

# ORI: Pontuação e o modelo de espaço vetorial

Marcelo Keese Albertini

Faculdade de Computação  
Universidade Federal de Uberlândia

Porque recuperação ordenada?

Frequência de termos

Peso tf-idf

O modelo espaço de vetores

# Veremos hoje

- ▶ **Ordenando** resultados de busca: importância (ao invés de apresentar um conjunto desordenado de resultados)

# Veremos hoje

- ▶ **Ordenando** resultados de busca: importância (ao invés de apresentar um conjunto desordenado de resultados)
- ▶ **Frequência de termos**: base da ordenação de resultados (*ranking*)

# Veremos hoje

- ▶ **Ordenando** resultados de busca: importância (ao invés de apresentar um conjunto desordenado de resultados)
- ▶ **Frequência de termos**: base da ordenação de resultados (*ranking*)
- ▶ **Tf-idf ranking**: esquema tradicional de ordenação

# Recuperação ordenada

# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.

# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.
  - ▶ Documentos são adequados ou não para uma consulta



# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.
  - ▶ Documentos são adequados ou não para uma consulta
- ▶ **Bom para usuários especialistas** com conhecimento avançado sobre a coleção

# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.
  - ▶ Documentos são adequados ou não para uma consulta
- ▶ **Bom para usuários especialistas** com conhecimento avançado sobre a coleção
- ▶ **Bom para programas**: programas podem processar milhares de resultados

# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.
  - ▶ Documentos são adequados ou não para uma consulta
- ▶ **Bom para usuários especialistas** com conhecimento avançado sobre a coleção
- ▶ **Bom para programas**: programas podem processar milhares de resultados
- ▶ **Não tão bom para usuários comuns**

# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.
  - ▶ Documentos são adequados ou não para uma consulta
- ▶ **Bom para usuários especialistas** com conhecimento avançado sobre a coleção
- ▶ **Bom para programas**: programas podem processar milhares de resultados
- ▶ **Não tão bom para usuários comuns**
- ▶ Consultas booleanas são de difícil escrita

# Recuperação ordenada

- ▶ Até agora, consultas foram **booleanas**.
  - ▶ Documentos são adequados ou não para uma consulta
- ▶ **Bom para usuários especialistas** com conhecimento avançado sobre a coleção
- ▶ **Bom para programas**: programas podem processar milhares de resultados
- ▶ **Não tão bom para usuários comuns**
- ▶ Consultas booleanas são de difícil escrita
- ▶ Usuários não olham centenas de resultados

# Problemas com busca booleana: tudo ou nada

## Problemas com busca booleana: tudo ou nada

`http://www.acervobiblioteca.ufu.br:8000/cgi-bin/gw/chameleon`

- ▶ Muito pouco ou resultados demais

## Problemas com busca booleana: tudo ou nada

<http://www.acervobiblioteca.ufu.br:8000/cgi-bin/gw/chameleon>

- ▶ Muito pouco ou resultados demais
- ▶ Exemplo consulta 1 (conjunção booleana): [recuperação AND informação]



# Problemas com busca booleana: tudo ou nada

<http://www.acervobiblioteca.ufu.br:8000/cgi-bin/gw/chameleon>

- ▶ Muito pouco ou resultados demais
- ▶ Exemplo consulta 1 (conjunção booleana): [recuperação AND informação]
  - ▶ → centenas de resultados – **demais**

# Problemas com busca booleana: tudo ou nada

<http://www.acervobiblioteca.ufu.br:8000/cgi-bin/gw/chameleon>

- ▶ Muito pouco ou resultados demais
- ▶ Exemplo consulta 1 (conjunção booleana): [recuperação AND informação]
  - ▶ → centenas de resultados – **demais**
- ▶ Exemplo 2 (conjunção booleana): [recuperação AND informação AND aplicação]

# Problemas com busca booleana: tudo ou nada

<http://www.acervobiblioteca.ufu.br:8000/cgi-bin/gw/chameleon>

- ▶ Muito pouco ou resultados demais
- ▶ Exemplo consulta 1 (conjunção booleana): [recuperação AND informação]
  - ▶ → centenas de resultados – **demais**
- ▶ Exemplo 2 (conjunção booleana): [recuperação AND informação AND aplicação]
  - ▶ → 2 resultados – **quase nada**

# Problemas com busca booleana: tudo ou nada

<http://www.acervobiblioteca.ufu.br:8000/cgi-bin/gw/chameleon>

- ▶ Muito pouco ou resultados demais
- ▶ Exemplo consulta 1 (conjunção booleana): [recuperação AND informação]
  - ▶ → centenas de resultados – **demais**
- ▶ Exemplo 2 (conjunção booleana): [recuperação AND informação AND aplicação]
  - ▶ → 2 resultados – **quase nada**
- ▶ difícil encontrar boa consulta para obter entre tudo ou nada

Tudo ou nada: não é problema com recuperação ordenada

# Tudo ou nada: não é problema com recuperação ordenada

- ▶ Com ordenação, número de resultados não é problema

## Tudo ou nada: não é problema com recuperação ordenada

- ▶ Com ordenação, número de resultados não é problema
- ▶ Por exemplo, mostrar somente os 10 mais relevantes

## Tudo ou nada: não é problema com recuperação ordenada

- ▶ Com ordenação, número de resultados não é problema
- ▶ Por exemplo, mostrar somente os 10 mais relevantes
- ▶ Não sobrecarrega usuário



# Tudo ou nada: não é problema com recuperação ordenada

- ▶ Com ordenação, número de resultados não é problema
- ▶ Por exemplo, mostrar somente os 10 mais relevantes
- ▶ Não sobrecarrega usuário
- ▶ O que é necessário? Desenvolver um algoritmo de ranking de relevância de documentos

# Avaliação como base de recuperação ordenada

# Avaliação como base de recuperação ordenada

- ▶ Pontuar mais os documentos mais relevantes à consulta

# Avaliação como base de recuperação ordenada

- ▶ Pontuar mais os documentos mais relevantes à consulta
- ▶ Atribuir pontuação em  $[0, 1]$  para cada par consulta-documento

# Avaliação como base de recuperação ordenada

- ▶ Pontuar mais os documentos mais relevantes à consulta
- ▶ Atribuir pontuação em  $[0, 1]$  para cada par consulta-documento
- ▶ Medida numérica e objetiva da relevância do documento para a consulta

# Pontuando consultas-documentos

- ▶ Como pontuamos um par consulta-documento?

