Termos Ponderados e Modelo Vetorial

Prof. Dr. Leandro Balby Marinho
http//www.dsc.ufcg.edu.br/~lbmarinho

UFCG CEEI Departamento de Sistemas e Computação

Sistemas de Recuperação da Informação

(Slides Adaptados de Cristopher D. Manning)

Roteiro

- Recuperação baseada em Ranking
- Atribuição de pesos a documentos
- Frequência de termos
- Esquemas de pesos
- Modelo Vetorial de RI
- Pesos em espaço vetorial

Recuperação Baseada em Ranking

- Até agora lidamos com consultas Booleanas
  - Documentos casam ou não casam.
- Boa para usuários especialistas com um entendimento preciso das suas necessidades e da coleção
  - Boa para aplicações: aplicações podem facilmente processar milhares de resultados.
- Ruim para a maioria dos usuários
  - Incapazes de escrever consultas Booleanas (ou são, mas acham muito trabalhoso)
  - Não querem procurar em milhares de resultados
    - Principalmente quando se trata de busca na Web

Problemas da Busca Booleana

- Consultas Booleanas resultam ou em poucos (=0) ou em muitos (milhares) resultados
- Precisa-se de muita habilidade para produzir consultas que gerem um número razoável de resultados.
  - AND muito poucos; OR demais

Recuperação Baseada em Ranking

- Em modelos de RI baseados em ranking, o sistema retorna uma ordenação dos documentos na coleção em relação a uma consulta.

- Consultas em texto livre: Em vez de uma linguagem de consulta com operadores e expressões, a consulta é apenas uma ou duas palavras em linguagem natural.

Recuperação Baseada em Ranking

- O número de resultados não é problema
  - Apenas os *k* (≈ 10) resultados são mostrados
  - O usuário não é sobrecarregado

  - Premissa: O algoritmo de ranking funciona

Atribuição de pesos

- Queremos retornar, em ordem de relevância, os documentos mais prováveis de satisfazer uma consulta.
- Como podemos ordenar (ranquear) os documentos em uma coleção de acordo com uma consulta?
- Atribuindo um peso – digamos em [0, 1] – para cada documento.
- Esse peso mensura quão bem o documento casa com a consulta.

Atribuição de pesos

- Precisamos de uma forma de atribuir um peso para um par consulta/documento.
- Vamos começar com consultas de um-termo.
- Se o termo de consulta não ocorre no documento: peso deve ser 0.
- Quanto mais frequente o termo de consulta no documento, maior o peso.

Coeficiente Jaccard

- Uma forma comum de medir a interseção entre os conjuntos A e B.

- $jaccard(A,B) = |A \cap B| / |A \cup B|$
- $jaccard(A,A) = 1$
- $jaccard(A,B) = 0$ *se* $A \cap B = 0$

- A e B não precisam ser do mesmo tamanho.

- Sempre atribui um número entre 0 e 1.

Atribuindo Pesos com Coeficiente de Jaccard

- Qual o peso que o coeficiente de Jaccard calcula para cada par consulta-documento abaixo?

- Consulta: idos *de março*
- Documento 1: *césar morreu em março*
- Documento 2: *águas de março*

Problemas com o coeficiente Jaccard

- Não considera *frequência dos termos.*
- Termos raros em uma coleção são mais informativos que termos frequentes.
- Precisamos de formas mais sofisticadas de normalizar o tamanho dos documentos.

Relembre a Matriz de Incidência Binária

	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

Cada documento é representado por um vetor binário $\in \{0,1\}^{|V|}$

Matriz de Frequencia termo-documento

- Considere o número de ocorrências de um termo em um documento:
  - Cada documento é um vetor de nr. de ocorrências de termos

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

O modelo “Bag of Words”

- Representação de vetores não considera a ordem das palavras em um documento.
- *João é mais alto que Maria e Maria é mais alta que João* possuem os mesmos vetores.
- Esse modelo é chamado de “bag of words”.

