

Recuperação da Informação

Esdras Lins Bispo Jr.
bispojr@ufg.br

Inteligência Artificial
Bacharelado em Ciência da Computação

24 de outubro de 2016

Plano de Aula

1 Pensamento

2 Recuperação da Informação - RI

- Matriz de incidência termo-documento
- Índice Invertido
- Processamento de consultas em RI

Sumário

1 Pensamento

2 Recuperação da Informação - RI

- Matriz de incidência termo-documento
- Índice Invertido
- Processamento de consultas em RI

Pensamento



Pensamento



Frase

Nós somos aquilo que fazemos repetidamente. Excelência, então, não é um modo de agir, mas um hábito.

Quem?

Aristóteles (384 a.C. - 322 a.C.)
Filósofo e lógico grego.

Sumário

1 Pensamento

2 Recuperação da Informação - RI

- Matriz de incidência termo-documento
- Índice Invertido
- Processamento de consultas em RI

Introdução à RI e Busca Web

O que é RI

Recuperação da Informação (*Information Retrieval*) é a atividade de encontrar material (normalmente documentos) de natureza não-estruturada (normalmente textos) que satisfaz uma necessidade de informação a partir de grandes coleções (normalmente armazenadas em computadores).

Introdução à RI e Busca Web

O que é RI

Recuperação da Informação (*Information Retrieval*) é a atividade de encontrar material (normalmente documentos) de natureza não-estruturada (normalmente textos) que satisfaz uma necessidade de informação a partir de grandes coleções (normalmente armazenadas em computadores).

Aplicações

Associamos RI diretamente à busca web, mas existem outras aplicações:

- Busca por emails;
- Busca por arquivos no seu PC;
- Recuperação de informações legais.

RI *versus* Banco de Dados

Dados estruturados

Tendem a referir informações através de tabelas:

Empregado	Gerente	Salário
Smith	Jones	R\$ 50.000,00
Chang	Smith	R\$ 60.000,00
Ivy	Smith	R\$ 50.000,00

RI *versus* Banco de Dados

Dados estruturados

Tendem a referir informações através de tabelas:

Empregado	Gerente	Salário
Smith	Jones	R\$ 50.000,00
Chang	Smith	R\$ 60.000,00
Ivy	Smith	R\$ 50.000,00

Características...

Normalmente é permitido realizar consultas exatas (através de texto)

RI *versus* Banco de Dados

Dados estruturados

Tendem a referir informações através de tabelas:

Empregado	Gerente	Salário
Smith	Jones	R\$ 50.000,00
Chang	Smith	R\$ 60.000,00
Ivy	Smith	R\$ 50.000,00

Características...

Normalmente é permitido realizar consultas exatas (através de texto)

Exemplo: Salário < 60000 AND Gerente = Smith

RI *versus* Banco de Dados

Dados não-estruturados

- Normalmente refere-se a textos livres;
- Permite consultas por palavras-chave (incluindo operadores);
- Modelo clássico de busca por documentos de texto.

Pressupostos básicos em RI

Coleção

Um conjunto de documentos
(assumimos ser estático, neste momento).

Pressupostos básicos em RI

Coleção

Um conjunto de documentos
(assumimos ser estático, neste momento).

Objetivo

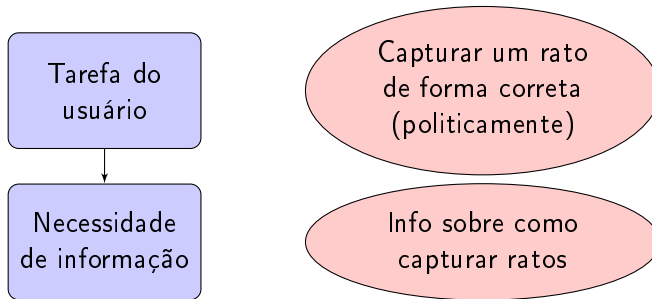
Recuperar documentos com informação que é relevante para **as necessidades de informação** do usuário e ajudá-lo a completar uma **tarefa**.

