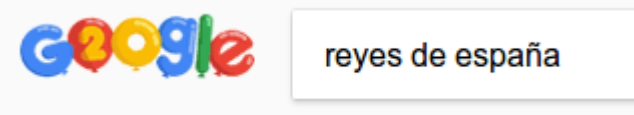
Modelos de Recuperación de Información

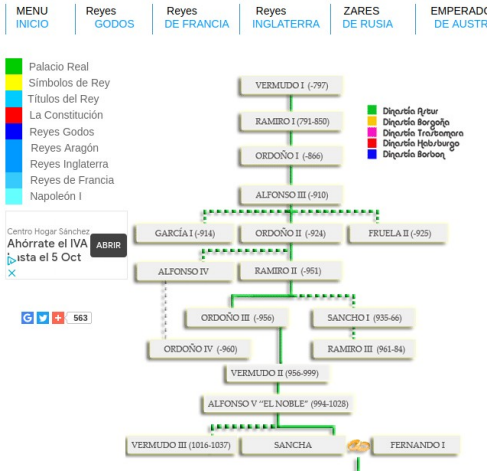
Esquema

- Introducción a los Modelos de Recuperación de Información
- Modelos Clásicos
  - Booleano
  - Vectorial
  - Probabilísticos

[algunas transparencias provienen de “Introduction to Information Retrieval”]

Un **Modelo de RI** es la especificación sobre cómo representar documentos y consultas, y cómo comparar unos y otras





Introducción

- Relevancia es el concepto básico en RI, pero no hay una clara definición.
- Se pueden utilizar distintos criterios para medir la relevancia
  - Documento y consulta hablan de la misma temática
    - Q: “Reyes de España”
    - D: artículo en wikipedia sobre los Reyes Católicos
  - Se pueden incorporar conceptos externos al contenido del doc.
    - Antiguedad, Estilo, Contenido novedoso (aporta nueva información)
- Cada criterio nos lleva a un modelo de recuperación de información distinto

Introducción

- ¿Cómo se mida la relevancia?

  Se asume que si un documento tiene una mayor similitud que otro a una consulta dada, entonces el primero será mas relevante que el segundo.

- La función de comparación ($\rightarrow$ ranking) se define como una medida de similitud entre documento y consulta,

$$f(q, d) \rightarrow R$$

  Es necesario hacer algunas suposiciones para simplificar los cálculos.

Simplificación I:

Bolsa de Palabras (Bag of words)

- El cálculo de la similitud no tiene en cuenta el orden de los términos:

  es una forma efectiva de abordar la problemática de la R.I.

  - Comparan palabras independientemente del orden en que aparecen en el texto:

- Por ejemplo, considerar los siguientes ordenes
  - Aleatorio:

    palabras orden aparecen texto comparan independientemente
  - Alfabético

    aparecen comparan independientemente orden palabras texto
  - Real

    Comparan palabras independientemente orden aparecen texto

Puede parecer extremo pero….

¿ De qué trata este documento ?

Documento original

Un estudiante con parálisis cerebral crea una aplicación para encontrar aparcamientos de movilidad reducida

Calos Cobos junto a los tutores de su proyecto, Alberto Íñigo y Nicolás Marín. / ALFREDO AGUILAR

El prototipo que ha creado un alumno de la UGRmuestra las plazas libres e identifica los usuarios que las utilizan sin tener autorización

Cada vez que Carlos va al Centro de Granada con su coche y se dispone a aparcarlo, se encuentra con la misma estampa: las plazas para personas con movilidad reducida –que debe utilizar a causa de su parálisis cerebral– suelen estar ocupadas por conductores que no cuentan con ningún tipo de discapacidad. Esto provoca que Carlos, como le sucede a la inmensa mayoría de personas que hacen uso de estos aparcamientos, tenga que perder tiempo esperando a que se vaya el coche, llamando a la Policía o dando vueltas por la ciudad en busca de otro hueco para aparcar su vehículo. Ahí se topó de lleno con el problema, le faltaba encontrar la solución. Y lo hizo gracias a sus estudios de Ingeniería Informática que recién acaba de finalizar en la Escuela de Informática y Telecomunicaciones de la Universidad de Granada. ...

Modelos RI para texto

- Usualmente,

  basados en bag of words

- cada modelo incluye una forma de definir un **peso** para cada palabra en cada documento.

Una función de recuperación f(q,D) --> R, función de ranking, se encarga de combinar los distintos pesos

Resumen de Modelos

- Clásicos
  - Booleano
  - Modelo de Espacio Vectorial
- Modelos Probabilísticos
  - BM25
  - Modelado del Lenguaje

Modelo Booleano

- Es el modelo más simple, basado en teoría de conjuntos

- Documento se representan como conjunto de términos
  - cada término toma dos valores
    - g(term_q,doc) = 1 si term pertenece al doc.
    - g(term_q, doc) = 0 si term NO pertenece al doc.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Modelos Booleano

- Hay dos posibles salidas
  - True y False (relevante/no relevante)
  - “Emparejamiento exacto”
  - No hay ranking

- Los primeros sistemas comerciales eran booleanos (3 décadas), y aun a dia de hoy muchos sistemas lo siguen siendo
  - Email, catálogos de bibliotecas, etc
  - Asumen usuarios expertos (búsqueda en textos legales, médicos, etc.)
  - Búsqueda por números es común.

Modelo Booleano

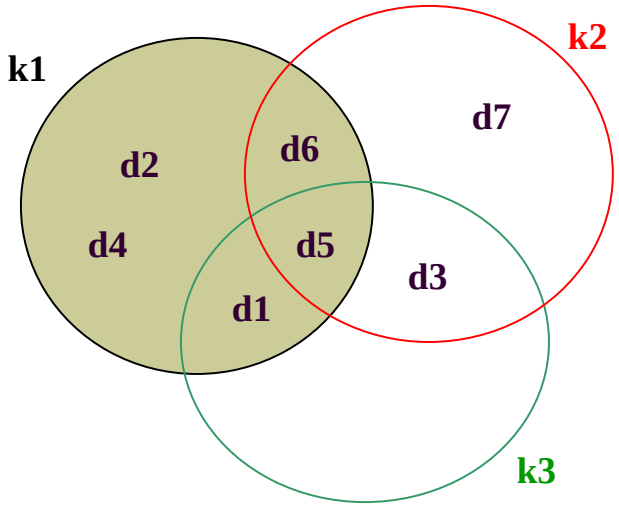
Consultas se representan como expresiones booleanas

- Conjunto de términos relacionados con conectores AND, OR, NOT.
  - *q1 = ta ^ tb*
  - *q2 = ta ˅ tc*
- La consulta original se transforma en consultas menores que se pueden lanzar de forma independiente y el resultado final se obtiene mezclado las distintas salidas.

  Se transforma en la forma normal disyuntiva

  q3 = ta ^ (tb ˅ ¬tc) =
  (ta^tb^tc) ˅ (ta^tb^¬tc) ˅ (ta^ ¬tb^¬tc)
- Pueden aplicar operadores de proximidad y comodines
- Permiten consultar por otras características como fecha
  - *formalismo claro y semántica precisa*

El modelo Booleano: Ejemplo



	k1	k2	k3	consulta
d1	1	0	1	
d2	1	0	0	Q1
d3	0	1	1	
d4	1	0	0	Q1
d5	1	1	1	Q0,Q1
d6	1	1	0	Q1
d7	0	1	0	Q2
Q0 = k1 ^ k2 ^ k3	Q1=k1 ^ (k2 ˅ ¬k3)		Q2=¬k1 ^ k2 ^ ¬k3	

Desventajas del Modelo Booleano

- El mundo no es ni blanco ni negro, también hay grises:
  - Recuperación está basada en criterios binarios, sin dar opción a un emparejamiento parcial
- No presenta los documentos por orden de relevancia (no puede proporcionar un grado de relevancia)
- Los usuarios pueden tener problema a la hora de especificar la consulta:
  - La necesidad de información debe ser expresada por una expresión booleana.

    Por tanto, las consultas tienden a ser muy simples
  - Por tanto, o bien un SRI booleano devuelve demasiados o muy pocos documentos al usuario.

Modelo Binario ==> Ranking Retrieval (ordenar)

- Objetivo: Devolver los mejores (-top) documentos ordenados según relevancia para la consulta
  - Permiten realizar consultas de texto libre.
  - En general, lo importante son los k primeros documentos, no tratan de recuperar un conjunto grande de docs

- Cómo se podría realizar el ranking?
  - Para cada doc asignar una puntuación (en un rango [0,1]) de relevancia para consulta q

Coeficiente de Jaccard

- Permite medir el solapamiento entre dos conjuntos A y B

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

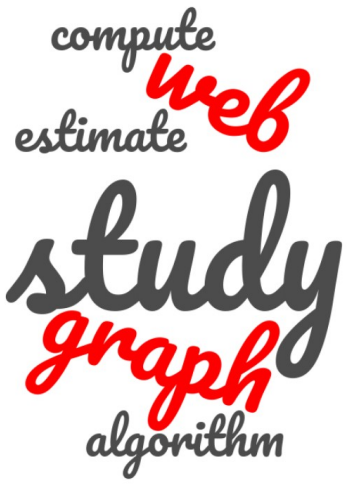
  - Jaccard (A,A) = 1;
  - Jaccard(A,B)=0 si no tienen términos en común

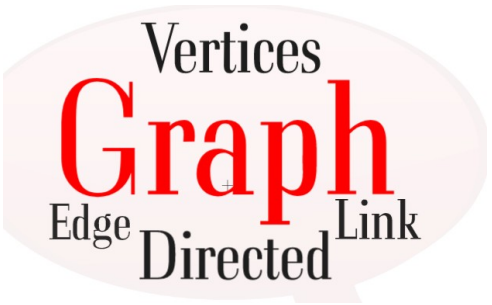
- Los conjuntos no tienen que tener el mismo tamaño.
  - Q= Don Quijote
  - D1 = singular batalla el jamás como se debe alabado caballero don Quijote
  - D2 = Cualquiera yantaría yo, respondió don Quijote?,