# Pontuando consultas-documentos

- ▶ Como pontuamos um par consulta-documento?
- ▶ Começamos com um consulta de um só termo

# Pontuando consultas-documentos

- ▶ Como pontuamos um par consulta-documento?
- ▶ Começamos com um consulta de um só termo
- ▶ Se o termo não ocorre no documento, pontuação 0



# Pontuando consultas-documentos

- ▶ Como pontuamos um par consulta-documento?
- ▶ Começamos com um consulta de um só termo
- ▶ Se o termo não ocorre no documento, pontuação 0
- ▶ Quanto maior a frequência do termo no documento, maior pontuação

# Pontuando consultas-documentos

- ▶ Como pontuamos um par consulta-documento?
- ▶ Começamos com um consulta de um só termo
- ▶ Se o termo não ocorre no documento, pontuação 0
- ▶ Quanto maior a frequência do termo no documento, maior pontuação
- ▶ Veremos alternativas

## Alternativa 1: coeficiente de Jaccard

## Alternativa 1: coeficiente de Jaccard

- ▶ Mede sobreposição de 2 conjuntos:  $A$  e  $B$

## Alternativa 1: coeficiente de Jaccard

- ▶ Mede sobreposição de 2 conjuntos:  $A$  e  $B$
- ▶ Coeficiente de Jaccard:

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

$(A \neq \emptyset \text{ ou } B \neq \emptyset)$

## Alternativa 1: coeficiente de Jaccard

- ▶ Mede sobreposição de 2 conjuntos:  $A$  e  $B$
- ▶ Coeficiente de Jaccard:

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

$(A \neq \emptyset \text{ ou } B \neq \emptyset)$

- ▶  $\text{JACCARD}(A, A) = 1$

## Alternativa 1: coeficiente de Jaccard

- ▶ Mede sobreposição de 2 conjuntos:  $A$  e  $B$
- ▶ Coeficiente de Jaccard:

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

$(A \neq \emptyset \text{ ou } B \neq \emptyset)$

- ▶  $\text{JACCARD}(A, A) = 1$
- ▶  $\text{JACCARD}(A, B) = 0$  se  $A \cap B = \emptyset$

## Alternativa 1: coeficiente de Jaccard

- ▶ Mede sobreposição de 2 conjuntos:  $A$  e  $B$
- ▶ Coeficiente de Jaccard:

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

$(A \neq \emptyset \text{ ou } B \neq \emptyset)$

- ▶  $\text{JACCARD}(A, A) = 1$
- ▶  $\text{JACCARD}(A, B) = 0$  se  $A \cap B = \emptyset$
- ▶  $A$  e  $B$  não tem que ser do mesmo tamanho



## Alternativa 1: coeficiente de Jaccard

- ▶ Mede sobreposição de 2 conjuntos:  $A$  e  $B$
- ▶ Coeficiente de Jaccard:

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

$(A \neq \emptyset \text{ ou } B \neq \emptyset)$

- ▶  $\text{JACCARD}(A, A) = 1$
- ▶  $\text{JACCARD}(A, B) = 0$  se  $A \cap B = \emptyset$
- ▶  $A$  e  $B$  não tem que ser do mesmo tamanho
- ▶ Sempre obtém número entre 0 e 1

## Exemplo: coeficiente de Jaccard

## Exemplo: coeficiente de Jaccard

- ▶ Qual é a pontuação pelo coeficiente de Jaccard para:

## Exemplo: coeficiente de Jaccard

- ▶ Qual é a pontuação pelo coeficiente de Jaccard para:
  - ▶ Consulta: “águas de março”

## Exemplo: coeficiente de Jaccard

- ▶ Qual é a pontuação pelo coeficiente de Jaccard para:
  - ▶ Consulta: “águas de março”
  - ▶ Documento “Pedro Álvares Cabral chegou nas águas brasileiras em março”

## Exemplo: coeficiente de Jaccard

- ▶ Qual é a pontuação pelo coeficiente de Jaccard para:
  - ▶ Consulta: “águas de março”
  - ▶ Documento “Pedro Álvares Cabral chegou nas águas brasileiras em março”
  - ▶  $JACCARD(q, d) = 2/10$

# Onde Jaccard falha?

## Onde Jaccard falha?

- ▶ Não considera frequência dos termos



## Onde Jaccard falha?

- ▶ Não considera frequência dos termos
- ▶ Termos raros são mais informativos que os frequentes

## Onde Jaccard falha?

- ▶ Não considera frequência dos termos
- ▶ Termos raros são mais informativos que os frequentes
- ▶ Precisamos de modos para normalizar pelo tamanho do documento

# Onde Jaccard falha?

- ▶ Não considera frequência dos termos
- ▶ Termos raros são mais informativos que os frequentes
- ▶ Precisamos de modos para normalizar pelo tamanho do documento
  - ▶ um documento grande provavelmente tem boa sobreposição com a maior parte das consultas mas não é necessariamente relevante

## Matriz de incidência binária

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
César	1	1	0	1	1	1	
Calpúrnia	0	1	0	0	0	0	
Cleópatra	1	0	0	0	0	0	
...							

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

## Matriz de incidência binária

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
César	1	1	0	1	1	1	
Calpúrnia	0	1	0	0	0	0	
Cleópatra	1	0	0	0	0	0	
...							

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

# Matriz de contagem

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
César	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleópatra	57	0	0	0	0	0	
...							

Cada documento é representado como vetor de contagem  $\in \mathbb{N}^{|V|}$ .

# Matriz de contagem

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
César	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleópatra	57	0	0	0	0	0	
...							

Cada documento é representado como **vetor de contagem**  $\in \mathbb{N}^{|V|}$ .

# Modelo Bag of words / coleção de palavras



## Modelo Bag of words / coleção de palavras

- ▶ Desconsidera **ordem** dos termos em um documento.

## Modelo Bag of words / coleção de palavras

- ▶ Desconsidera **ordem** dos termos em um documento.
- ▶ *João é mais rápido que José* tem mesma representação que *José é mais rápido que João*

## Modelo Bag of words / coleção de palavras

- ▶ Desconsidera **ordem** dos termos em um documento.
- ▶ *João é mais rápido que José* tem mesma representação que *José é mais rápido que João*
- ▶ Isso é chamado de **modelo bag of words**.

## Modelo Bag of words / coleção de palavras

- ▶ Desconsidera **ordem** dos termos em um documento.
- ▶ *João é mais rápido que José* tem mesma representação que *José é mais rápido que João*
- ▶ Isso é chamado de **modelo bag of words**.
- ▶ Seguiremos com mais detalhes do modelo bag of words.

## Frequência de termo $tf$

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$ .**

## Frequência de termo tf

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$** .
- ▶ Podemos usar tf para pontuar combinação consulta-documento.

# Frequência de termo tf

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$ .**
- ▶ Podemos usar tf para pontuar combinação consulta-documento.
- ▶ Porém, somente frequência não é bom porque:

# Frequência de termo $tf$

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$ .**
- ▶ Podemos usar  $tf$  para pontuar combinação consulta-documento.
- ▶ Porém, somente frequência não é bom porque:
- ▶ Um documento com  **$tf = 10$**  ocorrências de um termo é mais relevante que um documento com apenas uma ocorrência  **$tf = 1$ .**



# Frequência de termo tf

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$ .**
- ▶ Podemos usar tf para pontuar combinação consulta-documento.
- ▶ Porém, somente frequência não é bom porque:
- ▶ Um documento com **tf = 10** ocorrências de um termo é mais relevante que um documento com apenas uma ocorrência **tf = 1**.
- ▶ Mas não 10 vezes mais relevante

# Frequência de termo $tf$

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$ .**
- ▶ Podemos usar  $tf$  para pontuar combinação consulta-documento.
- ▶ Porém, somente frequência não é bom porque:
- ▶ **Um documento com  $tf = 10$  ocorrências de um termo é mais relevante que um documento com apenas uma ocorrência  $tf = 1$ .**
- ▶ Mas não 10 vezes mais relevante
- ▶ **Relevância não aumenta proporcionalmente com a frequência do termo.**

