]

$$\sum_{w \in q \cap d} c(w, q) \cdot \frac{tfn_w^d \cdot \log_2(tfn_w^d \cdot \lambda_w) + \log_2 e \cdot \left( \frac{1}{\lambda_w} - tfn_w^d \right) + 0.5 \cdot \log_2(2\pi \cdot tfn_w^d)}{tfn_w^d + 1}$$

$$tfn_w^d = c(w, d) \cdot \log_2 \left( 1 + c \cdot \frac{avdl}{|d|} \right), \lambda_w = \frac{N}{c(w, C)}$$

# PIV, DIR, BM25 y PL2 tienen eficacia similar.

## Rendimiento (MAP)

	AP88-89	DOE	FR88-89	Wt2g	Trec7	trec8
PIV	0.23	0.18	0.19	0.29	0.18	0.24
DIR	0.22	0.18	0.21	0.30	0.19	0.26
BM25	0.23	0.19	0.23	0.31	0.19	0.25
PL2	0.22	0.19	0.22	0.31	0.18	0.26