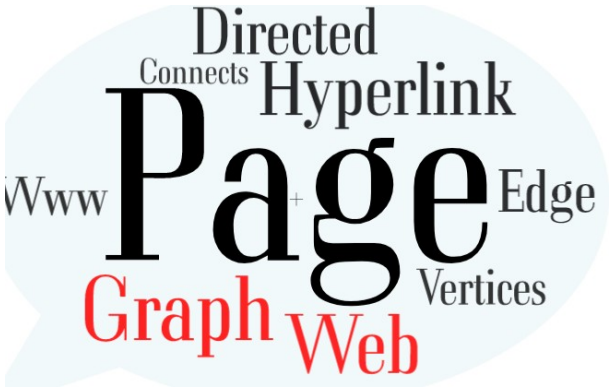
Asignando valores al ranking

- Por un lado, parece sensato que cuanto más términos de Q en el documento, mayor debería ser su valor.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$





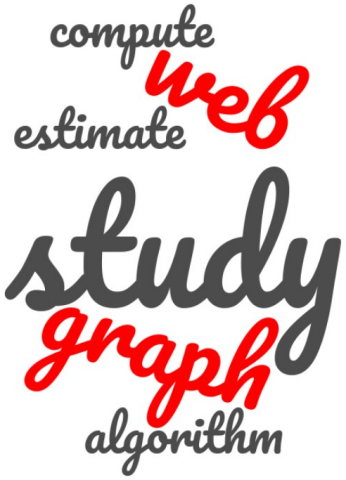


… Sesgo hacia documentos largos.

Penalizando la longitud del documento

- Por otro lado, los documentos mas grandes (con mayor número de términos) tienen una probabilidad mayor de emparejar con un gran número de consultas.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



web
graph ?

Problemas medida Jaccard

- El score decrece con la longitud del documento, pero tampoco hay que penalizar en exceso, ya que suelen tener mayor contenido
  - Necesitamos poder suavizar por la longitud del documento, una alternativa puede ser:

$$Jaccard_s(A,B) = \frac{|A \cap B|}{\sqrt{|A \cup B|}}$$

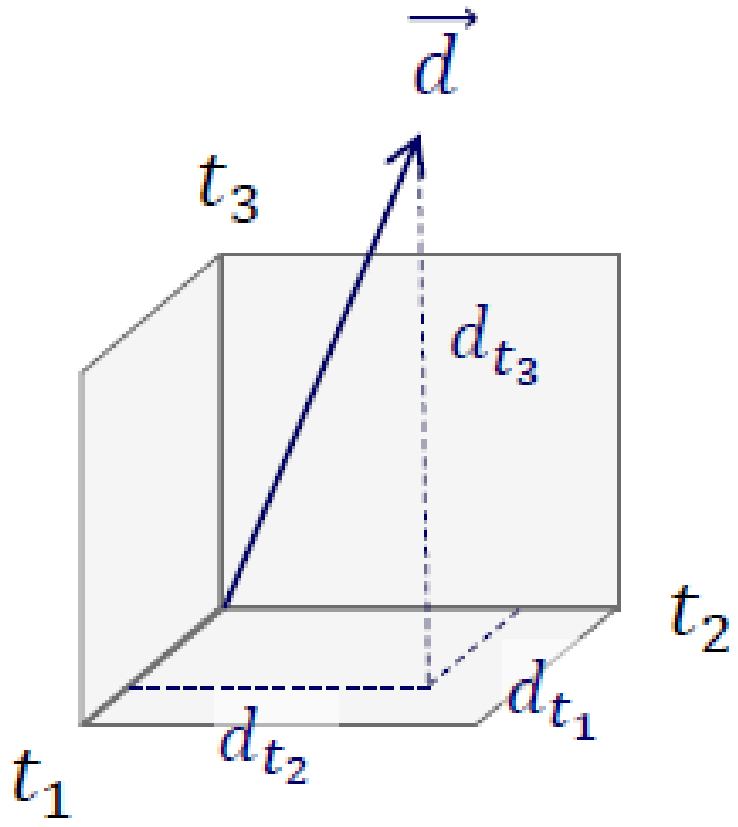
- Como modelo booleano, no considera la frecuencia del término en el documento.

Esquema

- Introducción a los Modelos de Recuperación de Información
- Modelos Clásicos
  - Booleanos (Clásico, Jaccard,Comb. lineal
  - Vectorial
  - Probabilístico

Modelo de Espacio Vectorial

- Se representan docs y consultas en un espacio vectorial, R^V
  - donde V es el vocabulario
  - bag of words
- Las coordenadas de los vectores para cada t en V son pesos d_t = w(t,d), que cuantifican como de representativo es un término en el doc.



Documentos son vistos como vectores con peso

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

Comparar consulta con doc

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

$Score(q,d)=(?)$

	database	SQL	index	regression	likelihood	linear
Q	1	0	0	0	0	0

Comparar consulta con doc

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

$$Score(q,d) = \sum_{t \in V} w_{t,d} * w_{t,q}$$

Peso es cero si un término no está en doc o consulta

$$Score(q,d) = \sum_{t \in d \cap q} w_{t,d} * w_{t,q}$$

	database	SQL	index	regression	likelihood	linear
Q	1	0	0	0	0	0

Elementos que forman parte de la definición de los pesos:

Buscamos algo como ….

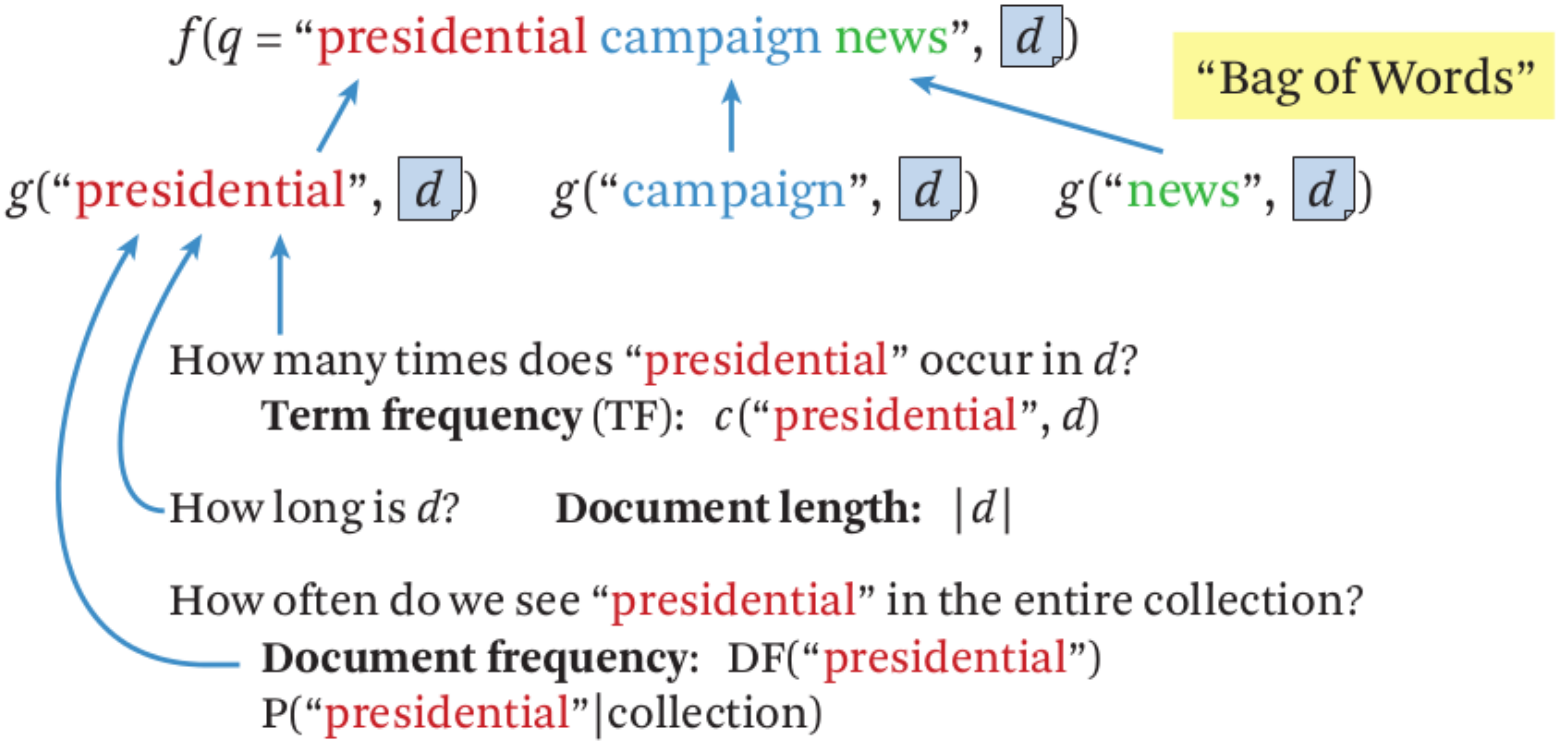


Illustration of common ideas for scoring with a bag-of-words representation.

