Modelos de Recuperação de Informação

Prof. Luciano Barbosa

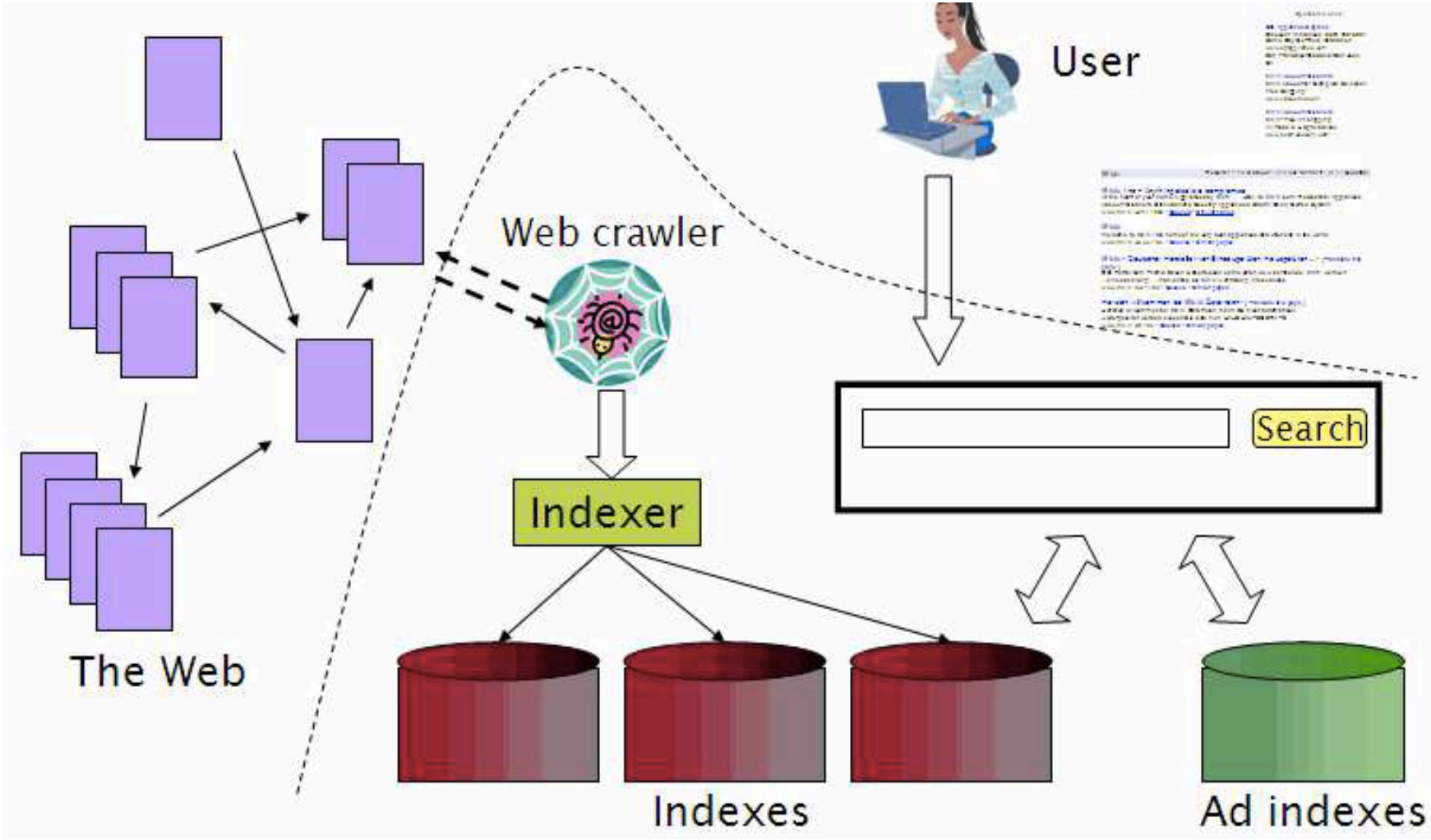
(Parte do material retirado dos slides dos livros adotados)