Teste

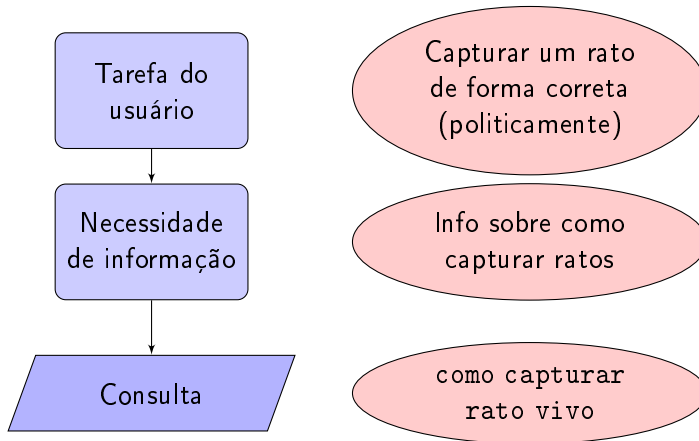
Tarefa do
usuário

Capturar um rato
de forma correta
(politicamente)

Teste



Teste

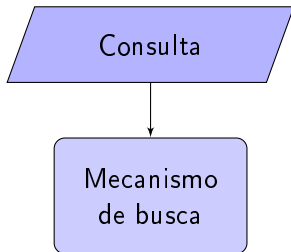


Teste

Consulta

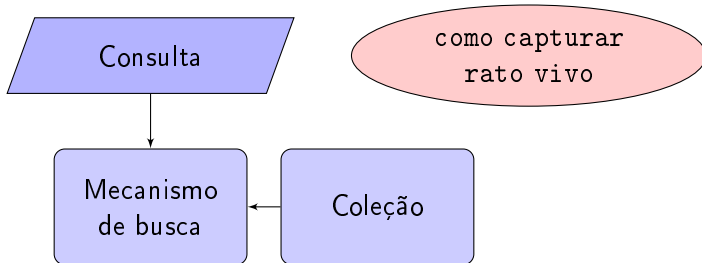
como capturar
rato vivo

Teste

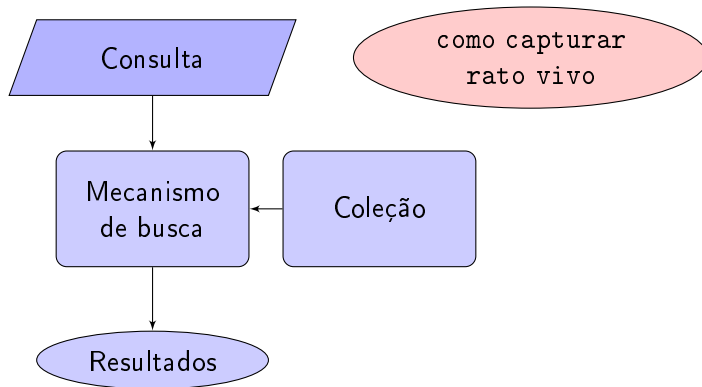


como capturar
rato vivo

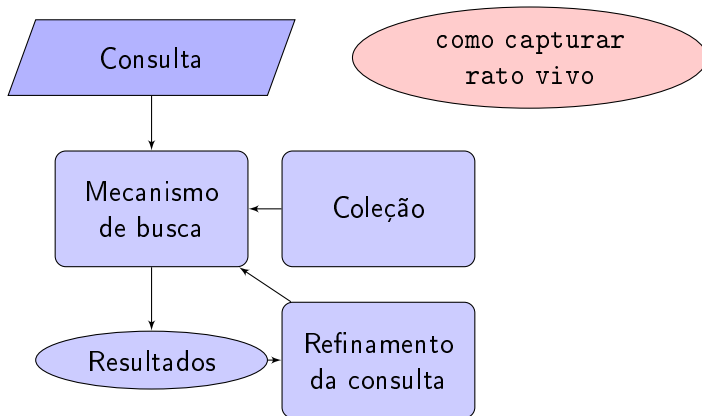
Teste



Teste



Teste



Dados não-estruturados em 1620

Obras de Shakespeare

- Quais peças de Shakespeare contêm as palavras 'Brutus' e 'Caesar', mas não 'Capurnia'?