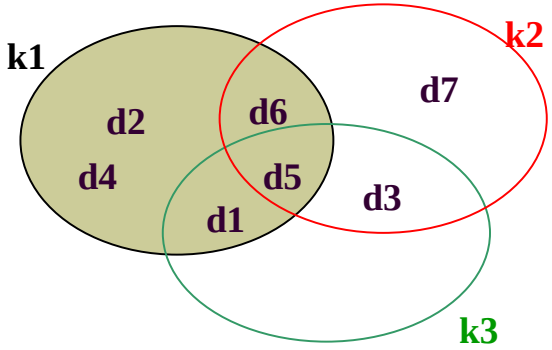
de: Text data management and analysis

Cómputo de los pesos: frecuencia de términos

- Supongamos que ante una consulta *q*,

$$Score(q,d) = \sum_{t \in q \cap d} t_{t,d} * t_{t,q}$$

- Ordenar los documentos según score



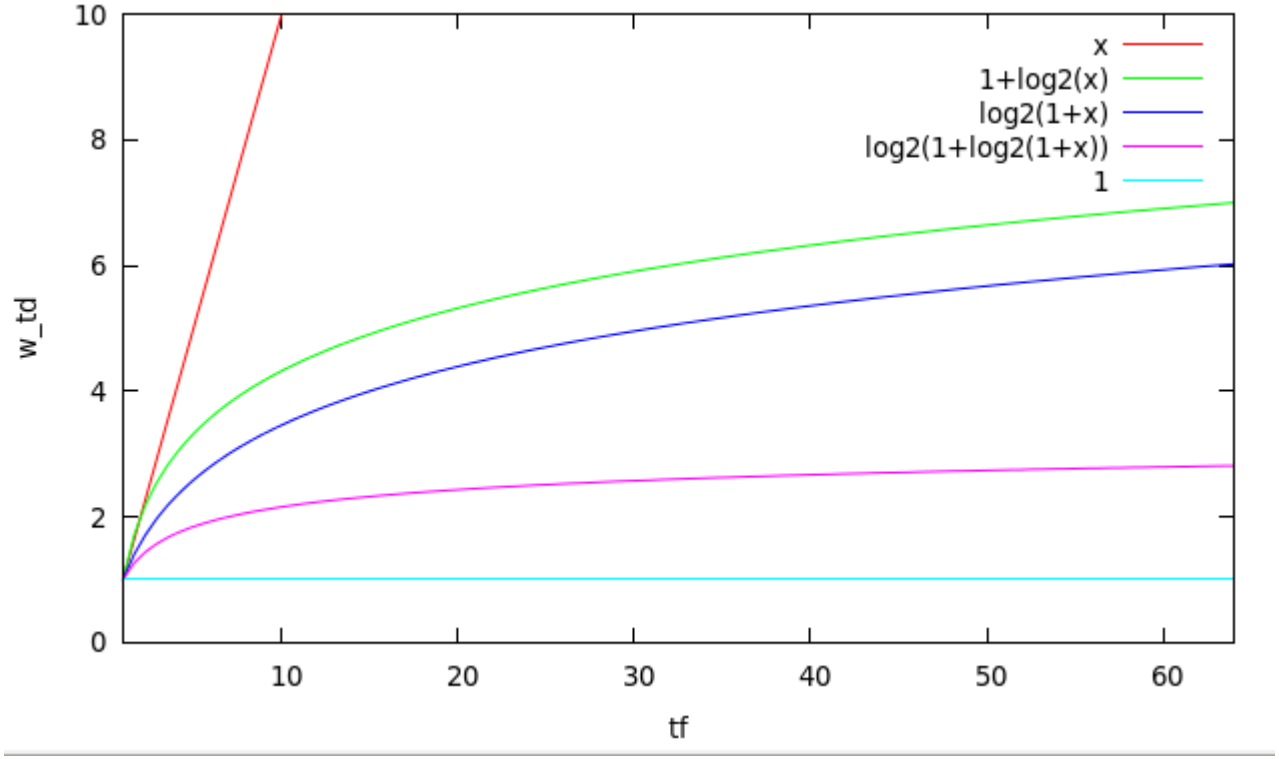
	k1	k2	k3	q • dj
d1	1	0	16	?
d2	8	0	0	?
d3	0	4	32	?
d4	1	0	0	?
d5	64	32	4	?
d6	1	256	0	?
d7	0	256	0	?
q	1	1	1	

Frecuencia del término

- La frecuencia del término, $tf_{t,d,}$ en el documento podría no ser realmente lo que buscamos...
  - Un documento con 10 ocurrencias de un término es más relevante que otro con sólo 1 ocurrencia de dicho termino
  - Pero no es 10 veces más relevante ...
- Hipótesis:
  - La relevancia no se incrementa de forma proporcional con la frecuencia.

Pesado basado en log de frecuencia

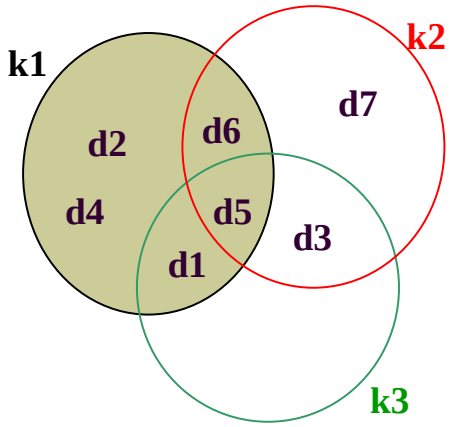
Se transforma la frecuencia del término en documento



- Dada una consulta *q*, en texto libre

$$Score(q,d) = \sum_{t \in d \cap q} w_{t,d} * w_{t,q} = \sum_{t \in d \cap q} (1 + \log(tf_{t,d})) * tf_{t,q}$$

Ordenar los documentos según score



	k1	k2	k3	q • dj
d1	1	0	16	?
d2	8	0	0	?
d3	0	4	32	?
d4	1	0	0	?
d5	64	32	4	?
d6	1	256	0	?
d7	0	256	0	?
q	1	1	1	

Ordenar según frecuencia de términos

- Problema:
  - Consulta "***ides of march***"
    - *Julius Caesar* tiene 5 ocurrencias de ***ides,***
    - Ninguna otra obra contiene ***ides,***
    - ***march*** ocurre en una docena,
    - Todas las obras incluyen ***of,***
  - Utilizando este criterio, la obra que está más arriba es la que tiene mas ***of*** .
- Queremos atenuar la influencia de los términos comunes. Pero, ¿Qué es común?

Sugerencia: Considerar la frecuencia **documental** del término en la colección (d*f*)

Tenemos en cuenta ambos criterios

- Importancia del término en el documento (tf):
  - Un término que aparece muchas veces en un documento es más importante que otro que sólo aparece una vez
    - *tf frecuencia del término en el documento*

- Importancia del término en la colección (idf):
  - un término que aparece pocas veces en la colección tiene un mayor poder discriminador que un término que aparezce en todos los documentos.
    - *idf (inverse document frequency) freq. documental inversa*

Esquema *tf-idf*

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

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

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

$\mathcal{D} =$ la colección de documentos (espacio de búsqueda)

$\mathcal{D}_t =$ documentos que contienen el término t

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

Esquema *tf-idf*

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

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

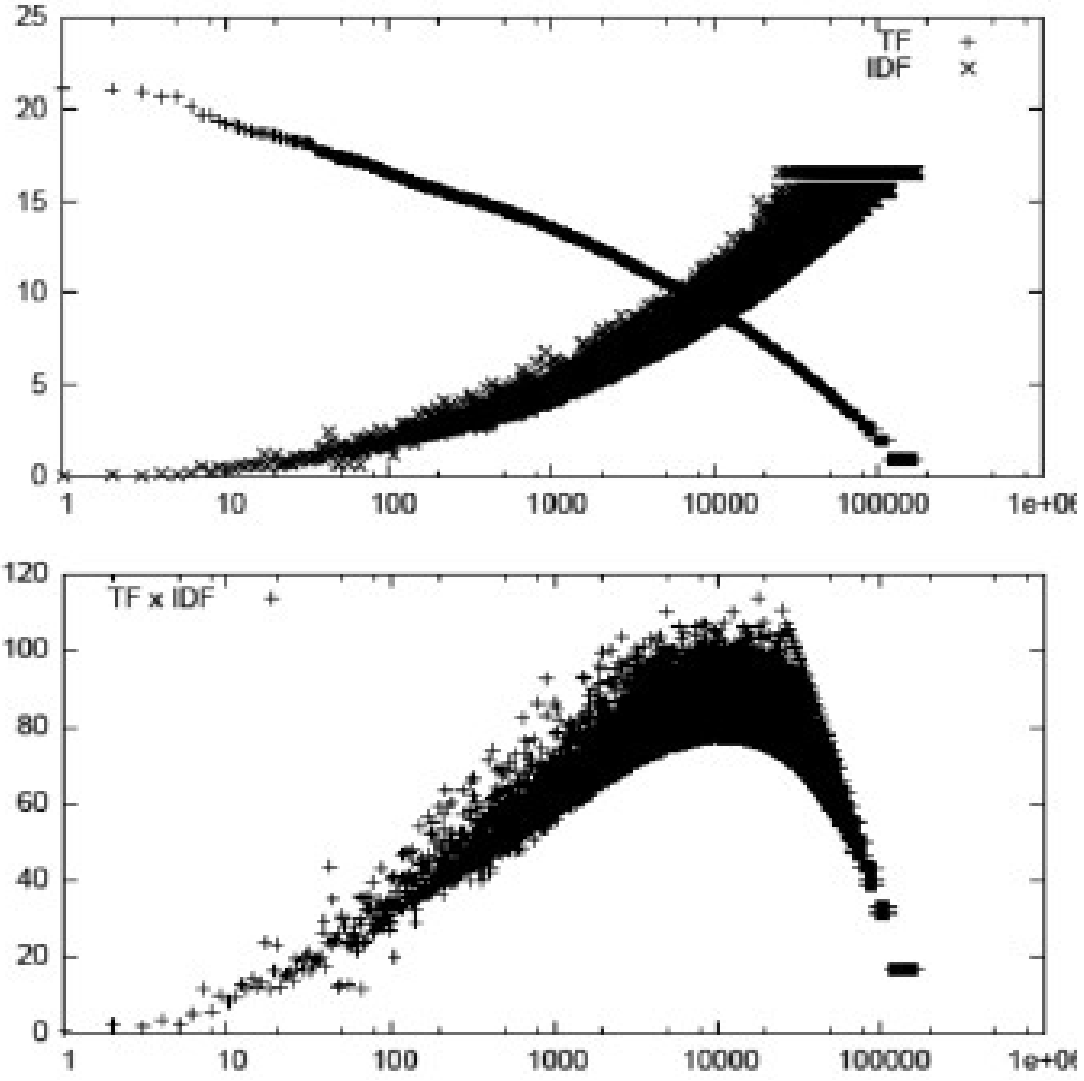
$$idf(t) = \log\left(1 + \frac{|\mathcal{D}|}{1 + |\mathcal{D}_t|}\right)$$