Centro de Informática UFPE

Universidade Federal de Pernambuco

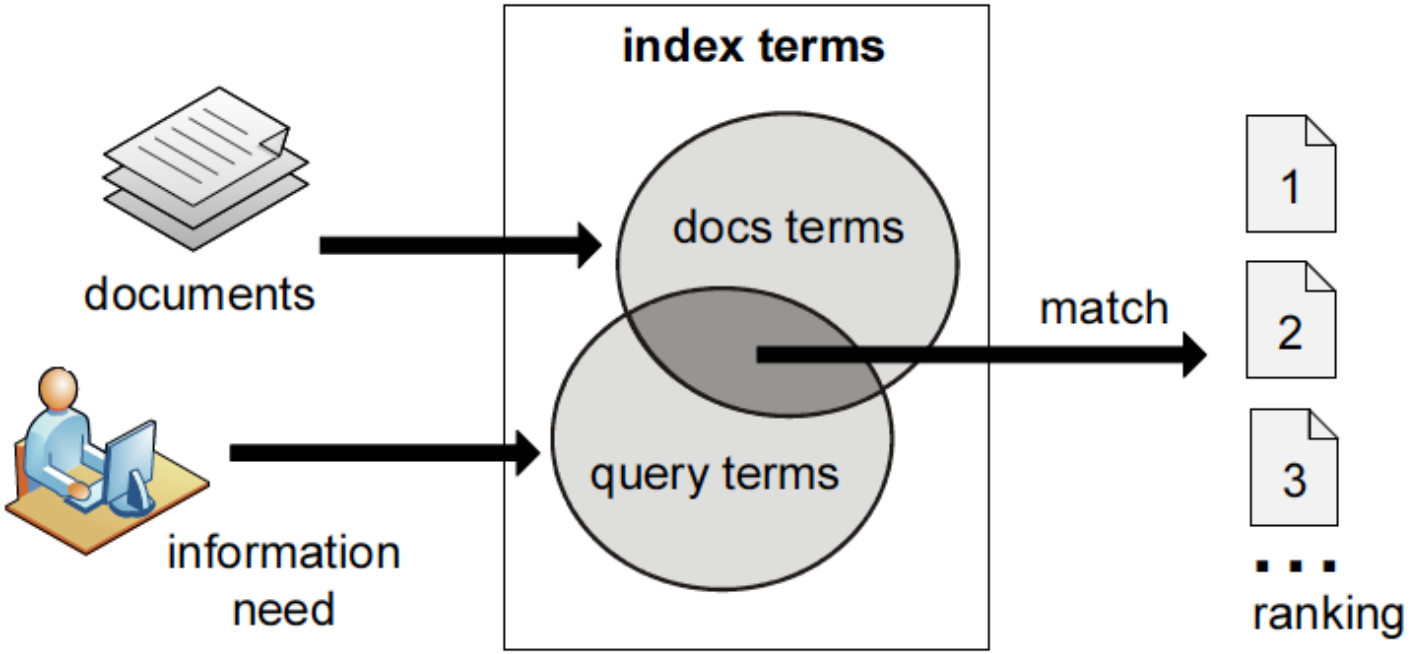
CIn.ufpe.br

Visão Geral de um Engenho de Busca



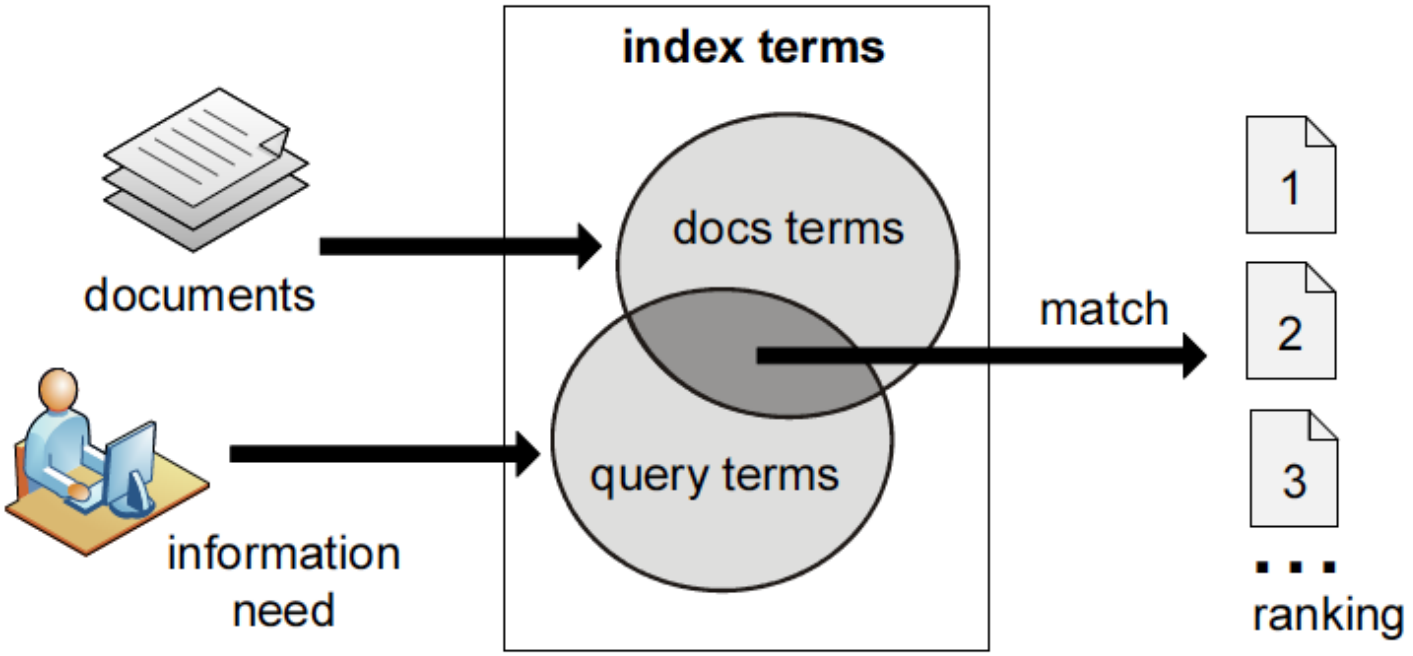
Objetivo de Recuperação de Informação

- Satisfazer necessidade de informação de usuário (consulta) por buscar nos dados (docs, multimídia etc)



Modelos de RI

- De forma concreta: função de similaridade entre consulta e documentos



Modelo Booleano

- Baseado na teoria de conjuntos e álgebra booleana
- Consultas: expressões booleanas
  - Ex.: q = casa e branca, q = camisa ou blusa
- Resultado: todos documentos que satisfazem a consulta
- Documento representado por um conjunto de palavras
- Considera apenas a existência do termo ou não

Exemplo

- Consulta: Brutus AND Cesar AND NOT Calpurnia
- Documentos:

	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	
…							

Modelo Booleano

- Vantagens
  - Fácil compreensão
  - Formalismo claro (álgebra booleana)
- Desvantagens
  - Muito rígido (não permite casamentos parciais)
  - Não é possível ranquear
  - Maioria dos usuários não usam

Peso dos Termos

- Termos em um documento não são igualmente úteis para descrever seu conteúdo