## Frequência de termo $tf$

- ▶ A frequência de termo  $tf_{t,d}$  do termo  $t$  no documento  $d$  é definido como **o número de vezes que  $t$  ocorre em  $d$ .**
- ▶ Podemos usar  $tf$  para pontuar combinação consulta-documento.
- ▶ Porém, somente frequência não é bom porque:
- ▶ Um documento com  **$tf = 10$**  ocorrências de um termo é mais relevante que um documento com apenas uma ocorrência  **$tf = 1$ .**
- ▶ Mas não 10 vezes mais relevante
- ▶ Relevância não aumenta proporcionalmente com a frequência do termo.
- ▶ **Um documento com diversos termos da consulta é mais relevante** que outro documento com muitas repetições de apenas um termo

Em vez de frequência:  $\log$  da frequência

## Em vez de frequência: log da frequência

- ▶ O log da frequência do termo  $t$  em  $d$  é definido:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{se } \text{tf}_{t,d} > 0 \\ 0 & \text{caso contrário} \end{cases}$$

## Em vez de frequência: log da frequência

- ▶ O log da frequência do termo  $t$  em  $d$  é definido:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{se } \text{tf}_{t,d} > 0 \\ 0 & \text{caso contrário} \end{cases}$$

- ▶  $\text{tf}_{t,d} \rightarrow w_{t,d}$ :  
 $0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4$  etc.

## Em vez de frequência: log da frequência

- ▶ O log da frequência do termo  $t$  em  $d$  é definido:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{se } \text{tf}_{t,d} > 0 \\ 0 & \text{caso contrário} \end{cases}$$

- ▶  $\text{tf}_{t,d} \rightarrow w_{t,d}$ :  
 $0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4$  etc.
- ▶ Pontuação para um par consulta-documento: soma em relação a termos  $t$  em  $q$  e  $d$ :

$$\text{pontuação-tf}(q, d) = \sum_{t \in q \cap d} (1 + \log \text{tf}_{t,d})$$

## Em vez de frequência: log da frequência

- ▶ O log da frequência do termo  $t$  em  $d$  é definido:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{se } \text{tf}_{t,d} > 0 \\ 0 & \text{caso contrário} \end{cases}$$

- ▶  $\text{tf}_{t,d} \rightarrow w_{t,d}$ :  
 $0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4$  etc.
- ▶ Pontuação para um par consulta-documento: soma em relação a termos  $t$  em  $q$  e  $d$ :  
**pontuação-tf** $(q, d) = \sum_{t \in q \cap d} (1 + \log \text{tf}_{t,d})$
- ▶ A pontuação é 0 se nenhum dos termos está presente no documento.



# Exercício

- ▶ Calcular a pontuação de Jaccard e pontuação de tf para os pares consulta-documento:
- ▶ q: [informação sobre carros] d: “tudo o que você sempre quis saber sobre carros”
- ▶ q: [informação sobre carros] d: “informação sobre caminhões, informação sobre aviões, informação sobre trens”
- ▶ q: [carros verdes e caminhões verdes] d: “a polícia para carros verdes mais frequentemente”

# Frequência no documento vs. frequência na coleção

# Frequência no documento vs. frequência na coleção

- ▶ Frequência de termo no documento

# Frequência no documento vs. frequência na coleção

- ▶ Frequência de termo no documento
- ▶ Frequência de termo **na coleção**

## Peso desejado para termos raros

## Peso desejado para termos raros

- ▶ Termos raros são mais informativos

## Peso desejado para termos raros

- ▶ Termos raros são mais informativos
- ▶ Considere um termo em uma consulta que é **raro** na coleção, e.g. estoicismo

## Peso desejado para termos raros

- ▶ Termos raros são mais informativos
- ▶ Considere um termo em uma consulta que é **raro** na coleção, e.g. estoicismo
- ▶ Um documento com esse termo é muito provavelmente relevante



## Peso desejado para termos raros

- ▶ Termos raros são mais informativos
- ▶ Considere um termo em uma consulta que é **raro** na coleção, e.g. estoicismo
- ▶ Um documento com esse termo é muito provavelmente relevante
- ▶ → Nós queremos **pesos altos para termos raros**

## Peso desejado para termos raros

- ▶ Termos raros são mais informativos
- ▶ Considere um termo em uma consulta que é **raro** na coleção, e.g. estoicismo
- ▶ Um documento com esse termo é muito provavelmente relevante
- ▶ → Nós queremos **pesos altos para termos raros**
- ▶ → Nós queremos **pesos baixos para termos frequentes**



# Peso idf

- ▶  $df_t$  é a frequência na coleção de documentos, ou seja, é o número de documentos em que  $t$  aparece

# Peso idf

- ▶  $df_t$  é a frequência na coleção de documentos, ou seja, é o número de documentos em que  $t$  aparece
- ▶  $df_t$  é uma medida inversa da **informação** do termo  $t$

# Peso idf

- ▶  $df_t$  é a frequência na coleção de documentos, ou seja, é o número de documentos em que  $t$  aparece
- ▶  $df_t$  é uma medida inversa da **informação** do termo  $t$
- ▶ Define-se **peso idf** do termo  $t$  como segue:

$$idf_t = \log_{10} \frac{N}{df_t}$$

( $N$  é o número de documentos na coleção.)

# Peso idf

- ▶  $df_t$  é a frequência na coleção de documentos, ou seja, é o número de documentos em que  $t$  aparece
- ▶  $df_t$  é uma medida inversa da **informação** do termo  $t$
- ▶ Define-se **peso idf** do termo  $t$  como segue:

$$idf_t = \log_{10} \frac{N}{df_t}$$

( $N$  é o número de documentos na coleção.)

- ▶  **$idf_t$**  é a medida de **informação** do termo

# Peso idf

- ▶  $df_t$  é a frequência na coleção de documentos, ou seja, é o número de documentos em que  $t$  aparece
- ▶  $df_t$  é uma medida inversa da **informação** do termo  $t$
- ▶ Define-se **peso idf** do termo  $t$  como segue:

$$idf_t = \log_{10} \frac{N}{df_t}$$

( $N$  é o número de documentos na coleção.)