...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



R. Baeza and B. Ribeiro. Modern Information Retrieval 2nd ed. Addison-Wesley 2012

Modelo Vectorial: Ejemplo

consulta		
q	*web graph*	
documentos	texto	términos
d_1	*The web graph web graph graph ...*	*web graph*
d_2	*Is a … web … net … graph … net … web … graph … web net page*	*graph web net page*
d_3	*A complex web page web web …*	*page web complex*

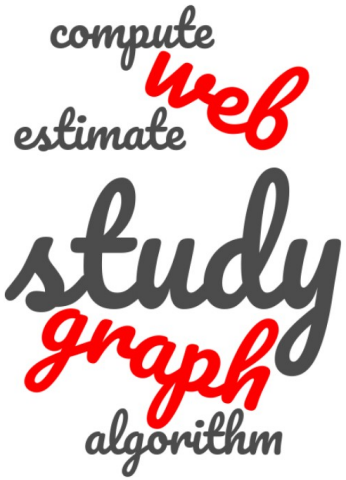
	web	graph	net	page	complex
q	1	1	0	0	0
D1	4	4	0	0	0
D2	4	5	98	42	0
D3	10	20	150	300	526

¿Qué salida obtenemos ?

¿ Qué nos falla?

Longitud del documento

web
graph ?



Longitud del documento

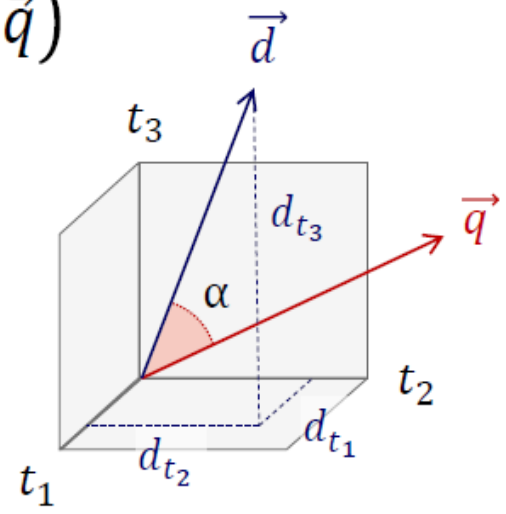
- Los documentos mas grandes (con mayor número de términos) tienen una probabilidad mayor de emparejar con un gran número de consultas.
  - Sesgo hacia documentos largos.
  - Pero tampoco hay que penalizar en exceso, ya que suelen tener mayor contenido

Modelo de espacio vectorial:
Medida COSENO

- Comparamos documento, d, y consulta, q, considerando el ángulo entre los dos vectores

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

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

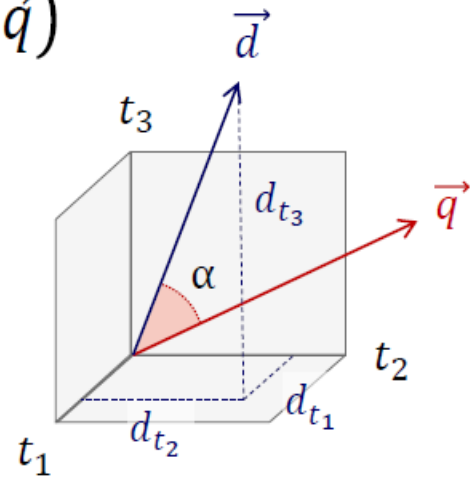


-

Modelo de espacio vectorial: función de ranking

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

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



Se el peso de un término que no está en el documento es cero

$$\cos(\vec{d},\vec{q}) = \frac{\sum_{t\in d\cap q} w_{t,d} * w_{t,q}}{\sqrt{\sum_{t\in d} w_{t,d}^2}\sqrt{\sum_{t\in q} w_{t,q}^2}}$$

Modelo de espacio vectorial: función de ranking

$$\cos(\vec{d},\vec{q}) = \frac{\sum_{t} w_{t,d} * w_{t,q}}{\sqrt{\sum_{t \in d} w_{t,d}^2}\sqrt{\sum_{t \in q} w_{t,q}^2}}$$

- Propiedades:
- cos(d,q) = 1 si los dos vectores son iguales
- cos(d,q) = 0 si no tienen términos en común (ortogonales)
- No es sensible a los cambios de escala (multiplicar pesos por cte)
- Si los pesos son vectores unitarios

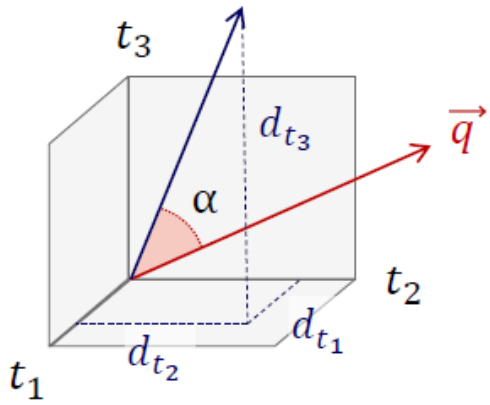
$$w^{\circ}_{t,d} = \frac{w_{t,d}}{\sqrt{\sum_{j \in d} w_{j,d}^2}} \quad \text{y} \quad w^{\circ}_{t,q} = \frac{w_{t,q}}{\sqrt{\sum_{j \in q} w_{j,q}^2}}$$

Entonces

$$\cos(\vec{d},\vec{q}) = \sum_{t \in d \cap q} w^{\circ}_{t,d} * w^{\circ}_{t,q}$$

Modelo de espacio vectorial: algunas notas sobre su implementación

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



- Vector q:
  - Se puede hacer tf binario (salvo q = doc)
  - Idf penaliza doblemente los términos comunes (se puede omitir)
  - Se puede omitir |q| en el denominador, = ranking
- Normalizar |d|
  - |d| evitar el sesgo a documentos largos.
  - Se puede precomputar.

Ejemplo:

$1 + \log_2 frec(t,d)$ ↑ $tf(t,d)$; $\log \frac{|\mathcal{D}|}{|\mathcal{D}_t|}$ ↗ $idf(t)$

$frec(t,d)$

	d_1	d_2	d_3	d_4
arbol	4			
hoja		4	2	
olivo			1	1
raiz			4	1
rama	1	4	2	1
savia	4		1	

$tf(t,d)$

	d_1	d_2	d_3	d_4
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

$idf(t)$
2
1
1
1
0
1

$tf\text{-}idf(t,d)$

	d_1	d_2	d_3	d_4
arbol	6	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

q
1
1
1

$\|d\|$	$\sqrt{45}$	3	$\sqrt{15}$	$\sqrt{2}$

$\sqrt{3}$	$\|q\|$

q = “hoja arbol olivo”

$\frac{d \cdot q}{|d||q|}$ ← $\cos(d,q)$

0.52	0.58	0.45	0.41

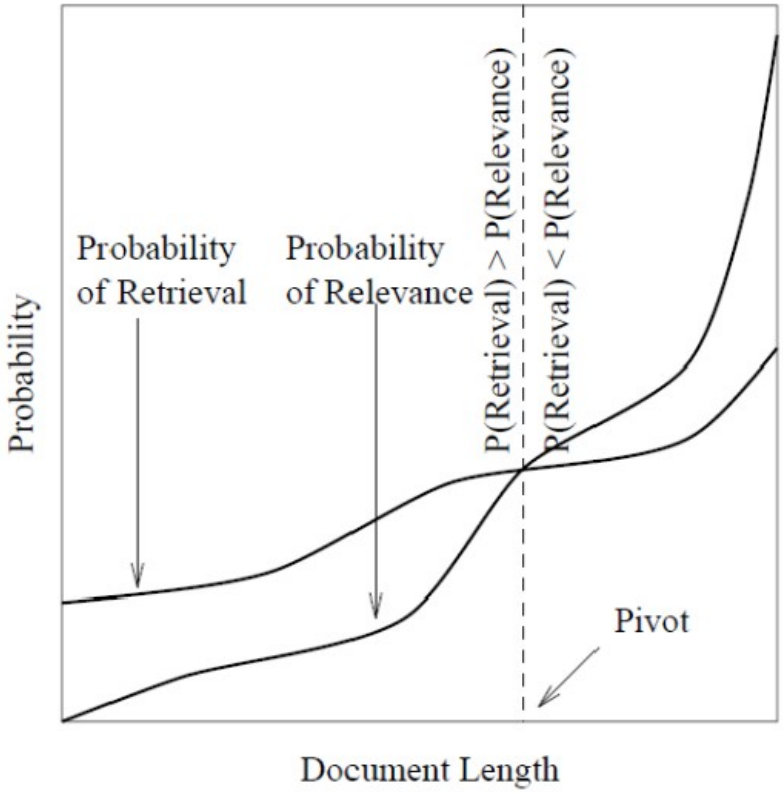
Modelo de Espacio Vectorial

- El modelo vectorial con pesos *tf-idf* representa una buena estrategia en colecciones generales.
- El modelo vectorial es competitivo frente a otras alternativas. Es simple y fácil de calcular
- Ventajas:
  - mejora la calidad recuperadora del sistema
  - permite recuperar documentos que sólo se emparejan parcialmente con las consultas.
  - permite ordenar los documentos por grado de similaridad con respecto a la consulta.
- Desventaja:
  - asume independencia de términos

Normalizar por pivote

- Propuesto por Amit Singhal et al. (fué Vicepresidente en Google) 1996
  - *A. Singhal, C. Bucley, M. Mitra. Pivoted Document Length Normalization. SIGIR 1996, pp 21-29*
- Hoy en día se incorpora de forma habitual en las implementaciones.
- Problema:
  - La norma del coseno es demasiado severa en general, penalizando en exceso los documentos grandes. Experimentalmente se demuestra que éstos tienden a ser más relevantes a la consulta.