  - Ex: palavras frequentes no documento -> importantes
  - Ex: palavras que aparecem em todos documentos da coleção -> não importantes
- Peso usado para caracterizar a importância do termo
- Útil para computar ranqueamento de documentos dada uma consulta
  - Documentos com termos da consulta com alto peso são melhores ranqueados

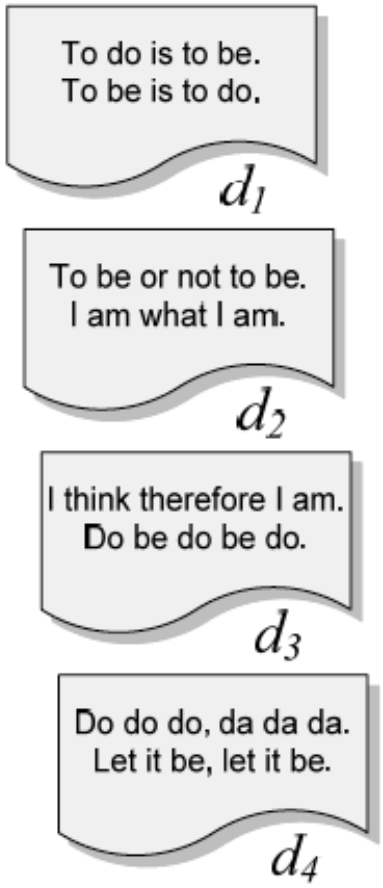
Frequência do Termo no Documento - TF

- Intuição: a importância do termo em um documento é proporcional à sua frequência nele

	tf weight
binary	{0,1}
raw frequency	$f_{i,j}$
log normalization	$1 + \log f_{i,j}$
double normalization 0.5	$0.5 + 0.5 \frac{f_{i,j}}{max_i f_{i,j}}$
double normalization K	$K + (1 - K) \frac{f_{i,j}}{max_i f_{i,j}}$

Frequência do Termo

- Usando a variação de tf com log



Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

Inverse Document Frequency (IDF)

- Medir a espcifidade de um termo
- Não mede a especificidade semântica de um termo
  - Depende do seu significado
  - Pode ser usado um theasurus: wordnet
  - Ex: o termo bebida é mais genérico que café ou chá
- Em RI, especifidade estatística ao invés da semântica
  - O inverso do número de documentos nos quais o termo ocorre