Frequencia de termo *tf*

- A freq. do termo *t* no documento *d*, $tf_{t,d,}$ é definida como o número de vezes que *t* ocorre em *d*.
- Podemos usar *tf* para calcular pesos para pares consulta-documento.
- Mas usar *tf* puro não é uma boa idéia:
  - Um doc. com 10 ocorrências de um termo é mais relevante que um documento com 1 ocorrência do termo.
  - Mas não 10 vezes mais relevante.
- Relevância não aumenta proporcionalmente com *tf*.
- Use o logaritmo de tf.

Peso Log-frequência

- O peso logarítmico da freq. de *t* em *d* é dado por

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{se } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{de outra forma} \end{cases}$$

- 0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, etc.
- Peso para um par consulta-documento: somatório sob os termos *t* em ambos *q* e *d*:
- Peso $= \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$
- O peso é 0 se nenhum dos termos de consulta estiver presente no documento.

Frequência de Documentos

- Termos raros são mais informativos que termos frequentes
  - Lembre das stop words
- Considere um termo na consulta que seja raro na coleção (e.g., *agorafobia*).
- Um documento contendo esse termo é muito provável de ser relevante a consulta *agorafobia.*
- É razoável atribuir maior peso para termos raros como *agorafobia*.

Frequência de Documentos

- A frequencia dos termos não é um indicador certo de relevância.
- Queremos um esquema que atribua maior peso aos termos raros em detrimento dos termos frequentes.

Peso idf

- Seja df_t a freq. de documento para *t*: o número de documentos que contém *t*
  - df_t é uma medida inversa da informatividade de *t*
  - $df_t \leq N$
- O idf (inverse document frequency) de *t* é definido por

$$idf_t = N / df_t$$

  - Normalmente log (N/df_t) é usado

Exemplo idf: N=10^8

term	df_t	idf_t
calpurnia		1
animal		100
domingo		1,000
mosca		10,000
abaixo		100,000
o		1,000,000

$$\text{idf}_t = \log_{10}(N/\text{df}_t)$$

Há um valor idf para cada termo *t* na coleção.

Freq. Coleção vs. Freq. Documento

- A frequencia de coleção de *t* é o número de ocorrências de *t* na coleção
- Exemplo:

Palavra	Frequencia da Coleção	Frequência de Documento
seguro	10440	3997
tentativa	10422	8760

- Qual palavra é um melhor termo de busca (e deveria obter um maior peso)?

Peso tf-idf

- O peso tf-idf de um termo é o produto do seu peso tf e seu peso idf.

$$w_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

- O melhor esquema de pesos conhecido da RI
  - Nomes alternativos: tf.idf, tf x idf
- Aumenta com o número de ocorrências dentro de um documento.
- Aumenta com a raridade do termo na coleção.

Peso tf-idf para um Documento

$$\text{Peso}(q,d) = \sum_{t \in q} \text{tf}_{t,d} . \text{idf}_t$$

Exercício

- Considere a tabela (a) de frequencias para os 3 documentos denotados por Doc1, Doc2, Doc3 abaixo. Calcule os pesos tf-idf para os termos “car”, “auto”, “insurance” e “best”, para cada documento, usando os valores de idf na tabela (b) abaixo.

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Tabela (a)

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Tabela (b)

Matriz de Pesos

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Cada documento é agora representado por um vetor de valores reais de pesos tf-idf $\in R^{|V|}$

Documentos como Vetores

- Termos e documentos são vetores em um espaço vetorial |V|-dimensional.
- Termos são os eixos do espaço.
- Documentos são pontos ou vetores nesse espaço.
- Muitas dimensões: milhões de dimensões quando se trata da Web.
- Vetores esparsos – maioria dos componentes é zero.