Solución: norm(d) < |d|



Ejemplo Coseno
Norma L2 penaliza docs largos

- Asumamos los siguientes pesos.
  - notar similitud términos en Q y docs

Q	0	0	1	0	1	0	1	0	0
d1	10	15	3	4	3	1	1	18	8
d2	0	1	3	4	3	0	1	4	1
d3	10	5	3	5	3	0	1	2	2
d4	20	30	6	8	6	2	2	36	16

$$\cos(\vec{d},\vec{q}) = \frac{\sum_t w_{t,d} * w_{t,q}}{\sqrt{\sum_{t\in d} w_{t,d}^2}\sqrt{\sum_{t\in q} w_{t,q}^2}}$$

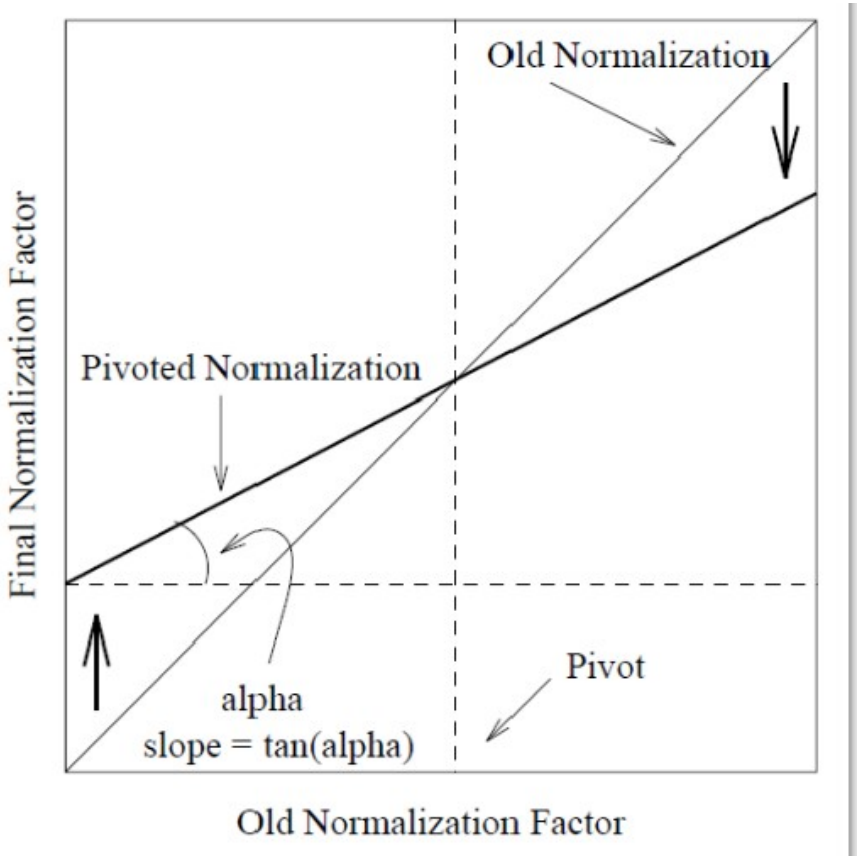
L2 Norma

d1	27,36
d2	7,280
d3	13,30
d4	54,73

Coseno
0,255
0,961
0,526
0,255

Normalizar por pivote

- La normalización depende de la longitud media de los documentos en la colección avdl
  - |d| > avdl → entonces se penaliza un poco
  - |d| = avdl → no se penaliza,
  - |d| < avdl → se puede favorecer



Normalización por el pivote

- Utilizan un parámetro,b, que controla el grado de penalización.
  - $b \in [0,1]$
  - su valor se obtiene de forma experimental.

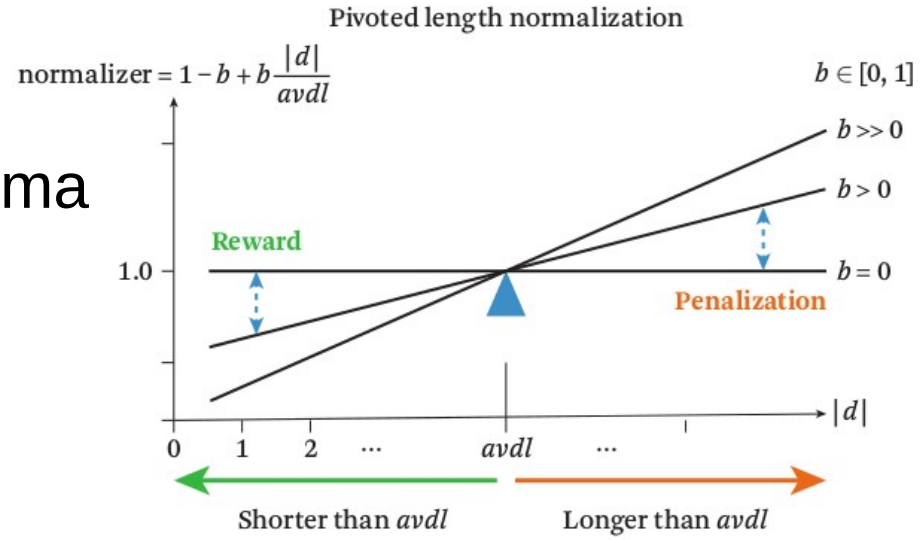


Illustration of pivoted document length normalization.

Modelo de Espacio Vectorial Norm Pivote

Estado del arte ….

$$score\,VSM(d,q) = \sum_{t \in d \cap q} tf_{t,q} \frac{\ln(1+\ln(1+tf_{t,d}))}{1-b+b\frac{|d|}{avdl}} \log\left(\frac{M+1}{df(t)}\right)$$

- IDF, depende de la importancia del t en la colección,
  - M número total de docs, df(t) #docs que contienen t
- TF, frecuencia del término en el documento
- |d|, longitud del documento,
  - b se optimiza para cada colección, |d| puede ser norma L2, avdl longitud media
- Frecuencia término en Q

Ejemplo Coseno Norma L2 vs NormPivote

- Asumamos pesos calculados como tfidf.
  - notar similitud términos en Q y docs

Q	0	0	1	0	1	0	1	0	0
d1	10	15	3	4	3	1	1	18	8
d2	0	1	3	4	3	0	1	4	1
d3	10	5	3	5	3	0	1	2	2
d4	20	30	6	8	6	2	2	36	16

L2 Norma

d1	27,36
d2	7,280
d3	13,30
d4	54,73

Coseno	Pivote b=0.2	Pivote b=0.8
0,255	6,908	6,648
0,961	8,170	16,39
0,526	7,746	11,38
0,255	11,41	7,346

Esquema

- Introducción a los Modelos de Recuperación de Información
- Modelos Clásicos
  - Booleano
  - Vectorial
  - Probabilístico
    - Modelo Binario
    - OKAPI BM25
    - Modelos Basados en Modelado del Lenguaje

Modelos de R. I. Probabilísticos

- El objetivo es abordar la problemática de la RI utilizando el formalismo probabilístico
- Asume que dada una consulta del usuario, hay una respuesta *ideal*
- Una consulta es una especificiación de las propiedades de dicha respuesta ideal
- Los modelos probabilísticos, siendo de los más antiguos, son hoy día unos de los más estudiados
  - *Los modelos tradicionales se basaban en ideas claras, pero no solían ganar en comportamiento. Actualmente la situación ha cambiado.*

Modelos RI Probabilísticos

Razones para usar la probabilidad:

- La disciplina de la RI esta plagada de incertidumbre
  - En la representación del contenido de los documentos (indexación)
  - En la descripción de la necesidad de información del usuario (consulta)
- La probabilidad proporciona una base teórica sólida para el manejo de incertidumbre y por tanto para el diseño de sistemas de R.I.

Modelos de R.I. Probabilísticos

Objetivo:Para cada documento y consulta, tratan de responder a la siguiente pregunta:

¿*Cuál es la probabilidad de que este documento sea relevante a esta consulta*?

- Se basa en el *Probability Ranking Principle (Robertson, 1977):*
  - *"La efectividad global de un sistema es la mejor posible cuando los documentos se presentan en orden creciente de la probabilidad de relevancia*

  (donde las probabilidades se estiman de la forma más precisa posible)"

Prob. Ranking Principle

Asumimos un caso simple:

- No hay costos asociados para recuperar un documento
- No se utilizan utilidades

- ***La decisión optimal según la Regla de Bayes***

D *es relevante sii*

$$p(R|D) > p(NR|D)$$

Modelos de R.I. Probabilísticos

Esquema de un Modelo Probabilístico:

- Estimar cómo los términos contribuyen a la relevancia (¿Cómo se determinan los valores de probabilidad?)
  - Binary Independence Retrieval (BIR) – es el modelo más simple
  - Fórmula OKAPI (S. Robertson)
    - Considera conceptos como tf, df, y longitud influyen en el concepto de relevancia

- Combinar, esto es, encontrar la relevancia del documento
- Ordenar los docs en orden decreciente prob.