- ▶  **$idf_t$**  é a medida de **informação** do termo
- ▶  $[\log N/df_t]$  em vez de  $[N/df_t]$  para amenizar o efeito de idf



## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	
animal	100	
domingo	1000	
voar	10,000	
sobre	100,000	
o	1,000,000	

## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	<b>6</b>
animal	100	
domingo	1000	
voar	10,000	
sobre	100,000	
o	1,000,000	

## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	<b>6</b>
animal	100	<b>4</b>
domingo	1000	
voar	10,000	
sobre	100,000	
o	1,000,000	

## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	<b>6</b>
animal	100	<b>4</b>
domingo	1000	<b>3</b>
voar	10,000	
sobre	100,000	
o	1,000,000	

## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	<b>6</b>
animal	100	<b>4</b>
domingo	1000	<b>3</b>
voar	10,000	<b>2</b>
sobre	100,000	
o	1,000,000	

## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	<b>6</b>
animal	100	<b>4</b>
domingo	1000	<b>3</b>
voar	10,000	<b>2</b>
sobre	100,000	<b>1</b>
o	1,000,000	

## Exemplos de idf

Calcular  $idf_t$  usando a fórmula:  $idf_t = \log_{10} \frac{1,000,000}{df_t}$

termo	$df_t$	$idf_t$
calpurnia	1	<b>6</b>
animal	100	<b>4</b>
domingo	1000	<b>3</b>
voar	10,000	<b>2</b>
sobre	100,000	<b>1</b>
o	1,000,000	<b>0</b>

# Efeito de idf no ranking



## Efeito de idf no ranking

- ▶ A medida **idf** influencia na ordenação quando há pelo menos 2 termos

## Efeito de idf no ranking

- ▶ A medida **idf** influencia na ordenação quando há pelo menos 2 termos
- ▶ Por exemplo, na consulta “estoicismo antigo”, peso idf **aumenta** o peso relativo de estoicismo e **reduz** peso relativo de antigo.

## Efeito de idf no ranking

- ▶ A medida **idf** influencia na ordenação quando há pelo menos 2 termos
- ▶ Por exemplo, na consulta “estoicismo antigo”, peso idf **aumenta** o peso relativo de estoicismo e **reduz** peso relativo de antigo.
- ▶ O **idf** tem **pouco efeito** em consultas com **um termo**.

## Frequência na coleção vs. frequência no documento

termo	frequência na coleção	frequência no documento
seguro	10440	3997
tentar	10422	8760

- ▶ Frequência de  $t$  na coleção: número de ocorrências de  $t$  na coleção
- ▶ Frequência de  $t$  em documentos: número de documentos em que  $t$  ocorre

## Frequência na coleção vs. frequência no documento

termo	frequência na coleção	frequência no documento
seguro	10440	3997
tentar	10422	8760

- ▶ Frequência de  $t$  na coleção: número de ocorrências de  $t$  na coleção
- ▶ Frequência de  $t$  em documentos: número de documentos em que  $t$  ocorre
- ▶ Qual termo é melhor como termo de busca?

## Frequência na coleção vs. frequência no documento

termo	frequência na coleção	frequência no documento
seguro	10440	3997
tentar	10422	8760

- ▶ Frequência de  $t$  na coleção: número de ocorrências de  $t$  na coleção
- ▶ Frequência de  $t$  em documentos: número de documentos em que  $t$  ocorre
- ▶ Qual termo é melhor como termo de busca?
- ▶ Este exemplo sugere que  $df$  (e  $idf$ ) é melhor como peso que  $cf$  (e “ $icf$ ”)

# Peso tf-idf

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

# Peso tf-idf

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



$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$



# Peso tf-idf

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



$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

# Peso tf-idf

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



$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

- ▶ peso tf

# Peso tf-idf

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



$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

- ▶ peso idf
- ▶ Esquema bastante conhecido em RI.

# Peso tf-idf

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



$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

- ▶ Esquema bastante conhecido em RI.
- ▶ Outros nomes: tf.idf, tf x idf

# Resumo: tf-idf

## Resumo: tf-idf

- ▶ Atribuir peso tf-idf para cada termo  $t$  em cada documento  $d$ :  
$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

## Resumo: tf-idf

- ▶ Atribuir peso tf-idf para cada termo  $t$  em cada documento  $d$ :  
$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$
- ▶ O peso tf-idf ...

## Resumo: tf-idf

- ▶ Atribuir peso tf-idf para cada termo  $t$  em cada documento  $d$ :  
$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$
- ▶ O peso tf-idf ...
  - ▶ ... aumenta com o número de ocorrência em um documento.  
(frequência do termo)



## Resumo: tf-idf

- ▶ Atribuir peso tf-idf para cada termo  $t$  em cada documento  $d$ :  
$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$
- ▶ O peso tf-idf ...
  - ▶ ... aumenta com o número de ocorrência em um documento. (frequência do termo)
  - ▶ ... aumenta com a raridade do termo na coleção. (frequência em document inversa)

## Exercício: frequência de termo, coleção e documento

Quantidade	Símbolo	Definição
frequência de termo	$tf_{t,d}$	número de ocorrências de $t$ em $d$
frequência de documentos	$df_t$	número de documentos em que $t$ ocorre
frequência de coleção	$cf_t$	número total de ocorrências de $t$ na coleção (incluindo repetições em documentos)

- ▶ Relação entre  $df$  e  $cf$ ?
- ▶ Relação entre  $tf$  e  $cf$ ?
- ▶ Relação entre  $tf$  e  $df$ ?

## Matriz de incidência binária

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
César	1	1	0	1	1	1	
Calpúrnia	0	1	0	0	0	0	
Cleópatra	1	0	0	0	0	0	
...							

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

## Matriz de incidência binária

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
César	1	1	0	1	1	1	
Calpúrnia	0	1	0	0	0	0	
Cleópatra	1	0	0	0	0	0	
...							

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

## Matriz de contagem

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
César	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleópatra	57	0	0	0	0	0	
...							

Cada documento é representado como vetor de contagem  $\in \mathbb{N}^{|V|}$ .

# Matriz de contagem

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
César	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleópatra	57	0	0	0	0	0	
...							

Cada documento é representado como **vetor de contagem**  $\in \mathbb{N}^{|V|}$ .

## Binário → contagem → matriz de pesos

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	5.25	3.18	0.0	0.0	0.0	0.35	
Brutus	1.21	6.10	0.0	1.0	0.0	0.0	
César	8.59	2.54	0.0	1.51	0.25	0.0	
Calpúrnia	0.0	1.54	0.0	0.0	0.0	0.0	
Cleópatra	2.85	0.0	0.0	0.0	0.0	0.0	
misericórdia	1.51	0.0	1.90	0.12	5.25	0.88	
pior	1.37	0.0	0.11	4.15	0.25	1.95	
...							

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

## Binário → contagem → matriz de pesos

	Marco Antônio	Júlio César	A Tempestade	Hamlet	Otelo	Macbeth	...
Antônio	5.25	3.18	0.0	0.0	0.0	0.35	
Brutus	1.21	6.10	0.0	1.0	0.0	0.0	
César	8.59	2.54	0.0	1.51	0.25	0.0	
Calpúrnia	0.0	1.54	0.0	0.0	0.0	0.0	
Cleópatra	2.85	0.0	0.0	0.0	0.0	0.0	
misericórdia	1.51	0.0	1.90	0.12	5.25	0.88	
pior	1.37	0.0	0.11	4.15	0.25	1.95	
...							

Cada documento é representado como um **vetor de valores reais** de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .



# Documentos na forma de vetores

## Documentos na forma de vetores

- ▶ Cada documento é representado em um vetor de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .

## Documentos na forma de vetores

- ▶ Cada documento é representado em um vetor de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .
- ▶ Então temos um espaço vetorial com  $|V|$  dimensões.

# Documentos na forma de vetores

- ▶ Cada documento é representado em um vetor de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .
- ▶ Então temos um espaço vetorial com  $|V|$  dimensões.
- ▶ Termos são **eixos** do espaço.