Consultas como Vetores

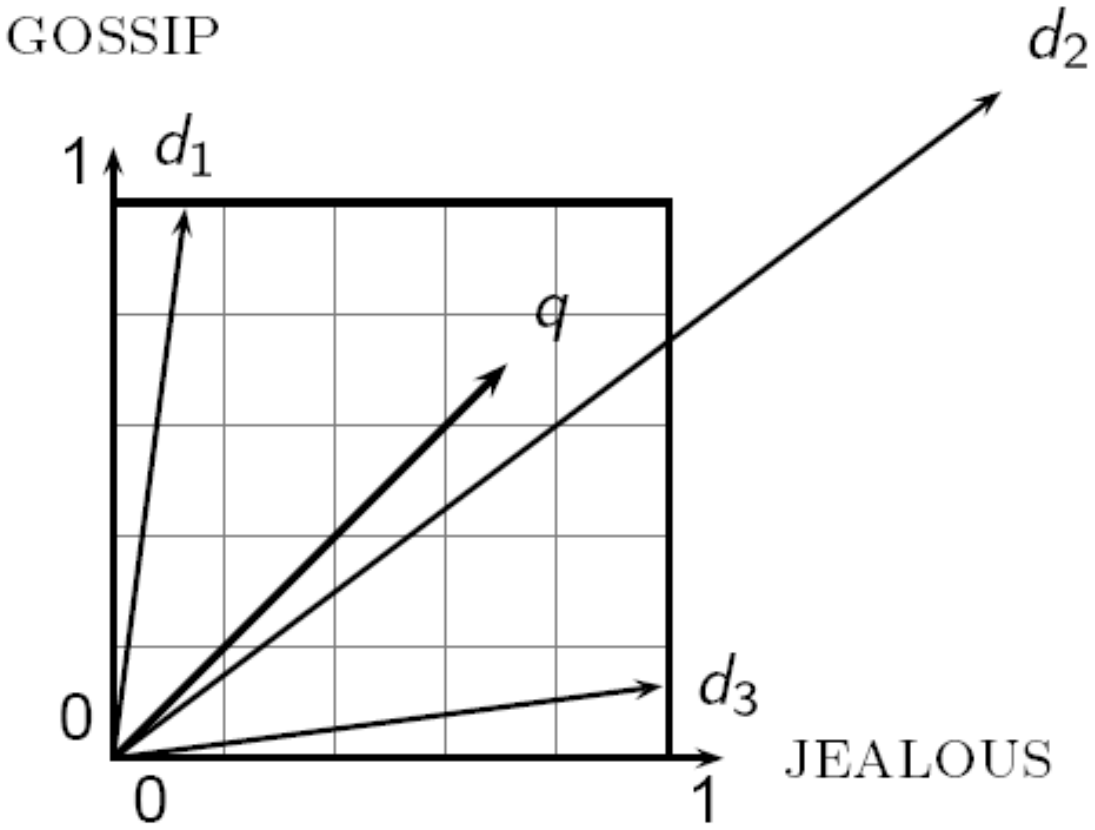
- Ideia-chave 1: Representar consultas como vetores no espaço.
- Ideia-chave 2: Ordenar os documentos de acordo com a sua proximidade com a consulta nesse espaço.
- proximidade = similaridade de vetores.
- proximidade ≈ inverso da distância.
- Lembrete: queremos nos livrar do modelo Booleano.
- Em vez disso: ranquear os documentos mais relevantes na frente dos não relevantes.

Proximidade entre vetores

- Primeira tentativa: distância entre dois pontos
  - (= distância entre os pontos finais dos dois vetores)
- Distância Euclidiana?
- Distância Euclidiana é uma má ideia . . .
- . . . porque a distância Euclidiana é grande para vetores de tamanhos diferentes.

Por que não distância Euclidiana

A distância Euclidiana entre *q* e d_2 é grande mesmo quando a distribuição de termos na consulta *q* e *documento* d_2 são muito similares.