Dados não-estruturados em 1620

Obras de Shakespeare

- Quais peças de Shakespeare contêm as palavras 'Brutus' e 'Caesar', mas não 'Capurnia'?
- Poderíamos fazer um `grep all` das peças de Shakespeare para 'Brutus' e 'Caesar', e daí retirar as linhas que contêm 'Calpurnia'?



Dados não-estruturados em 1620

Obras de Shakespeare

- Quais peças de Shakespeare contêm as palavras 'Brutus' e 'Caesar', mas não 'Capurnia'?
- Poderíamos fazer um `grep all` das peças de Shakespeare para 'Brutus' e 'Caesar', e daí retirar as linhas que contêm 'Calpurnia'?
- Por que não deveríamos fazer isto?



Dados não-estruturados em 1620

Obras de Shakespeare

- Quais peças de Shakespeare contêm as palavras 'Brutus' e 'Caesar', mas não 'Capurnia'?
- Poderíamos fazer um `grep all` das peças de Shakespeare para 'Brutus' e 'Caesar', e daí retirar as linhas que contêm 'Calpurnia'?
- Por que não deveríamos fazer isto?
 - Lento (para coleções grandes);

Dados não-estruturados em 1620

Obras de Shakespeare

- Quais peças de Shakespeare contêm as palavras 'Brutus' e 'Caesar', mas não 'Capurnia'?
- Poderíamos fazer um `grep all` das peças de Shakespeare para 'Brutus' e 'Caesar', e daí retirar as linhas que contêm 'Calpurnia'?
- Por que não deveríamos fazer isto?
 - Lento (para coleções grandes);
 - NOT 'Calpurnia' não é trivial;

Dados não-estruturados em 1620

Obras de Shakespeare

- Quais peças de Shakespeare contêm as palavras 'Brutus' e 'Caesar', mas não 'Capurnia'?
- Poderíamos fazer um `grep all` das peças de Shakespeare para 'Brutus' e 'Caesar', e daí retirar as linhas que contêm 'Calpurnia'?
- Por que não deveríamos fazer isto?
 - Lento (para coleções grandes);
 - NOT 'Calpurnia' não é trivial;
 - Outras operações não são viáveis (e.g. encontrar a palavra 'Romans' próximo de 'countrymen').

Matriz incidência termo-documento

	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	
...							

► **Figure 1.1** A term-document incidence matrix. Matrix element (t, d) is 1 if the play in column d contains the word in row t , and is 0 otherwise.

Matriz incidência termo-documento

	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	
...							

► **Figure 1.1** A term-document incidence matrix. Matrix element (t, d) is 1 if the play in column d contains the word in row t , and is 0 otherwise.

Pergunta...

Como fazer a consulta: Brutus AND Caesar BUT NOT Calpurnia ?

Vetores de incidência

Uma solução...

- Temos vetores 0/1 para cada termo;

Vetores de incidência

Uma solução...

- Temos vetores 0/1 para cada termo;
- Obtenha os vetores
Brutus, **Caesar** e **Calpurnia** (seu complemento):

Vetores de incidência

Uma solução...

- Temos vetores 0/1 para cada termo;
- Obtenha os vetores
Brutus, **Caesar** e **Calpurnia** (seu complemento):
 - Realize a operação binária AND entre os vetores:

Vetores de incidência

Uma solução...

- Temos vetores 0/1 para cada termo;
- Obtenha os vetores
Brutus, **Caesar** e **Calpurnia** (seu complemento):
 - Realize a operação binária AND entre os vetores:
 $110100 \text{ AND } 110111 \text{ AND } 101111 = 100100$.

Respostas à consulta

Antony and Cleopatra, Act III, Scene ii

Agrippa [Aside to Domitius Enobarbus]: Why, Enobarbus,
 When Antony found Julius Caesar dead,
 He cried almost to roaring; and he wept
 When at Philippi he found Brutus slain.

Hamlet, Act III, Scene ii

Lord Polonius: I did enact Julius Caesar: I was killed i' the
 Capitol; Brutus killed me.

► **Figure 1.2** Results from Shakespeare for the query Brutus AND Caesar AND NOT Calpurnia.