## Documentos na forma de vetores

- ▶ Cada documento é representado em um vetor de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .
- ▶ Então temos um espaço vetorial com  $|V|$  dimensões.
- ▶ Termos são **eixos** do espaço.
- ▶ Documentos são **pontos** ou **vetores** nesse espaço.

## Documentos na forma de vetores

- ▶ Cada documento é representado em um vetor de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .
- ▶ Então temos um espaço vetorial com  $|V|$  dimensões.
- ▶ Termos são **eixos** do espaço.
- ▶ Documentos são **pontos** ou **vetores** nesse espaço.
- ▶ Alto número de dimensões: dezenas de milhões de dimensões em mecanismos de busca

# Documentos na forma de vetores

- ▶ Cada documento é representado em um vetor de pesos tf-idf  $\in \mathbb{R}^{|V|}$ .
- ▶ Então temos um espaço vetorial com  $|V|$  dimensões.
- ▶ Termos são **eixos** do espaço.
- ▶ Documentos são **pontos** ou **vetores** nesse espaço.
- ▶ Alto número de dimensões: dezenas de milhões de dimensões em mecanismos de busca
- ▶ Cada vetor usa muito espaço (maior parte das dimensões é zero)

# Consultas como vetores



# Consultas como vetores

- ▶ Ideia 1: fazer o mesmo para as consultas: representar no espaço de alta-dimensionalidade

# Consultas como vetores

- ▶ Ideia 1: fazer o mesmo para as consultas: representar no espaço de alta-dimensionalidade
- ▶ Ideia 2: Rankear documentos de acordo com sua proximidade à consulta

# Consultas como vetores

- ▶ Ideia 1: fazer o mesmo para as consultas: representar no espaço de alta-dimensionalidade
- ▶ Ideia 2: Rankear documentos de acordo com sua proximidade à consulta
- ▶ proximidade = similaridade

# Consultas como vetores

- ▶ Ideia 1: fazer o mesmo para as consultas: representar no espaço de alta-dimensionalidade
- ▶ Ideia 2: Ranquear documentos de acordo com sua proximidade à consulta
- ▶ proximidade = similaridade
- ▶ proximidade  $\approx$  distância negativa

# Consultas como vetores

- ▶ Ideia 1: fazer o mesmo para as consultas: representar no espaço de alta-dimensionalidade
- ▶ Ideia 2: Ranquear documentos de acordo com sua proximidade à consulta
- ▶ proximidade = similaridade
- ▶ proximidade  $\approx$  distância negativa
- ▶ Objetivo: estamos evitando modelo booleana e resultados tudo ou nada.

## Consultas como vetores

- ▶ Ideia 1: fazer o mesmo para as consultas: representar no espaço de alta-dimensionalidade
- ▶ Ideia 2: Ranquear documentos de acordo com sua proximidade à consulta
- ▶ proximidade = similaridade
- ▶ proximidade  $\approx$  distância negativa
- ▶ Objetivo: estamos evitando modelo booleana e resultados tudo ou nada.
- ▶ Objetivo: ranquear documentos relevantes em melhores posições que os não relevantes

# Como formalizamos a similaridade no espaço vetorial

- ▶ distância (negativa) entre dois “pontos”

# Como formalizamos a similaridade no espaço vetorial

- ▶ distância (negativa) entre dois “pontos”
- ▶ ( = distância entre pontos finais entre pares de vetores)



# Como formalizamos a similaridade no espaço vetorial

- ▶ distância (negativa) entre dois “pontos”
- ▶ ( = distância entre pontos finais entre pares de vetores)
- ▶ Distância euclidiana

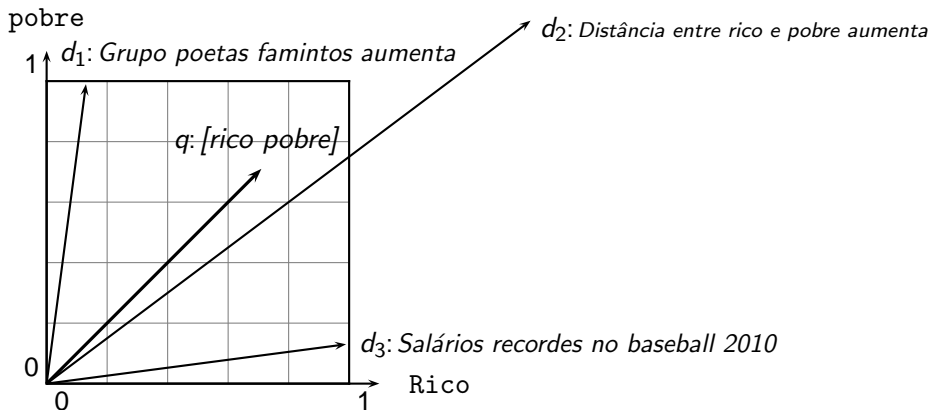
# Como formalizamos a similaridade no espaço vetorial

- ▶ distância (negativa) entre dois “pontos”
- ▶ ( = distância entre pontos finais entre pares de vetores)
- ▶ Distância euclidiana
- ▶ Distância euclidiana é uma má ideia ...

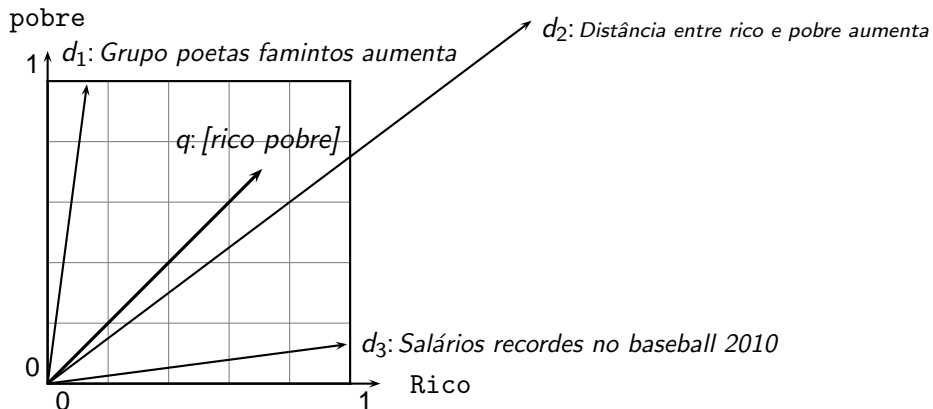
# Como formalizamos a similaridade no espaço vetorial

- ▶ distância (negativa) entre dois “pontos”
- ▶ ( = distância entre pontos finais entre pares de vetores)
- ▶ Distância euclidiana
- ▶ Distância euclidiana é uma má ideia ...
- ▶ ... porque distância euclidiana é **grande** para vetores de diferentes comprimentos

# Porque distância euclidiana é uma má ideia



# Porque distância euclidiana é uma má ideia



A distância euclidiana de  $\vec{q}$  e  $\vec{d}_2$  é grande, embora a distribuição de termos na consulta  $q$  e a distribuição dos termo no documento  $d_2$  são muito similares.

Usar ângulo em vez de distância

## Usar ângulo em vez de distância

- ▶ Ordena documento de acordo com o ângulo em relação à consulta

## Usar ângulo em vez de distância

- ▶ Ordena documento de acordo com o ângulo em relação à consulta
- ▶ Avalie: pegue um documento  $d$  e adicione-o a si mesmo em  $d'$ .