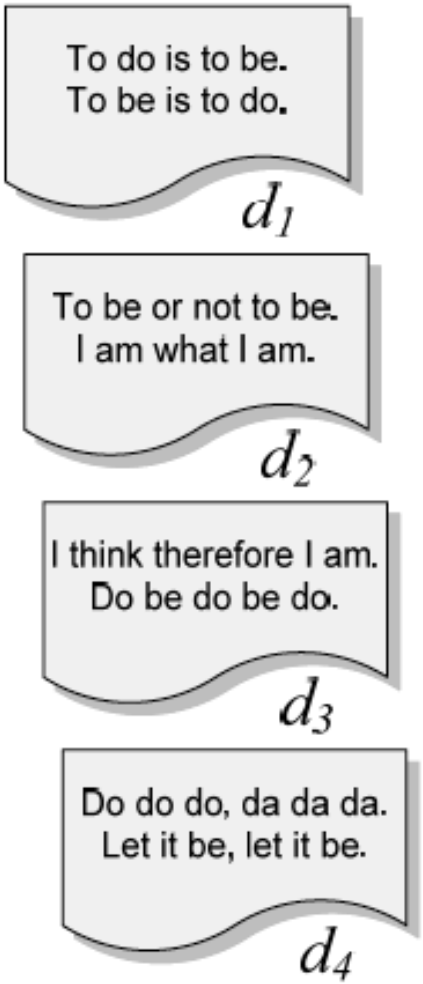
$$idf_i = \log \frac{N}{n_i}$$

- Usado amplamente em algoritmos de ranqueamento

Inverse Document Frequency (IDF): Variações

	idf weight
unary	1
inverse frequency	$\log \frac{N}{n_i}$
inv frequency smooth	$\log(1 + \frac{N}{n_i})$
inv frequeny max	$\log(1 + \frac{max_i n_i}{n_i})$
probabilistic inv frequency	$\log \frac{N - n_i}{n_i}$

Inverse Document Frequency (IDF)



	term	n_i	$idf_i = \log(N/n_i)$
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

TF-IDF

- Combinação do tf com o idf

$$tf_{ij} \times idf_i$$

- Variações:

weighting scheme	document term weight	query term weight
1	$f_{i,j} * \log \frac{N}{n_i}$	$(0.5 + 0.5 \frac{f_{i,q}}{max_i \ f_{i,q}}) * \log \frac{N}{n_i}$
2	$1 + \log f_{i,j}$	$\log(1 + \frac{N}{n_i})$
3	$(1 + \log f_{i,j}) * \log \frac{N}{n_i}$	$(1 + \log f_{i,q}) * \log \frac{N}{n_i}$

To do is to be.
To be is to do.
d_1

To be or not to be.
I am what I am.
d_2

I think therefore I am.
Do be do be do.
d_3

Do do do, da da da.
Let it be, let it be.
d_4

		d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

TF-IDF

	term	n_i	$idf_i = \log(N/n_i)$
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

IDF

Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

TF

To do is to be.
To be is to do.
d_1

To be or not to be.
I am what I am.
d_2

I think therefore I am.
Do be do be do.
d_3

Do do do, da da da.
Let it be, let it be.
d_4

		d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

TF-IDF

	term	n_i	$idf_i = \log(N/n_i)$
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

IDF

Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

TF

To do is to be.
To be is to do.
d_1

To be or not to be.
I am what I am.
d_2

I think therefore I am.
Do be do be do.
d_3

Do do do, da da da.
Let it be, let it be.
d_4

		d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

TF-IDF

	term	n_i	$idf_i = \log(N/n_i)$
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

IDF

Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

TF

Distância e Similaridade

- Para vários problemas temos que medir quanto dois objetos estão próximos
- Exemplos:
  - Recomendação
  - Busca na Web
  - Páginas duplicadas

Distância

- Medida numérica de quanto dois objetos são diferentes
- Propriedades desejadas:
  1. d(p,p) = 0 (distância mínima)
  2. d(p,q) = d(q,p) para todo p e q (simetria)
- Tipos para vetores de números reais