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;
- Suponha que haja 500K termos distintos entre si;

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;
- Suponha que haja 500K termos distintos entre si;
- $500K \times 1M$ tem meio trilhão de 0's e 1's!!!

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;
- Suponha que haja 500K termos distintos entre si;
- $500K \times 1M$ tem meio trilhão de 0's e 1's!!!
- Mas não mais que um bilhão de 1's (**Por quê**)?

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;
- Suponha que haja 500K termos distintos entre si;
- $500K \times 1M$ tem meio trilhão de 0's e 1's!!!
- Mas não mais que um bilhão de 1's (**Por quê**)?
- Esta matriz é extremamente esparsa;

Coleções muito grandes

Problema...

- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;
- Suponha que haja 500K termos distintos entre si;
- $500K \times 1M$ tem meio trilhão de 0's e 1's!!!
- Mas não mais que um bilhão de 1's (**Por quê?**)?
- Esta matriz é extremamente esparsa;

Qual a melhor representação?

Guardar apenas as células com 1's.



Índice Invertido

Brutus →

1	2	4	11	31	45	173	174
---	---	---	----	----	----	-----	-----

Caesar →

1	2	4	5	6	16	57	132	...
---	---	---	---	---	----	----	-----	-----

Calpurnia →

2	31	54	101
---	----	----	-----

Como fazer...

- Para cada termo t , devemos armazenar uma lista de todos os documentos que contêm t ;

Índice Invertido

Brutus →

1	2	4	11	31	45	173	174
---	---	---	----	----	----	-----	-----

Caesar →

1	2	4	5	6	16	57	132	...
---	---	---	---	---	----	----	-----	-----

Calpurnia →

2	31	54	101
---	----	----	-----

Como fazer...

- Para cada termo t , devemos armazenar uma lista de todos os documentos que contêm t ;
- Identifique cada documento por um docID (um identificador único do documento);



Índice Invertido

Brutus →

1	2	4	11	31	45	173	174
---	---	---	----	----	----	-----	-----

Caesar →

1	2	4	5	6	16	57	132	...
---	---	---	---	---	----	----	-----	-----

Calpurnia →

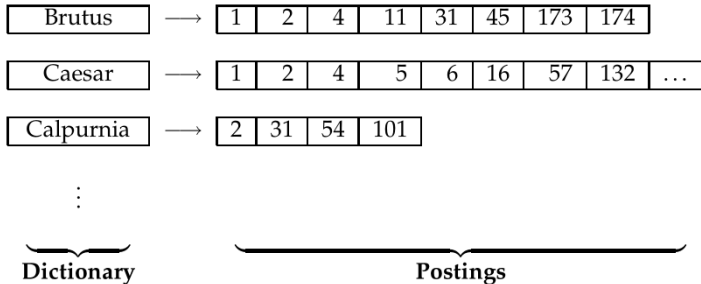
2	31	54	101
---	----	----	-----

Como fazer...

- Para cada termo t , devemos armazenar uma lista de todos os documentos que contêm t ;
- Identifique cada documento por um docID (um identificador único do documento);
- Poderíamos utilizar vetores de tamanho fixo neste caso?



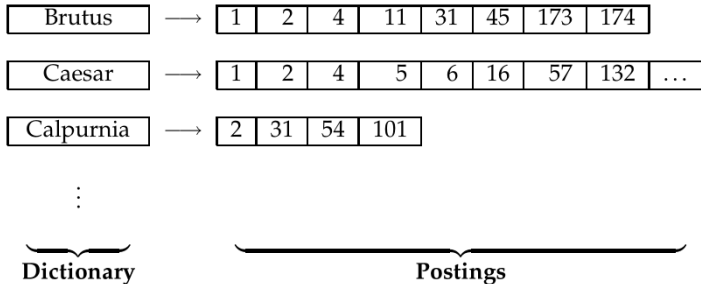
Índice Invertido



Como fazer...