OKAPI: Modelos Probabilísticos

- OKAPI: Es el nombre dado a una familia de SRI experimentales basados en el modelo probabilístico de Robertson-Spark Jones.

- Considera hasta cinco factores distintos para dar un peso a cada par término-documento. :
  - Frecuencia en la colección
  - Frecuencia en el documento
  - Información de relevancia
  - Longitud del documento
  - Frecuencia en la consulta

- Finalmente estos valores individuales se combinan para dar un peso final que nos diga cómo de relevante es el documento a la consulta.

OKAPI: Cálculo de Pesos

- Frecuencia del término en el documento.

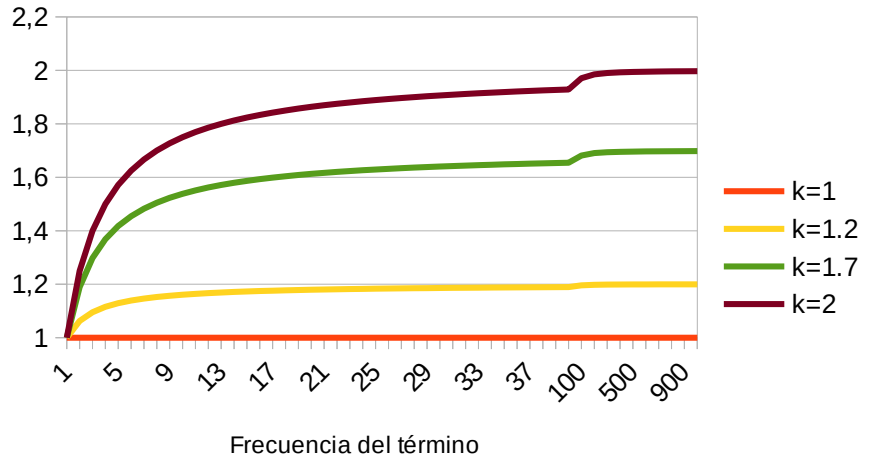
Según Robertson, Zaragoza, y Taylor (2004)

“Most modern weighting functions based on term frequencies (tf) are nonlinear in this parameter. This is desirable because of the statistical de pendence of term occurrences : the information gained on observing a term the first time is greater than the information gained on subsequently seeing the same term”

OKAPI: Cálculo de Pesos

- Frecuencia del término en el documento.

$$w_{t,d} = \frac{tf_{t,d} * (k+1)}{tf_{t,d} + k}$$



Propiedades:

- es cero para TF=0
- crece con TF
- tiene un límite asintótico (k+1), nos permite controlar el peso máximo (elevar la frecuencia de aparición del término no afecta a la relevancia → spam
- K es constante que mide cómo crece el peso con TF, (valores entre 1.2-2)
- K=0 modelo booleano (no tiene en cuenta la frec)
- Valores altos de K, el incremento en la frecuencia sigue manteniendo influencia en los pesos.

OKAPI: Cálculo de Pesos

- Tener en cuenta la longitud del documento

  La longitud del documento dependerá de:

  – Verbosidad: algunos autores utilizan más palabras que otros para decir lo mismo => conviene considerar por la longitud del doc.

  – Ámbito: Algunos autores tienen más que decir => no es conveniente considerar la longitud

  Suele ser una mezcla de ambas situaciones

- Se puede considerar distintas formas para medir la longitud del documento:
  - bytes, línea, sentencia, párrafos, …
  - número de palabras únicas (con o sin stopwords)
  - número de palabras (es el que se suele utilizar)

OKAPI: Cálculo de Pesos

- Tener en cuenta la longitud del documento

  – La frecuencia del término se suele normalizar por la longitud de documento medida como

$$L_d = \left(1 - b + b\,\frac{|d|}{avdl}\right)$$

Donde b es una constante (aprox. 0.75), |d| es la longitud del documento, y AVDL el la longitud media del documento

  – y por tanto $tf^n_{t,d} = \frac{tf_{t,d}}{L_d}$

que al sustituir en $w_{t,d}$ queda como

$$w^n_{t,d} = \frac{\frac{tf_{t,d}}{L_d} \times (k+1)}{\frac{tf_{t,d}}{L_d} + k} = \frac{tf_{t,d} \times (k+1)}{tf_{t,f} + k \times (1 - b + b \times |d| / avdl)}$$

OKAPI: Cálculo de Pesos

Además BM25 puede tener en cuenta:

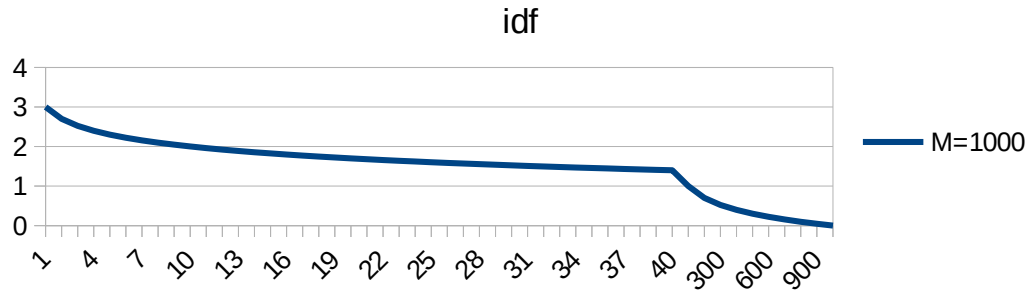
- El peso del término en la consulta, ya que un documento grande podría ser parte una consulta,.

$$tf_{t,q} = 1 \; si \; consulta \; corta$$

$$tf_{t,q} = \frac{tf_{tq} \times (k_2 + 1)}{tf_{tq} + k_2} \; si \; consulta \; larga$$

- La frecuencia del término en la colección

$$idf_t = \log\left(\frac{M+1}{df(t)}\right)$$



Modelo Okapi BM25 (Best Match)

Estado del arte ….

$$score\ BM\ 25(d,q) = \sum_{t \in d \cap q} tf_{t,q} \frac{(k+1)\, tf_{t,d}}{tf_{t,d} + k\left(1 - b + b\frac{|d|}{avdl}\right)} \log\left(\frac{M+1}{df(t)}\right)$$

- IDF, depende de la importancia del t en la colección,
  - M número total de docs, df(t) #docs que contienen t
- TF, frecuencia del término en el documento
- |d|, longitud del documento,
  - b se optimiza para cada colección, |d| puede ser norma L2, avdl longitud media
- Frecuencia término en Q

Esquema

- Introducción a los Modelos de Recuperación de Información
- Modelos Clásicos
  - Booleano
  - Vectorial
  - Probabilístico
    - Modelos Basados en Modelado del Lenguaje

¿Qué es un modelo probabilístico de lenguaje?

- Consiste en definir una distribución de probabilidad sobre un conjunto de palabras (alfabeto) para un determinado modelo del lenguaje, M, (idioma, temática, …)

  – Es un componente esencial para el procesamiento de lenguaje natural (PLN)

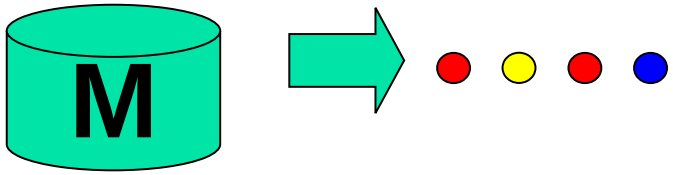
- Por ejemplo, sea S = “marca un gol por la escuadra”
  - P(S | deportes) > P(S | quijote)
  - P(S |español) > P(S | inglés)

¿Como se calcula

P(“marca un gol por la escuadra”| español) ?

Modelos de lenguajes probabilísticos

- Un MLP modela la *probabilidad* de generar cadenas en el lenguaje (usualmente todas las cadenas en un alfabeto ∑)



P (● ● ● ● | M) = P (● | M)

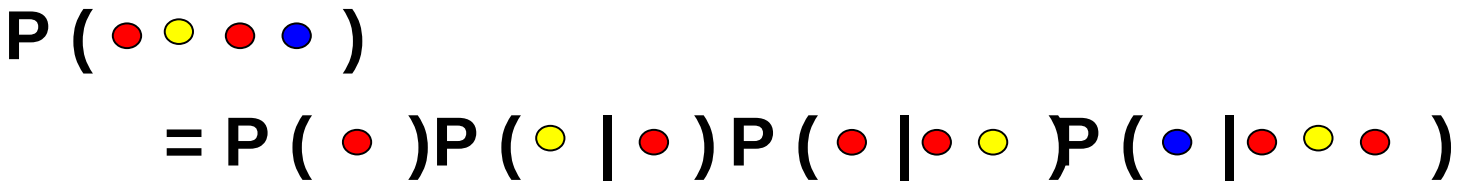
P (● | M, ●)

P (● | M, ● ●)

P (● | M, ● ● ●)

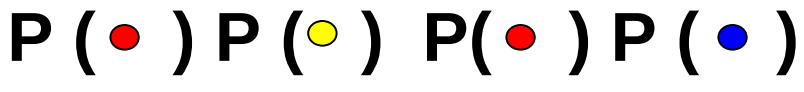
Intuitivamente, el i-ésimo factor es la probabilidad de que (en el lenguaje modelado M) aparezca la palabra wi continuación de la secuencia w1,...,wi−1

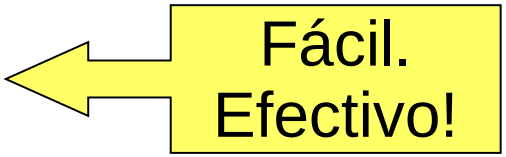
Modelos de lenguaje probabilísticos



Asumimos Independencia:
Unigramas y modelos de orden mayor

Unigram: Modelo más simple





Bigram basado en coocurrencias (dos palabras consecutivas)