$$L_p(x,y) = [|x_1 - y_1|^p + \cdots + |x_d - y_d|^p]^{1/p}$$ **Minkowski**

$$L_2(x,y) = \sqrt{|x_1 - y_1|^2 + \cdots + |x_d - y_d|^2}$$ **Euclidiana**

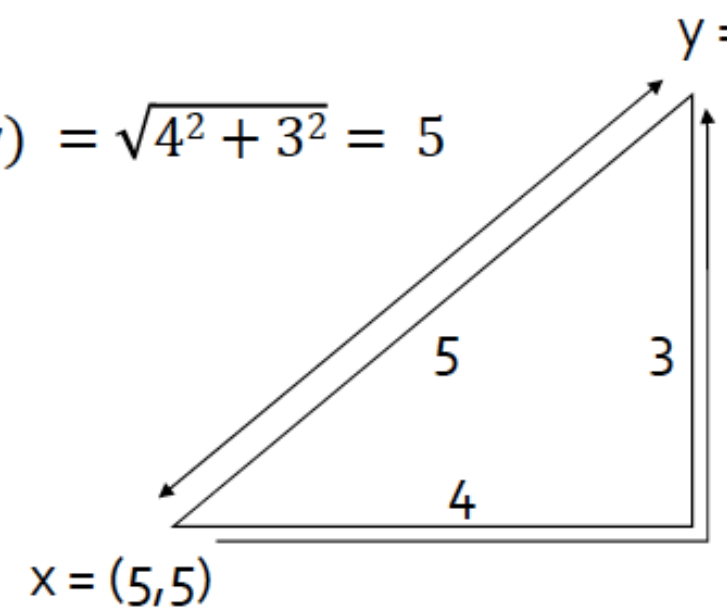
$$L_1(x,y) = |x_1 - y_1| + \cdots + |x_d - y_d|$$ **Manhattan**

$$L_\infty(x,y) = \max\{|x_1 - y_1|, \ldots, |x_d - y_d|\}$$ **Chebyshev**

Exemplo de Distâncias

L_2-norm:

$$dist(x,y) = \sqrt{4^2 + 3^2} = 5$$



L_1-norm:

$$dist(x,y) = 4 + 3 = 7$$

L_∞-norm:

$$dist(x,y) = \max\{3,4\} = 4$$

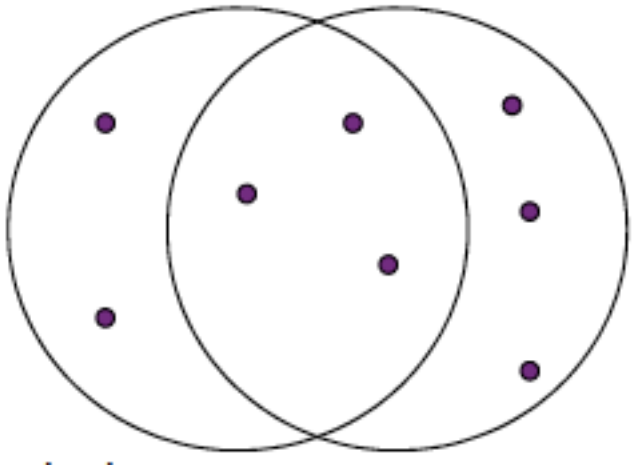
Similaridade

- Medida numérica de quanto dois objetos são parecidos
  - Função que mapeia dois objetos para um número real
  - Usualmente em intervalos [0,1] ou [-1,1]
- Propriedades desejadas:
  1. s(p,p) = 1 (similaridade máxima)
  2. s(p,q) = s(q,p) para todo p e q (simetria)

Similaridade de Jaccard

- Definição: tamanho da intersecção dividido pela união

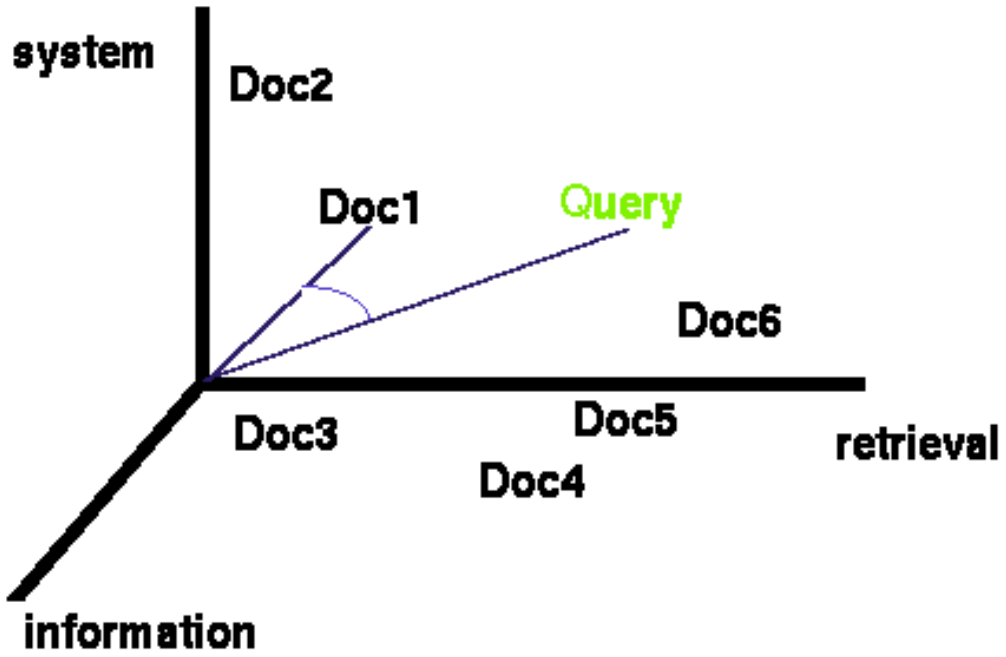
$$JSim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|.$$