Ângulo em vez de distância

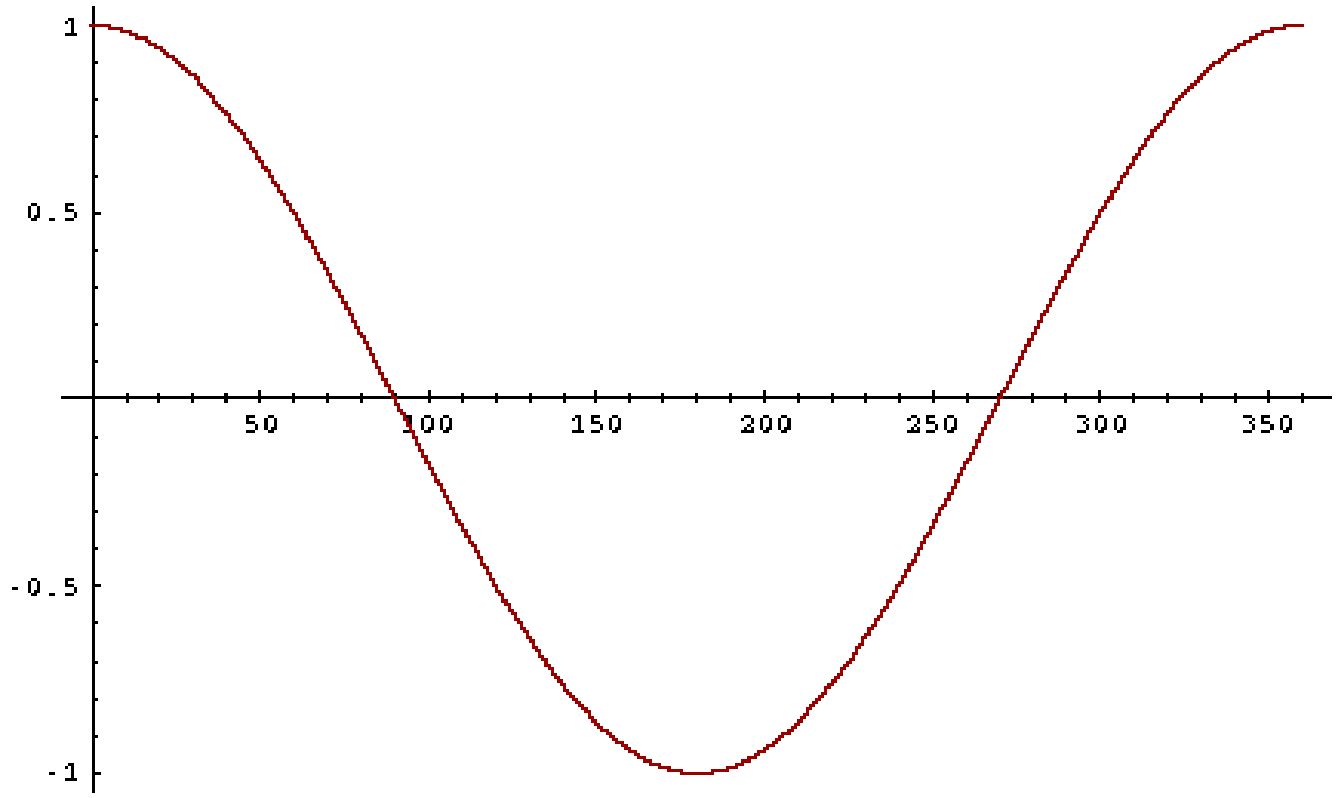
- Experimento mental: pegue um documento *d* e concatene-o a ele mesmo. Chame esse documento de *d′*.
- “Semanticamente” d e d′ possuem o mesmo conteúdo.
- A distância Euclidiana entre os dois documentos pode ser grande mesmo assim.

- O ângulo entre os dois documentos é 0, correspondendo a similaridade máxima.
- Ideia-chave: ordenar documentos de acordo com o seu ângulo com a consulta.