## Usar ângulo em vez de distância

- ▶ Ordena documento de acordo com o ângulo em relação à consulta
- ▶ Avalie: pegue um documento  $d$  e adicione-o a si mesmo em  $d'$ .
- ▶  $d$  e  $d'$  têm mesma informação

## Usar ângulo em vez de distância

- ▶ Ordena documento de acordo com o ângulo em relação à consulta
- ▶ Avalie: pegue um documento  $d$  e adicione-o a si mesmo em  $d'$ .
- ▶  $d$  e  $d'$  têm mesma informação
- ▶ O ângulo entre os dois documentos é 0, máxima similaridade  
...

## Usar ângulo em vez de distância

- ▶ Ordena documento de acordo com o ângulo em relação à consulta
- ▶ Avalie: pegue um documento  $d$  e adicione-o a si mesmo em  $d'$ .
- ▶  $d$  e  $d'$  têm mesma informação
- ▶ O ângulo entre os dois documentos é 0, máxima similaridade  
...
- ▶ ... mesmo que a distância euclidiana entre os dois documentos seja grande

# De ângulos a cosenos

# De ângulos a cosenos

- ▶ As seguintes noções são equivalentes:

# De ângulos a cosenos

- ▶ As seguintes noções são equivalentes:
  - ▶ Ordenar documentos de acordo com o **ângulo** entre consulta e documento em ordem decrescente

# De ângulos a cosenos

- ▶ As seguintes noções são equivalentes:
  - ▶ Ordenar documentos de acordo com o **ângulo** entre consulta e documento em ordem decrescente
  - ▶ Ordenar documentos de acordo com **coseno**(consulta, documento) em ordem crescente

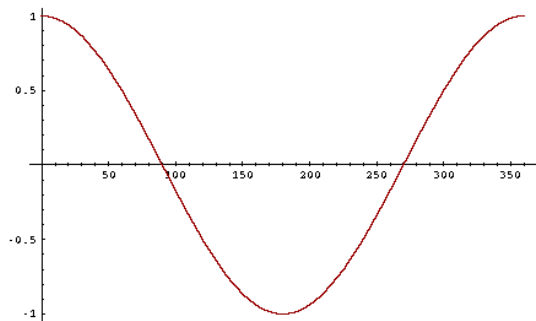
# De ângulos a cosenos

- ▶ As seguintes noções são equivalentes:
  - ▶ Ordenar documentos de acordo com o **ângulo** entre consulta e documento em ordem decrescente
  - ▶ Ordenar documentos de acordo com **coseno**(consulta, documento) em ordem crescente
- ▶ Coseno é uma função monotonicamente decrescente do ângulo para o intervalo  $[0^\circ, 180^\circ]$





# Coseno



# Normalização de magnitude

# Normalização de magnitude

- ▶ Como calcular o coseno?

# Normalização de magnitude

- ▶ Como calcular o coseno?
- ▶ Um vetor pode ter magnitude normalizada a 1 com (norma  $L_2$ ):  $\vec{x} = \frac{\vec{x}}{\|\vec{x}\|}$

# Normalização de magnitude

- ▶ Como calcular o coseno?
- ▶ Um vetor pode ter magnitude normalizada a 1 com (norma  $L_2$ ):  $\vec{x} = \frac{\vec{x}}{\|\vec{x}\|}$
- ▶ Essa operação mapeia os vetores na unidade esférica ...

# Normalização de magnitude

- ▶ Como calcular o coseno?
- ▶ Um vetor pode ter magnitude normalizada a 1 com (norma  $L_2$ ):  $\vec{x} = \frac{\vec{x}}{\|\vec{x}\|}$
- ▶ Essa operação mapeia os vetores na unidade esférica ...
- ▶ Assim, documentos mais extensos ou curtos tem mesma informação

# Normalização de magnitude

- ▶ Como calcular o coseno?
- ▶ Um vetor pode ter magnitude normalizada a 1 com (norma  $L_2$ ):  $\vec{x} = \frac{\vec{x}}{\|\vec{x}\|}$
- ▶ Essa operação mapeia os vetores na unidade esférica ...
- ▶ Assim, documentos mais extensos ou curtos tem mesma informação
- ▶ Efeito nos documentos  $d$  e  $d'$  ( $d$  “dobrado”) : mesmo vetor depois da normalização



# Similaridade cosseno entre consulta e documento

## Similaridade coseno entre consulta e documento

$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{|\mathcal{V}|} q_i d_i}{\sqrt{\sum_{i=1}^{|\mathcal{V}|} q_i^2} \sqrt{\sum_{i=1}^{|\mathcal{V}|} d_i^2}}$$

- ▶  $q_i$  é o peso tf-idf do termo  $i$  na consulta.

## Similaridade coseno entre consulta e documento

$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{|\mathcal{V}|} q_i d_i}{\sqrt{\sum_{i=1}^{|\mathcal{V}|} q_i^2} \sqrt{\sum_{i=1}^{|\mathcal{V}|} d_i^2}}$$

- ▶  $q_i$  é o peso tf-idf do termo  $i$  na consulta.
- ▶  $d_i$  é o peso tf-idf do termo  $i$  no documento.

## Similaridade coseno entre consulta e documento

$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{|\mathcal{V}|} q_i d_i}{\sqrt{\sum_{i=1}^{|\mathcal{V}|} q_i^2} \sqrt{\sum_{i=1}^{|\mathcal{V}|} d_i^2}}$$

- ▶  $q_i$  é o peso tf-idf do termo  $i$  na consulta.
- ▶  $d_i$  é o peso tf-idf do termo  $i$  no documento.
- ▶  $|\vec{q}|$  e  $|\vec{d}|$  são as magnitudes de  $\vec{q}$  e  $\vec{d}$ .

## Similaridade cosseno entre consulta e documento

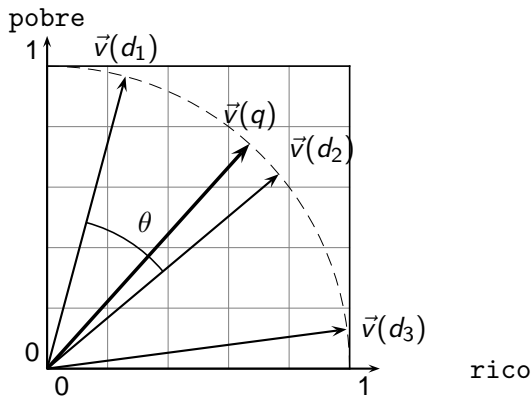
$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{|\mathcal{V}|} q_i d_i}{\sqrt{\sum_{i=1}^{|\mathcal{V}|} q_i^2} \sqrt{\sum_{i=1}^{|\mathcal{V}|} d_i^2}}$$

- ▶  $q_i$  é o peso tf-idf do termo  $i$  na consulta.
- ▶  $d_i$  é o peso tf-idf do termo  $i$  no documento.
- ▶  $|\vec{q}|$  e  $|\vec{d}|$  são as magnitudes de  $\vec{q}$  e  $\vec{d}$ .
- ▶ Esta é a similaridade **cosseno** entre  $\vec{q}$  e  $\vec{d}$  . . . . . ou, de maneira equivalente, o cosseno do ângulo entre  $\vec{q}$  e  $\vec{d}$ .