- Exemplo: $Jsim(C_1,C_2) = 3/8$

Modelo de Espaço de Vetores

- Documento e consulta representados por um vetor de palavras
- Cada palavra é uma dimensão do vetor



Modelo de Espaço de Vetores

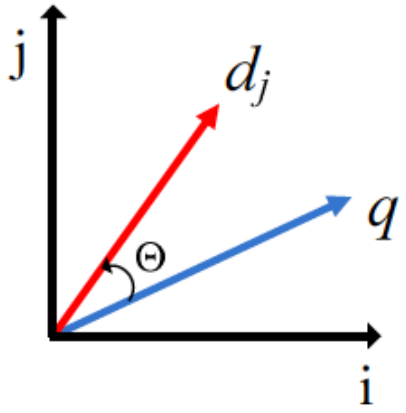
- Espaço é do tamanho do vocabulário (alta dimensão)
- Documentos são vetores esparsos
- Similaridade entre os vetores da consulta e dos documentos
  - Tamanho da intersecção
  - Jaccard:

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

  - Cosseno

Similaridade de Cosseno

- Documentos ranqueados pela proximidade de pontos representando a consulta e os documentos



$$\vec{d_j} = (w_{1j}, w_{2j}, \ldots, w_{tj})$$
$$\vec{q} = (w_{1q}, w_{2q}, \ldots, w_{tq})$$

$$cos(\theta) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|}$$

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}$$

Cálculo da Similaridade

- Considere dois documentos D_1 e D_2 e uma consulta Q
  - D_1 = (0.5, 0.8, 0.3), D_2 = (0.9, 0.4, 0.2), Q = (1.5, 1.0, 0)

$$Cosine(D_1, Q) = \frac{(0.5 \times 1.5) + (0.8 \times 1.0)}{\sqrt{(0.5^2 + 0.8^2 + 0.3^2)(1.5^2 + 1.0^2)}}$$

$$= \frac{1.55}{\sqrt{(0.98 \times 3.25)}} = 0.87$$

$$Cosine(D_2, Q) = \frac{(0.9 \times 1.5) + (0.4 \times 1.0)}{\sqrt{(0.9^2 + 0.4^2 + 0.2^2)(1.5^2 + 1.0^2)}}$$

$$= \frac{1.75}{\sqrt{(1.01 \times 3.25)}} = 0.97$$

Exemplo: Cálculo usando TF-IDF

- Consulta: to do

doc	rank computation	rank
d_1	$\frac{1*3+0.415*0.830}{5.068}$	0.660
d_2	$\frac{1*2+0.415*0}{4.899}$	0.408
d_3	$\frac{1*0+0.415*1.073}{3.762}$	0.118
d_4	$\frac{1*0+0.415*1.073}{7.738}$	0.058

Modelo de Espaço de Vetores

- Vantagens:
  - Eficiente
  - Permite casamento parcial
  - Fácil de implementar
  - Funciona bem na prática
- Cons:
  - Assume independência dos termos
  - Sem informação semântica e sintática

BM25

- Um dos mais populares e efetivos algoritmos de ranqueamento
- Baseado no BIM
- 3 princípios básicos: tf, idf e normalização pelo tamanho do documento
- Criado como resultado de experimentos em variações de modelos probabilísticos
- Usado como baseline em experimentos de RI

BM25

$$\sum_{i \in Q} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \cdot \frac{(k_1+1) f_i}{K + f_i} \cdot \frac{(k_2+1) q f_i}{k_2 + q f_i}$$

BIM

Normalização pelo tamanho do documento

TF

- k1, k2 e b são parâmetros definidos empiricamente (depende da coleção)

$$K = k_1((1 - b) + b \cdot \frac{dl}{avdl})$$

- dl: tamanho do documento

BM25: Exemplo

- Query with two terms, “president lincoln”, (*qf = 1)*
- No relevance information (*r and R are* zero)
- N = 500,000 documents
- *“president”* occurs in 40,000 documents (n_1 = 40, 000)
- *“lincoln”* occurs in 300 documents (n_2 = 300)
- “president” occurs 15 times in doc (f_1 = 15)
- *“lincoln”* occurs 25 times (f_2 = 25)
- document length is 90% of the average length (*dl/avdl* = .9)
- k_1 = 1.2, b = 0.75, and k_2 = 100
- $K = 1.2 \cdot (0.25 + 0.75 \cdot 0.9) = 1.11$