De Ângulos a Cosenos

- As duas noções seguintes são equivalentes:
  - Ordene documentos em ordem decrescente do ângulo entre a consulta e os documentos
  - Ordene documentos em ordem crescente do coseno (consulta,documento)
- Coseno é uma função decrescente monotônica para o intervalo [0^o, 180^o].

De Ângulos a Cosenos



Normalização do Tamanho do Vetor

- O tamanho de um vetor pode ser normalizado dividindo seus componentes pelo seu tamanho – norma Euclidiana ou L_2:

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$$

- Dividindo um vetor por sua norma Euclidiana o torna um vetor unitário.
- Efeito nos dois documentos d e d′ do slide 30: eles possuem vetores idênticos depois da normalização.
  - Documentos longos e curtos agora possuem pesos compatíveis

Coseno(consulta,documento)

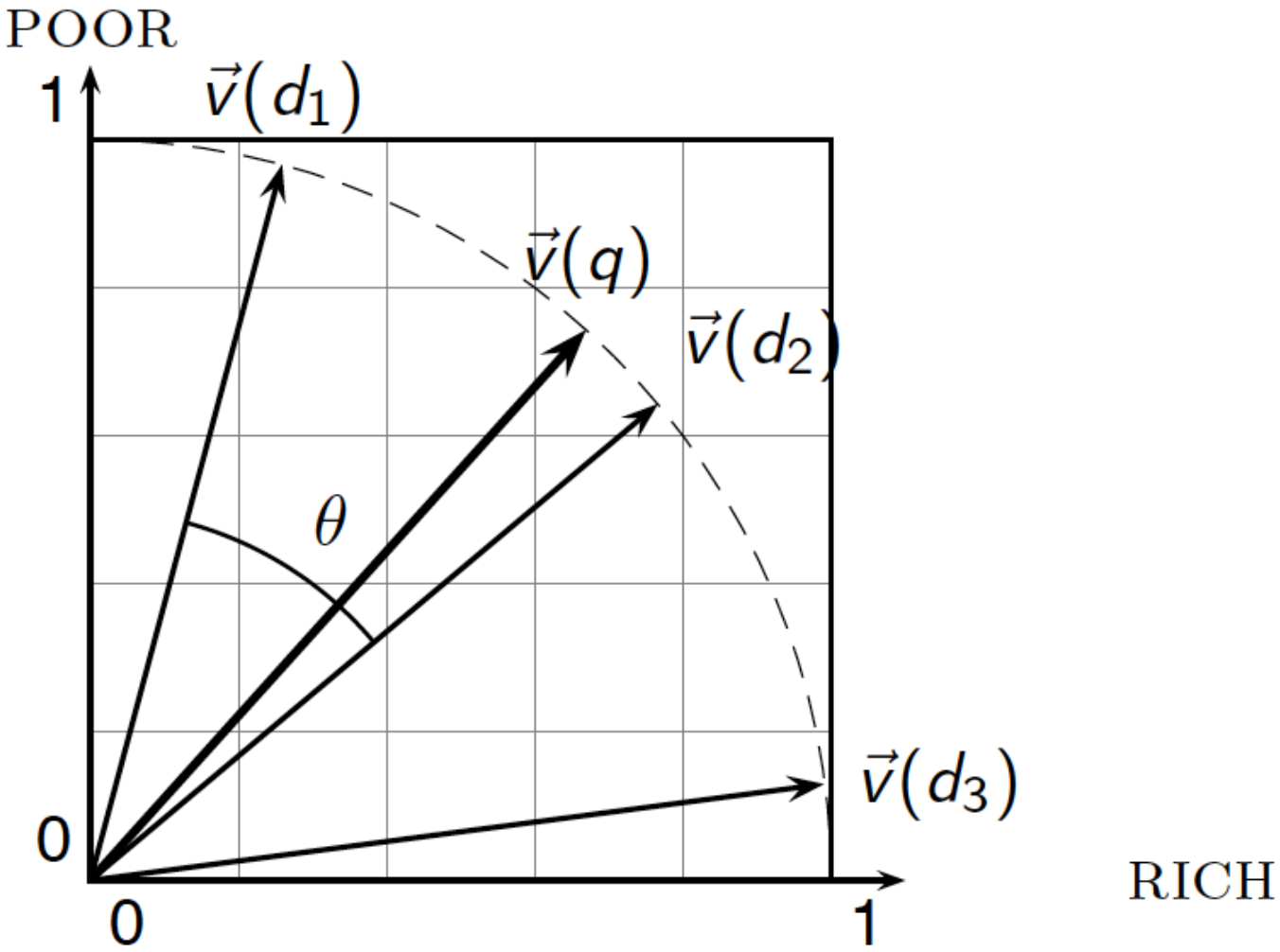
$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\cdot\frac{\vec{d}}{|\vec{d}|} = \frac{\sum q_i d_i}{\sqrt{\sum q_i^2}\sqrt{\sum d_i^2}}$$