# Coseno para vetores normalizados

- ▶ Para vetores normalizados, o coseno é equivalente ao produto escalar (também conhecido como produto interno).
- ▶  $\cos(\vec{q}, \vec{d}) = \vec{q} \cdot \vec{d} = \sum_i q_i \cdot d_i$ 
  - ▶ (se  $\vec{q}$  e  $\vec{d}$  são normalizados).

## Similaridade de coseno ilustrada



## Coseno: exemplo

O quão similares  
são esses livros?

ReS: Razão e  
Sensibilidade

OeP: Orgulho e  
Preconceito

MVU: Colina dos  
Vendavais



## Coseno: exemplo

O quão similares  
são esses livros?

ReS: Razão e  
Sensibilidade

OeP: Orgulho e  
Preconceito

MVU: Colina dos  
Vendavais

frequência de termos (contagem)

termo	ReS	OeP	MVU
afeição	115	58	20
ciúmes	10	7	11
fofoca	2	0	6
vendaval	0	0	38

## Coseno: exemplo

frequência de termos (tf)

termo	ReS	OeP	MVU
afeição	115	58	20
ciúmes	10	7	11
fofoca	2	0	6
vendaval	0	0	38

## Coseno: exemplo

frequência de termos (tf)

termo	ReS	OeP	MVU
afeição	115	58	20
ciúmes	10	7	11
fofoca	2	0	6
vendaval	0	0	38

$1.0 + \log$  da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

## Coseno: exemplo

frequência de termos (tf)

termo	ReS	OeP	MVU
afeição	115	58	20
ciúmes	10	7	11
fofoca	2	0	6
vendaval	0	0	38

$1.0 + \log$  da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

Para simplificar este exemplo, não usaremos idf.

Se fosse usar, como seria o cálculo?

$$idf_t = \log \frac{N}{df_t}$$

presença de termos (df)

termo	ReS	OeP	MVU
afeição	1	1	1
ciúmes	1	1	1
fofoca	1	0	1
vendaval	0	0	1

Se fosse usar, como seria o cálculo?

$$idf_t = \log \frac{N}{df_t}$$

presença de termos (df)

idf

termo	ReS	OeP	MVU
afeição	1	1	1
ciúmes	1	1	1
fofoca	1	0	1
vendaval	0	0	1

termo	idf
afeição	$\log(3/3) = 0$
ciúmes	$\log(3/3) = 0$
fofoca	$\log(3/2) = 0.17$
vendaval	$\log(3/1) = 0.47$

## Coseno: exemplo

log da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

## Coseno: exemplo

log da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

log da frequência  
& normalização do coseno

termo	ReS	OeP	MVU
afeição	0.789	0.832	0.524
ciúmes	0.515	0.555	0.465
fofoca	0.335	0.0	0.405
vendaval	0.0	0.0	0.588



## Coseno: exemplo

log da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

log da frequência  
& normalização do coseno

termo	ReS	OeP	MVU
afeição	0.789	0.832	0.524
ciúmes	0.515	0.555	0.465
fofoca	0.335	0.0	0.405
vendaval	0.0	0.0	0.588

- ▶  $\cos(\text{ReS}, \text{OeP}) \approx$   
 $0.789 * 0.832 + 0.515 * 0.555 + 0.335 * 0.0 + 0.0 * 0.0 \approx 0.94.$

## Coseno: exemplo

log da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

log da frequência  
& normalização do coseno

termo	ReS	OeP	MVU
afeição	0.789	0.832	0.524
ciúmes	0.515	0.555	0.465
fofoca	0.335	0.0	0.405
vendaval	0.0	0.0	0.588

- ▶  $\cos(\text{ReS}, \text{OeP}) \approx 0.789 * 0.832 + 0.515 * 0.555 + 0.335 * 0.0 + 0.0 * 0.0 \approx 0.94.$
- ▶  $\cos(\text{ReS}, \text{MVU}) \approx 0.79$

## Coseno: exemplo

log da frequência

termo	ReS	OeP	MVU
afeição	3.06	2.76	2.30
ciúmes	2.0	1.85	2.04
fofoca	1.30	0	1.78
vendaval	0	0	2.58

log da frequência  
& normalização do coseno

termo	ReS	OeP	MVU
afeição	0.789	0.832	0.524
ciúmes	0.515	0.555	0.465
fofoca	0.335	0.0	0.405
vendaval	0.0	0.0	0.588

- ▶  $\cos(\text{ReS}, \text{OeP}) \approx 0.789 * 0.832 + 0.515 * 0.555 + 0.335 * 0.0 + 0.0 * 0.0 \approx 0.94.$
- ▶  $\cos(\text{ReS}, \text{MVU}) \approx 0.79$
- ▶  $\cos(\text{OeP}, \text{MVU}) \approx 0.69$

# Componentes do peso tf-idf

Frequência de termos		Frequência em Documentos		Normalização	
n (natural)	$tf_{t,d}$	n (não)	1	n (nenhum)	1
l (logaritmo)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosseno)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (aumentado)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivotado único)	$1/u$
b (booleano)	$\begin{cases} 1 & \text{se } tf_{t,d} > 0 \\ 0 & \text{senão} \end{cases}$				
L (log média)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{media}_{t \in d}(tf_{t,d}))}$				

# Componentes do peso tf-idf

Frequência de termos		Frequência em Documentos		Normalização	
n (natural)	$tf_{t,d}$	n (não)	1	n (nenhum)	1
l (logaritmo)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosseno)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (aumentado)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivotado único)	$1/u$
b (booleano)	$\begin{cases} 1 & \text{se } tf_{t,d} > 0 \\ 0 & \text{senão} \end{cases}$				
L (log média)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{media}_{t \in d}(tf_{t,d}))}$				

Melhor combinação conhecida de opções de pesos

# Componentes do peso tf-idf

Frequência de termos		Frequência em Documentos		Normalização	
n (natural)	$tf_{t,d}$	n (não)	1	n (nenhum)	1
l (logaritmo)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosseno)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (aumentado)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivotado único)	$1/u$
b (booleano)	$\begin{cases} 1 & \text{se } tf_{t,d} > 0 \\ 0 & \text{senão} \end{cases}$				
L (log média)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{media}_{t \in d}(tf_{t,d}))}$				

Padrão: sem peso

# Exemplo tf-idf

## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.



## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq

# Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq
- ▶ Exemplo: Inc.ltn

## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq
- ▶ Exemplo: Inc.ltn
- ▶ documento: log tf, sem peso df, normalização coseno

## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq
- ▶ Exemplo: Inc.ltn
- ▶ documento: log tf, sem peso df, normalização coseno
- ▶ consulta: log tf, idf, sem normalização

## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq
- ▶ Exemplo: Inc.ltn
- ▶ documento: log tf, sem peso df, normalização coseno
- ▶ consulta: log tf, idf, sem normalização
- ▶ **É ruim não colocar peso idf no documento?**

## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq
- ▶ Exemplo: Inc.ltn
- ▶ documento: log tf, sem peso df, normalização coseno
- ▶ consulta: log tf, idf, sem normalização
- ▶ **É ruim não colocar peso idf no documento?**
- ▶ Exemplo consulta: “melhor seguro carro”

## Exemplo tf-idf

- ▶ Frequentemente utiliza-se **diferentes opções de pesos** para consultas e documentos.
- ▶ Notação: ddd.qqq
- ▶ Exemplo: Inc.ltn
- ▶ documento: log tf, sem peso df, normalização coseno
- ▶ consulta: log tf, idf, sem normalização
- ▶ **É ruim não colocar peso idf no documento?**
- ▶ Exemplo consulta: “melhor seguro carro”
- ▶ Exemplo documento: “melhor seguro carro auto”

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto										
melhor										
carro										
seguro										

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento



## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0									
melhor	1									
carro	1									
seguro	1									

Colunas: **tf**: (sem peso) frequência de termo , **tf-com-peso**: log frequência de termo , **df**: frequência de documento , **idf**: frequência de documento inversa, **peso**: o peso final do termo na consulta ou documento, **norm.**: pesos de documentos depois de normalização , **produto**: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0					1				
melhor	1					0				
carro	1					1				
seguro	1					2				

Colunas: **tf:** (sem peso) frequência de termo , **tf-com-peso:** log frequência de termo , **df:** frequência de documento , **idf:** frequência de documento inversa, **peso:** o peso final do termo na consulta ou documento, **norm.:** pesos de documentos depois de normalização , **produto:** produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0				1				
melhor	1	1				0				
carro	1	1				1				
seguro	1	1				2				

Colunas: tf: (sem peso) frequência de termo , **tf-com-peso: log frequência de termo** , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0				1	1			
melhor	1	1				0	0			
carro	1	1				1	1			
seguro	1	1				2	1.3			

Colunas: tf: (sem peso) frequência de termo , **tf-com-peso: log frequência de termo** , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000			1	1			
melhor	1	1	50000			0	0			
carro	1	1	10000			1	1			
seguro	1	1	1000			2	1.3			

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3		1	1			
melhor	1	1	50000	1.3		0	0			
carro	1	1	10000	2.0		1	1			
seguro	1	1	1000	3.0		2	1.3			

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , **idf: frequência de documento inversa**, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3	0	1	1			
melhor	1	1	50000	1.3	1.3	0	0			
carro	1	1	10000	2.0	2.0	1	1			
seguro	1	1	1000	3.0	3.0	2	1.3			

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, **peso: o peso final do termo na consulta ou documento**, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3	0	1	1			
melhor	1	1	50000	1.3	1.3	0	0			
carro	1	1	10000	2.0	2.0	1	1			
seguro	1	1	1000	3.0	3.0	2	1.3			

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, **peso: o peso final do termo na consulta ou documento**, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento



## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3	0	1	1	1		
melhor	1	1	50000	1.3	1.3	0	0	0		
carro	1	1	10000	2.0	2.0	1	1	1		
seguro	1	1	1000	3.0	3.0	2	1.3	1.3		

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, **peso: o peso final do termo na consulta ou documento**, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3	0	1	1	1	0.52	
melhor	1	1	50000	1.3	1.3	0	0	0	0	
carro	1	1	10000	2.0	2.0	1	1	1	0.52	
seguro	1	1	1000	3.0	3.0	2	1.3	1.3	0.68	

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, **norm.:** pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

$$\sqrt{1^2 + 0^2 + 1^2 + 1.3^2} \approx 1.92$$

$$1/1.92 \approx 0.52$$

$$1.3/1.92 \approx 0.68$$

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3	0	1	1	1	0.52	0
melhor	1	1	50000	1.3	1.3	0	0	0	0	0
carro	1	1	10000	2.0	2.0	1	1	1	0.52	1.04
seguro	1	1	1000	3.0	3.0	2	1.3	1.3	0.68	2.04

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , **produto**: produto do peso final da consulta e peso final do documento

## Exemplo tf-idf : Inc.ltn

Consulta: "melhor seguro carro". Documento: "carro seguro auto seguro".

palavra	consulta					documento				produto
	tf	tf-com-peso	df	idf	peso	tf	tf-com-peso	peso	norm.	
auto	0	0	5000	2.3	0	1	1	1	0.52	0
melhor	1	1	50000	1.3	1.3	0	0	0	0	0
carro	1	1	10000	2.0	2.0	1	1	1	0.52	1.04
seguro	1	1	1000	3.0	3.0	2	1.3	1.3	0.68	2.04

Colunas: tf: (sem peso) frequência de termo , tf-com-peso: log frequência de termo , df: frequência de documento , idf: frequência de documento inversa, peso: o peso final do termo na consulta ou documento, norm.: pesos de documentos depois de normalização , produto: produto do peso final da consulta e peso final do documento

Resultado final de similaridade entre consulta e documento:

$$\sum_i w_{qi} \cdot w_{di} = 0 + 0 + 1.04 + 2.04 = 3.08$$

## Computando a pontuação cosseno

PONTUACAO COSSENO( $q$ )

```
1 float Pontuacao[N] = 0 // pontuacao de cada documento
2 float Tamanho[N] // tamanho de cada documento
3 for each termo consulta  $t$ 
4 do calcular  $w_{t,q}$  e obter lista de referências para  $t$ 
5   for each par( $d, tf_{t,d}$ ) na lista de referências
6     do  $Pontuacao[d] + = w_{t,d} \times w_{t,q}$ 
7   for each  $d$ 
8     do // normalização
9        $Pontuacao[d] = Pontuacao[d] / Tamanho[d]$ 
10  return Top  $K$  componentes da  $Pontuacao[]$ 
```

Resumo: recuperação ordenada no modelo de espaço vetorial

## Resumo: recuperação ordenada no modelo de espaço vetorial

- ▶ Representar a consulta como um vetor tf-idf com pesos

## Resumo: recuperação ordenada no modelo de espaço vetorial

- ▶ Representar a consulta como um vetor tf-idf com pesos
- ▶ Representar cada documento como um vetor tf-idf com pesos



## Resumo: recuperação ordenada no modelo de espaço vetorial

- ▶ Representar a consulta como um vetor tf-idf com pesos
- ▶ Representar cada documento como um vetor tf-idf com pesos
- ▶ Calcular a similaridade coseno entre o vetor consulta e vetor documento

## Resumo: recuperação ordenada no modelo de espaço vetorial

- ▶ Representar a consulta como um vetor tf-idf com pesos
- ▶ Representar cada documento como um vetor tf-idf com pesos
- ▶ Calcular a similaridade cosseno entre o vetor consulta e vetor documento
- ▶ Ranquear documentos em relação à consulta

## Resumo: recuperação ordenada no modelo de espaço vetorial

- ▶ Representar a consulta como um vetor tf-idf com pesos
- ▶ Representar cada documento como um vetor tf-idf com pesos
- ▶ Calcular a similaridade coseno entre o vetor consulta e vetor documento
- ▶ Ranquear documentos em relação à consulta
- ▶ Exibir os  $K$  melhores resultados (e.g.,  $K = 10$ ) ao usuário