BM25: Exemplo

$$\sum_{i \in Q} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \cdot \frac{(k_1+1) f_i}{K + f_i} \cdot \frac{(k_2+1) q f_i}{k_2 + q f_i}$$

$$\begin{aligned} BM25(Q, D) \quad = \quad & \log \frac{(0 + 0.5)/(0 - 0 + 0.5)}{(40000 - 0 + 0.5)/(500000 - 40000 - 0 + 0 + 0.5)} \\ & \times \frac{(1.2 + 1)15}{1.11 + 15} \times \frac{(100 + 1)1}{100 + 1} \\ & + \log \frac{(0 + 0.5)/(0 - 0 + 0.5)}{(300 - 0 + 0.5)/(500000 - 300 - 0 + 0 + 0.5)} \\ & \times \frac{(1.2 + 1)25}{1.11 + 25} \times \frac{(100 + 1)1}{100 + 1} \end{aligned}$$

$$\begin{aligned} = \quad & \log 460000.5/40000.5 \cdot 33/16.11 \cdot 101/101 \\ & + \log 499700.5/300.5 \cdot 55/26.11 \cdot 101/101 \\ = \quad & 2.44 \cdot 2.05 \cdot 1 + 7.42 \cdot 2.11 \cdot 1 \\ = \quad & 5.00 + 15.66 = 20.66 \end{aligned}$$

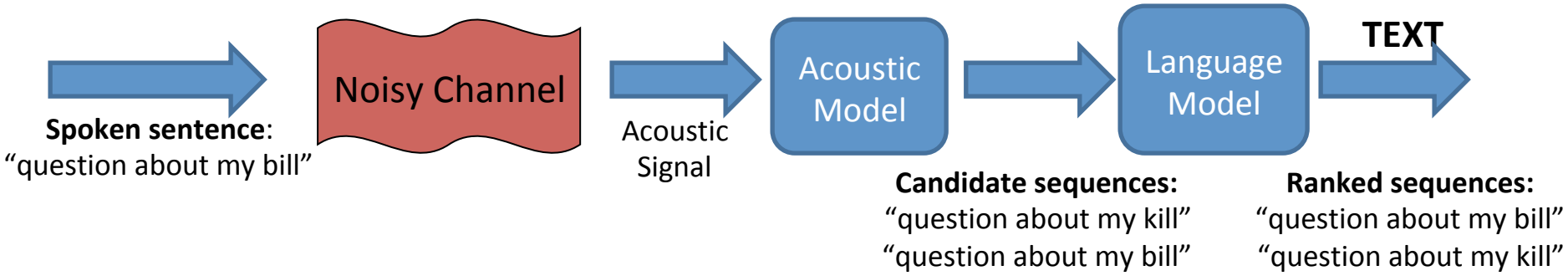
BM25: Exemplo

- Efeito da frequência dos termos

Frequency of "president"	Frequency of "lincoln"	BM25 score
15	25	20.66
15	1	12.74
15	0	5.00
1	25	18.2
0	25	15.66

Modelo de Linguagem

- Usado em várias tarefas de PLN: reconhecimento de fala, tradução automática de texto e reconhecimento de escrita
- Reconhece se um dado n-grama é bem formado



- Probabilidade do n-grama ter sido criado pelo modelo de linguagem
- Modelo de linguagem criado a partir de um corpus de dados

Modelo de Linguagem em RI

- Corpus do modelo de linguagem: documento
- N-gram: consulta
- Dada uma consulta qual a probabilidade dela ter sido criada pelo documento: P(d|q)
- Vários tipos de modelo de linguagem
  - Unigramas
  - Bigramas
  - Alta ordem
- Suposição de independência (unigramas): usados extensivamente em RI

Modelo de Linguagem em RI

- Ranquear docs baseado em:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}$$

Probabilidade a priori (Importância independente de termo)

Mesmo para todos documentos da consulta

- A priori uniform:

$$P(q|d) \approx P(d|q)$$

Modelo de Linguagem em RI

- Calculando:

$$P(q|M_d) = \prod_{\text{distinct term } t \text{ in } q} P(t|M_d)^{\text{tf}_{t,q}}$$

$$\hat{P}(t|M_d) = \frac{\text{tf}_{t,d}}{|d|}$$

freq do termo t no doc d

- Problema se as palavras na consulta não aparecem no documento -> probabilidade igual a 0

Exemplo

language model of d_1

w	$P(w\|.)$	w	$P(w\|.)$
STOP	.2	toad	.01
the	.2	said	.03
a	.1	likes	.02
frog	.01	that	.04
		...	...