En general, *n*-gram

Modelos de lenguajes probabilísticos

- Ejemplo:

Modelo M

0.2	the
0.1	a
0.01	man
0.01	woman
0.03	said
0.02	likes
…	

the	man	likes	the	woman
0.2	0.01	0.02	0.2	0.01

multiplicar

P(s | M) = 0.00000008

Modelos de lenguajes probabilísticos

- *Probabilidad* de generar una cadena dada

Model M1

0.2	the
0.01	class
0.0001	sayst
0.0001	pleaseth
0.0001	yon
0.0005	maiden
0.01	woman

Model M2

0.2	the
0.0001	class
0.03	sayst
0.02	pleaseth
0.1	yon
0.01	maiden
0.0001	woman

	the	class	pleaseth	yon	maiden
M1	0.2	0.01	0.0001	0.0001	0.0005
M2	0.2	0.0001	0.02	0.1	0.01

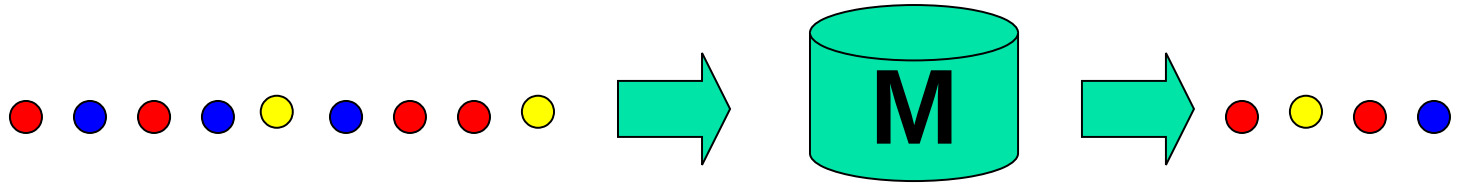
$$P(s|M2) > P(s|M1)$$

El problema esencial de LM

- Usualmente no conocemos el modelo **M**
  - Pero tenemos un texto que representa el documento, doc = (●●●●●●●●●●)

P(●●●●|M(●●●●●●●●●●)

- Estimar un LM a partir del ejemplo (docs)
- Entonces se computan las probabilidad de las observaciones



Dos noticias ... del mundo

El presidente valenciano confirma que la decision de cerrar RTVV es 'innegociable'. 'No voy a ser yo el que ponga 40 millones para poder cumplir la sentencia del TSJ'. 'Ante todo tenemos que mantener nuestros servicios basicos', dice el jefe del Consell. Fabra asegura que es 'la decision mas dificil' que ha tomado desde que es presidente. Anuncia la inminente dimision de Rosa Vidal y elude cualquier responsabilidad politica .....

La Fundacion Progreso y Democracia -dependiente de UPyD- ha calculado cual seria el coste economico y politico de una hipotetica independencia de Cataluña y del Pais Vasco, aun considerando que ambas comunidades siguieran en la UE. Y sus resultados son demoledores.

El informe -que se presentara hoy - calcula que el PIB catalan retrocederia un 10% si esta comunidad autonoma pasara a ser independiente. .....

- CANAL 9

0 rtvv 0.0214477

1 fabra 0.0187668

2 presidente 0.0162858

3 millones 0.0162858

4 decision 0.0162858

5 televisión 0.01604

6 gobierno 0.0134048

7 comunidad 0.0134048

8 asegurado 0.0134048

9 sentencia 0.0134048

10 cierre 0.0107239

11 coste 0.0107239

12 servicios 0.0107239

13 político 0.0107239

14 explicado 0.0107239

- CATALUÑA

0 españa 0.0402145

1 upyd 0.0187668

2 independencia 0.0187668

3 cataluña 0.0187668

4 calcula 0.0107239

6 ciudadanos 0.0107239

7 vasco 0.0080429

8 hipotetica 0.0080429

9 pais 0.0080429

10 político 0.0080429

11 caida 0.0080429

12 cierre 0.0080429

…..

34 coste 0.001012

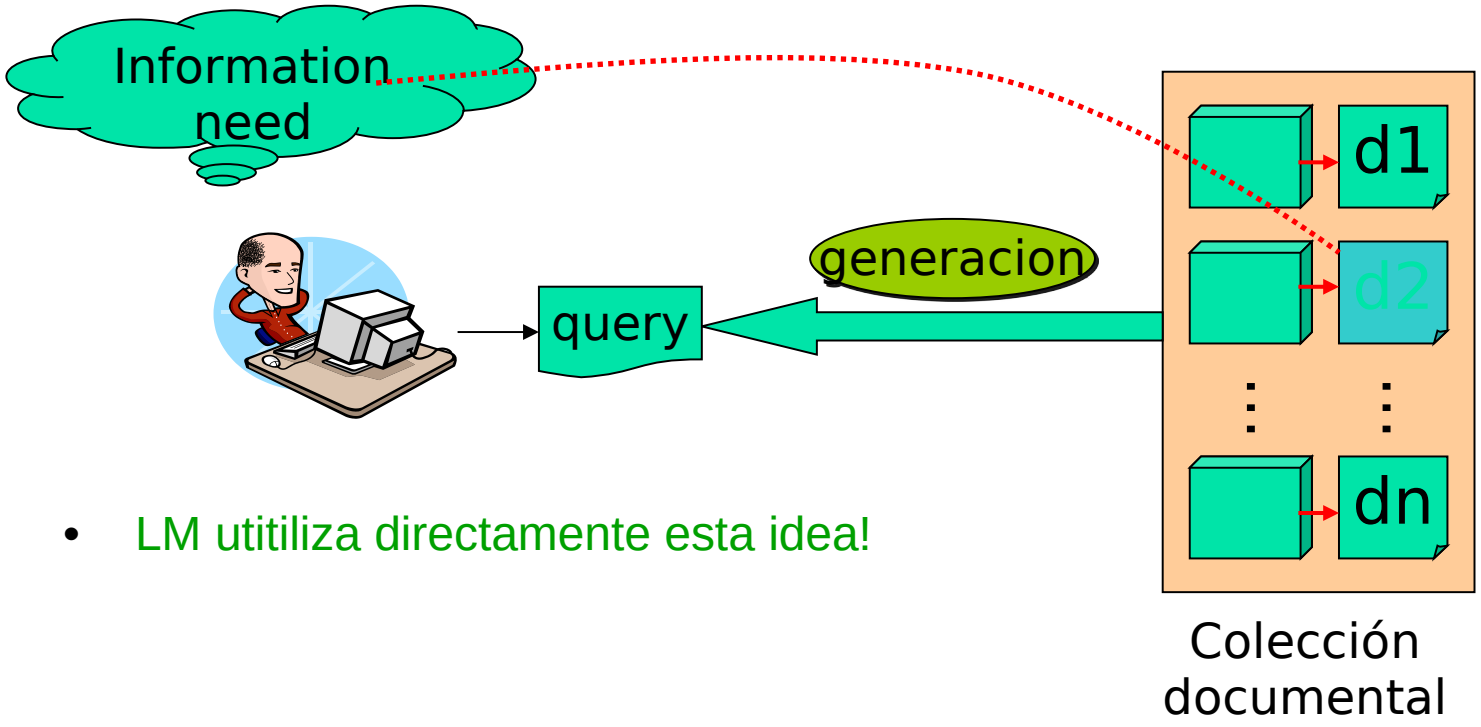
60 televisión 0.00036193

P(“coste político cierre televisión”| canal9) = 0,0107*0,0107*0,0107*0,0160= $2E^{-8}$

P(“coste político cierre televisión”| catal.) = 0,0010*0,0080*0,0080*0,0003=2E-12

RI basada en LM

- IDEA: Un usuario tienen una ida razonable sobre los términos que esperan que aparezcan en los documentos de interés.
  - Seleccionarán términos para la consulta que permitan distinguir estos documentos de otros en la colección.



- LM utitiliza directamente esta idea!

Modelado del lenguaje en RI

Un modelo de recuperación basado en Modelado del Lenguaje se compone de:

- Un conjunto de modelos de lenguaje, uno por cada documento de la colección.
- Una distribución de probabilidad que permite estimar la verosimilitud de que el Modelo i-ésimo generase cada uno de los términos de la consulta, $P(M_d|q)$
  - Considera la generación de consultas como un proceso aleatorio
- Una función de ranking que combina las distintas probabilidades para obtener un ranking de documentos

Uso de Modelos del lenguaje en RI

- Considera cada documento como el origen del modelo (ej., unigram estadísticos suficientes)
- Ordenar los documentos considerando P(M_d | q)

$$P(M_d|Q) = \frac{P(Q|M_d) \times P(M_d)}{P(Q)} \simeq P(Q|M_d) \times P(M_d) = P(M_d, Q)$$

donde

- P(Q) es constante, se ignora
- P(M_d) [a priori] es usualmente considerado constante, para todo d
  - Pero se podrían considerar criterios como autoridad, longitud, género, etc.
- P(q | Md) es la probabilidad de q dado el modelo d

Aproximación muy general

Probabilidad de generación de consulta

- Formula para ordenar (Ranking)

$$p(Q,d) = p(d)\,p(Q \mid d) \approx p(d)\,p(Q \mid M_d)$$

- La probabilidad de producir la consulta dado el modelo lingüístico del documento d utilizando MLE es:

$$\hat{p}(Q \mid M_d) = \prod_{t \in Q} \hat{p}_{ml}(t \mid M_d) = \prod_{t \in Q} \frac{tf_{(t,d)}}{dl_d}$$

Hipótesis Unigram:
Dado un LM concreto, los terminos de la consulta ocurren de forma independiente

p(t|M): LM para el documento d

tf : frec del término t en el documento d

dl : Número total de terminos en el doc d

Q = "desert" "people"

Docs:

- d1: "This is the desert. There are no people in the desert. The Earth is large."
- d2: "'Where are the people?' resumed the little prince at last. 'It's a little lonely in the desert…' ,' It is lonely when you're among people, too,' said the snake."
- d3: " 'What makes the desert beautiful,' said the little prince, 'is that somewhere it hides a well' "

$$p(Q|d1) = \frac{2}{15} \times \frac{1}{15} = 0.0089$$

$$p(Q|d2) = \frac{1}{28} \times \frac{2}{28} = 0.00255$$

$$p(Q|d3) = \frac{1}{16} \times \frac{0}{16} = 0$$

Problema: datos insuficientes

- Probabilidad cero (probabilidades nulas son catastróficas)
  - No deberíamos asignar una probabilidad cero para un documento que no incluye uno o más términos de la consulta [proporciona semántica conjuntiva]

    Que un término no ocurra en el corpus no significa que no pudiese ocurrir en un futuro

  - Necesitamos suavizar las probabilidades
    - Mecanismo de descuento sobre probabilidades con valor distinto de cero
    - Asignar masa de probabilidad a los elemenots no vistos
  - Hay un amplio espacio para mecanismos que suavicen las distribuciones de probabilidad, como añadir 1, ½ o ε a los contadores, a priori de Dirichlet, descontar, interpolar, etc.