q_i é o peso (e.g. tf-idf) do termo *i* na consulta

d_i é o peso (e.g. tf-idf) do termo *i* no document

cos(*q*,*d*) é a similaridade do coseno de *q* e *d* ... ou, equivalentemente, o coseno do ângulo entre *q* e *d*.

Similaridade do Coseno



Exemplo de Coseno entre 3 docs.

Quão similares são os documentos?

SaS: *Sense and Sensibility*

PaP: *Pride and Prejudice*, and

WH: *Wuthering Heights*?

termo	SaS	PaP	WH
affection	115	58	20
jealous	10	7	11
gossip	2	0	6
wuthering	0	0	38

Freq. de termos

Exemplo cont.

Log-frequencia

term	SaS	PaP	WH
affection	3.06	2.76	2.30
jealous	2.00	1.85	2.04
gossip	1.30	0	1.78
wuthering	0	0	2.58

Depois da normalização

term	SaS	PaP	WH
affection	0.789	0.832	0.524
jealous	0.515	0.555	0.465
gossip	0.335	0	0.405
wuthering	0	0	0.588

$\cos(SaS,PaP) \approx 0.789 \times 0.832 + 0.515 \times 0.555 + 0.335 \times 0.0 + 0.0 \times 0.0 \approx 0.94$
$\cos(SaS,WH) \approx 0.79$
$\cos(PaP,WH) \approx 0.69$

Algoritmo de Ranqueamento com Coseno

```
CosineScore(q)
 1  float Scores[N] = 0
 2  float Length[N]
 3  for each query term t
 4  do calculate w_{t,q} and fetch postings list for t
 5      for each pair(d, tf_{t,d}) in postings list
 6      do Scores[d] += w_{t,d} × w_{t,q}
 7  Read the array Length
 8  for each d
 9  do Scores[d] = Scores[d]/Length[d]
10  return Top K components of Scores[]
```

Outros Pesos

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$				

Esquemas de Pesos

- Muitos motores de busca permitem esquemas de peso diferentes para consultas vs. documentos
- Notação SMART: denota a combinação em uso por um motor, no qual a notação *ddd.qqq,* usa os acrônimos da tabela prévia.
- Um esquema padrão é: lnc.ltc
- Documento: tf logarítmico, sem idf e normalização coseno
- Consulta: tf logarítmico tf , idf, sem normalização ...

Sumário da Aula de Hoje

- Representar a consulta como um vetor de pesos.
- Representar cada documento como um vetor de pesos de termos.
- Calcular a similaridade de coseno entre o vetor de consulta e cada documento.
- Ordenar os documentos de acordo com sua similidade com a consulta.
- Retornar os top K (e.g., $K = 10$) ao usuário