language model of d_2

w	$P(w\|.)$	w	$P(w\|.)$
STOP	.2	toad	.02
the	.15	said	.03
a	.08	likes	.02
frog	.01	that	.05
		...	...

query: frog said that toad likes frog STOP

$$P(\text{query}|M_{d1}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.2$$
$$= 0.000000000048 = 4.8 \cdot 10^{-12}$$

$$P(\text{query}|M_{d2}) = 0.01 \cdot 0.03 \cdot 0.05 \cdot 0.02 \cdot 0.02 \cdot 0.01 \cdot 0.2$$
$$= 0.000000000120 = 12 \cdot 10^{-12}$$

$$P(\text{query}|M_{d1}) < P(\text{query}|M_{d2})$$

Smoothing

- Estima a probabilidade de um termo ausente
- Diminui (desconta) a probabilidade para palavras que aparecem no documento
- Usa a probabilidade que sobra para estimar palavras não vistas no documento

Jelinek-Mercer Smoothing

$$P(t|d) = \lambda P(t|M_d) + (1 - \lambda)P(t|M_c)$$

documento

coleção (idf)

- Parâmetro:
  - Altos valores:
    - Mais ênfase no tf
    - Melhor para consultas menores
  - Baixos valores:
    - Mais ênfase no idf
    - Melhor para consultas maiores

Exemplo Usando J-M

$$P(t|d) = \lambda P(t|M_d) + (1-\lambda)P(t|M_c)$$

Collection: d_1 and d_2

d_1: Jackson was one of the most talented entertainers of all time

d_2: Michael Jackson anointed himself King of Pop

Query q: Michael Jackson

Use mixture model with $\lambda = 1/2$

$P(q|d_1) = [(0/11 + 1/18)/2] \cdot [(1/11 + 2/18)/2] \approx 0.003$

$P(q|d_2) = [(1/7 + 1/18)/2] \cdot [(1/7 + 2/18)/2] \approx 0.013$

Ranking: $d_2 > d_1$

Michael Jackson

Dirichlet Smoothing

$$\hat{P}(t|d) = \frac{\mathrm{tf}_{t,d} + \alpha \hat{P}(t|M_c)}{L_d + \alpha}$$

prob a priori na coleção (idf)

tamanho do doc

efeito da probilidade a priori

- Ambos precisam de refinamento

Chamada

Latent Semantic Indexing

- RI tradicional não leva em consideração termos relacionados
- Necessidade de informação do usuário é mais relacionada com conceitos do que termos
- Documentos que compartilham conceitos com outros relevantes devem ser considerados

Latent Semantic Indexing

- Objetivo: mapear documentos e consultas para um espaço dimensional de conceitos
- Redução de dimensionalidade: singular value decomposition (SVD)
- $C = U\Sigma V^T$

  C: matriz termo-documento
- Computar um novo C' "reduzido" para obter melhores similaridades
- Chamado de Latent Semantic Indexing

Exemplo de $C = U\Sigma V^T$: Matriz C

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

Exemplo de $C = U\Sigma V^T$: Matriz U

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

- Captura diferentes tópicos: dimensões semânticas dos termos
- Peso: relacionamento do termo i com a dimensão
- Dimensão 2: terra/água

Exemplo de $C = U\Sigma V^T$: Matriz Σ

- Valores singulares de C
- Magnitude do valor mede a importância da dimensão semântica

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

Exemplo de $C = U\Sigma V^T$: Matriz V^T

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

- Captura dimensões semânticas dos documentos
- Peso: quanto o documento i é relacionado com a dimensão j

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

$=$

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

$\times$

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

$\times$

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

Como Usar?

- Cada valor singular representa a importância de uma dimensão
- Zerar dimensões menos importantes para:
  - Reduzir o ruído
  - Tornar documentos não similares no espaço original em similares no novo espaço

Redução em 2 Dimensões

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

zera valores singulares

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

=

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

×

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

×

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

Resultado

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

Similaridade de d_2 e d_3 = 0

Similaridade de d_2 e d_3 = 0.52

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

Latent Semantic Indexing

- Representa documentos em um espaço reduzido
- Documentos não são similares no espaço original podem se tornar similares
- Ataca o problema de sinônimos e relacionamento semântico de palavras
- Sinônimos contribuem para a similaridade

Usando LSI para Consultas

- Computa SVD para a matrix termo-documento da coleção
- Reduz o espaço e computa as representações dos documentos reduzidas
- Projeta a consulta para o espaço reduzido: $\vec{q}_k = \Sigma_k^{-1} U_k^T \vec{q}.$
- Computa similaridade q_k com V_k