Datos insuficientes

Una idea simple que funciona bien en términos prácticos es utilizar una mezcla entre las distribuciones multinomiales asociadas al documento y la colección…

- Un termino que no aparezca es posible, pero no debería ser más probable de lo que se podría esperar en la colección completa.
- Si tf=0,

$$p(t \mid M_d) = \frac{cf_t}{cs}$$

cf : frecuencia del termino t en la colección c

cs : tamaño de la colección (número total de tokens)

Modelos mixtos

- Utilizan un background (estadísticas de la colección o el idioma) para mejorar la estimación de las probabilidades para los términos que no están en un documento.
  - Mejora el rendimiento de los sistemas (precisión) y evita el problema de obtener probabilidad cero para palabras no vistas.

- Construyen un modelo utilizando tanto estadístico del documento como la colección:

  Encontramos dos modelos
  - Jelinek Mercer smoothing
  - Dirichlet smoothing

Modelos mixtos
Jelinek Mercer smoothing

La ecuación representa la probabilidad de que el documento que el usuario tiene en mente sea en realidad el que se está considerando.

$$P(t|doc) = (1 - \lambda)P_{mle}(t|M_{corpus}) + \lambda P_{mle}(t|M_{doc})$$

- Mezcla la probabiliad que se estima del documento con la frecuencia general del término en la colección.
- Determinar de forma correcta λ es muy importante, λ está en $[0,1]$
  - Un valor alto de lambda hace que la búsqueda sea “del tipo conjuntiva” - útil para consultas pequeñas
  - Un valor bajo es bueno para consultas de tamaño largo

Modelo mixto básico

- Formulación general del LM para IR

$$p(Q,d) = p(d) \prod_t ((1-\lambda)p(t|M_C) + \lambda p(t|M_d))$$

Modelo linguístico general

Modelo individual del documento

Ejemplo

- Colección documental (2 documentos)
  - d_1: Xerox reports a profit but revenue is down
  - d_2: Lucent narrows quarter loss but revenue decreases further
- Modelo: MLE unigram de los documentos; $\lambda = ½$
- Consulta: *revenue down*
  - $P(Q|d_1) = [(1/8 + 2/16)/2] \times [(1/8 + 1/16)/2]$
    $= 1/8 \times 3/32 = 3/256$
  - $P(Q|d_2) = [(1/8 + 2/16)/2] \times [(0 + 1/16)/2]$
    $= 1/8 \times 1/32 = 1/256$
- Ranking: $d_1 > d_2$

Cálculo práctico del score: Se toman logaritmos

Según hemos visto

$$\log p(Q,d) = \log p(d) + \sum \log((1-\lambda)p(t|M_C) + \lambda p(t|doc))$$

Se toman logaritmos para evitar problemas de precisión al multiplicar números muy pequeños ….

0.004x0.00006x 0.2 x .....

Fórmula de Jelinek-Mercer

Según hemos visto

$$\log p(Q,d) = \log p(d) + \log p(Q|d) = \log p(d) + \sum \log((1-\lambda)\, p(t|M_c) + \lambda\, p(t|d))$$

Que tras algunas aproximaciones, llegamos a la fórmula de Jelinek-Mercer

$$P(q|d) = \sum_{t \in q} \log(1 + \frac{(\lambda)\frac{tf_{t,d}}{L_d}}{(1-\lambda)\frac{tf_t+1}{L_c+1}})$$

Ld y Lc representan el número total de términos en el doc y colección, resp.

Se toman logaritmos para evitar problemas de precisión al
Multiplicar números muy pequeños ....
0.004x0.00006x 0.2 x .....

$$P(q|d) = \sum_{t \in q} \log(1 + \frac{(\lambda)\frac{tf_{t,d}}{L_d}}{(1-\lambda)\frac{tf_t+1}{L_c+1}})$$

Q = "desert" "people"

Docs:

- d1: "This is the desert. There are no people in the desert. The Earth is large."
- d2: "'Where are the people?' resumed the little prince at last. 'It's a little lonely in the desert…' ,' It is lonely when you're among people, too,' said the snake."
- d3: " 'What makes the desert beautiful,' said the little prince, 'is that somewhere it hides a well' "

$\lambda = 0.9; L_c = 59$

$$P(q|M_{d1}) = \log(1 + \frac{0.9 * \frac{2}{15}}{0.1 * \frac{5}{60}}) + \log(1 + \frac{0.9 * \frac{1}{15}}{0.1 * \frac{4}{60}}) = 2.7343674 + 2.302585 = 5.036952$$

$$P(q|M_{d2}) = \log(1 + \frac{0.9 * \frac{1}{28}}{0.1 * \frac{5}{60}}) + \log(1 + \frac{0.9 * \frac{2}{28}}{0.1 * \frac{4}{60}}) = 1.5804503 + 2.364889 = 3.9453392$$

$$P(q|M_{d3}) = \log(1 + \frac{0.9 * \frac{1}{16}}{0.1 * \frac{5}{60}}) + \log(1 + \frac{0.9 * \frac{0}{16}}{0.1 * \frac{4}{60}}) = 2.0476928 + 0 = 2.0476928$$

Modelos Mixtos:
suavizado de Dirichlet

- La intuición tras el suavizado de Dirichlet es añadir μ términos a cada documento (μ suele tomar valores en torno a 2000) y distribuirlos según las estadísticas de la colección.

$$P(t|d) = \frac{tf_{t,d} + \mu P(t|M_C)}{L_d + \mu}$$

	p(t|Col)	mu=100
t1	0,1	10
t2	0,2	20
t3	0,3	30
t4	0,4	40

	Doc	p(t|doc)=tf/L_d	p'(t|d)
t1	150	0,375	0,32
t2	0	0,000	0,04
t3	120	0,300	0,3
t4	130	0,325	0,34

Modelos Mixtos: suavizado de Dirichlet

- Podemos finalmente calcular el valor

$$p(Q|d) = \sum_{t \in Q} log(\frac{tf_{t,d} + \mu p(t|M_C)}{L_d + \mu}$$

$$p(Q|d) = \sum_{t \in Q} \log\left(\frac{tf_{t,d} + \mu p(t|M_C)}{L_d + \mu}\right)$$

Q = "desert" "people"

Docs:

- d1: "This is the desert. There are no people in the desert. The Earth is large."
- d2: "'Where are the people?' resumed the little prince at last. 'It's a little lonely in the desert…' ,' It is lonely when you're among people, too,' said the snake."
- d3: " 'What makes the desert beautiful,' said the little prince, 'is that somewhere it hides a well' "

$$\psi = 10 ; L_C = 59$$

$$P(q|M_{d1}) = \log\left(\frac{2+10\times\frac{4}{59}}{15+10}\right) + \log\left(\frac{1+10\times\frac{3}{59}}{15+10}\right) = \log(0{,}1071) + \log(0{,}0603) = -2{,}1895$$

$$P(q|M_{d2}) = \log\left(\frac{1+10\times\frac{4}{59}}{28+10}\right) + \log\left(\frac{2+10\times\frac{3}{59}}{28+10}\right) = \log(0{,}0441) + \log(0{,}0660) = -2{,}5353$$

$$P(q|M_{d3}) = \log\left(\frac{1+10\times\frac{4}{59}}{16+10}\right) + \log\left(\frac{0+10\times\frac{3}{59}}{16+10}\right) = \log(0{,}0645) + \log(0{,}0195) = -2{,}8988$$

LM vs. BM25

- La diferencia principal es si la “Relevancia” aparece de forma explícita en el modelo o no
  - LM no modelizan la relevancia
- LM asume que los documentos y las consultas son del mismo tipo
- Tratable computationalmente, intuitivamente atractivo

LM vs Espacio Vectorial

- Frecuencia del término está de forma explícita en el modelo
- El uso de probabilidades normaliza las frecuencias de los términos
- El efecto de realizar una combinación con la colección completa se puede considerar como un idf:
  - términos que son raros en la colección pero comunes en algunos documentos tendrán una mayor influencia en el ranking

Modelos probabilísticos vs Espacio Vectorial

- Similitudes
  - Los pesos de los términos están basados en frecuencias
  - Términos se utilizan de forma independiente
  - Frecuencia documental Inversa (idf) es utilizado
  - Utiliza normalización sobre la longitud es útil
- Diferencias
  - Basado en probabilidades en lugar de similaridades
    - Intuición es probabilística y no geométrica (coseno)
  - Mecanismos para utilizar longitud, frecuencia, etc. son diferentes