- Precisamos criar uma lista de *postings*;

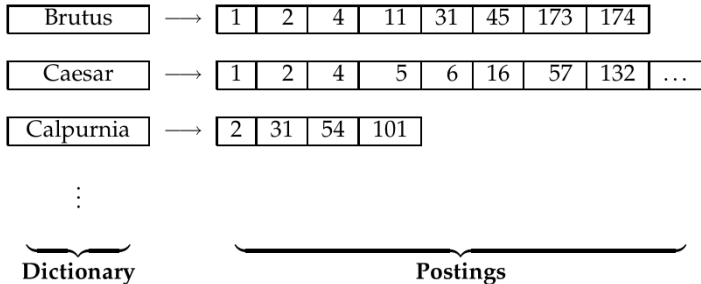
Índice Invertido



Como fazer...

- Precisamos criar uma lista de *postings*;
 - Em disco, o armazenamento contíguo é melhor e mais comum;

Índice Invertido



Como fazer...

- Precisamos criar uma lista de *postings*;
 - Em disco, o armazenamento contíguo é melhor e mais comum;
 - Em memória, podemos utilizar listas ligadas ou vetores de tamanho variável!

Índice invertido

Passo-a-passo..

- 1 Colete os documentos a serem indexados:

Friends, Romans, countrymen. So let it be with Caesar ...



UFMG
Campus Jataí

Índice invertido

Passo-a-passo..

- 1 Colete os documentos a serem indexados:

Friends, Romans, countrymen. So let it be with Caesar ...

- 2 Quebre o texto, transformando cada documento em uma lista de *tokens*:

Friends Romans countrymen So ...

Índice invertido

Passo-a-passo..

- 1 Colete os documentos a serem indexados:

Friends, Romans, countrymen. So let it be with Caesar ...

- 2 Quebre o texto, transformando cada documento em uma lista de *tokens*:

Friends Romans countrymen So ...

- 3 Faça o pré-processamento linguístico, produzindo uma lista de *tokens* normalizados:

friend roman countryman so ...

Índice invertido

Passo-a-passo..

- 1 Colete os documentos a serem indexados:

Friends, Romans, countrymen. So let it be with Caesar ...

- 2 Quebre o texto, transformando cada documento em uma lista de *tokens*:

Friends Romans countrymen So ...

- 3 Faça o pré-processamento linguístico, produzindo uma lista de *tokens* normalizados:

friend roman countryman so ...

- 4 Indexe os documentos a partir da ocorrência dos termos, criando um índice invertido, consistindo de um dicionário e *postings*.

Índice Invertido

Doc 1

I did enact Julius Caesar: I was killed
i' the Capitol; Brutus killed me.

term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2

Doc 2

So let it be with Caesar. The noble Brutus
hath told you Caesar was ambitious:

term	docID
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Índice Invertido

term	docID	term	docID		term	docID	term	docID
I	1	let	2		ambitious	2	julius	1
did	1	it	2		be	2	killed	1
enact	1	be	2		brutus	1	killed	1
julius	1	with	2		brutus	2	let	2
caesar	1	caesar	2		capitol	1	me	1
I	1	the	2	⇒	caesar	1	noble	2
was	1	noble	2		caesar	2	so	2
killed	1	brutus	2		caesar	2	the	1
i'	1	hath	2		did	1	the	2
the	1	told	2		enact	1	told	2
capitol	1	you	2		hath	1	you	2
brutus	1	caesar	2		I	1	was	1
killed	1	was	2		I	1	was	2
me	1	ambitious	2		i'	1	with	2
so	2				it	2		

Índice Invertido

term	docID	term	docID
ambitious	2	julius	1
be	2	killed	1
brutus	1	killed	1
brutus	2	let	2
capitol	1	me	1
caesar	1	noble	2
caesar	2	so	2
caesar	2	the	1
did	1	the	2
enact	1	told	2
hath	1	you	2
I	1	was	1
I	1	was	2
i'	1	with	2
it	2		

⇒

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2

term	doc. freq.	→	postings lists
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Construímos apenas o índice...

- O que fazemos para processar uma consulta?

Construímos apenas o índice...

- O que fazemos para processar uma consulta?
- Quais tipos de consultas nós podemos processar?

Processamento de consultas: AND

Brutus AND Caesar

- Localize Brutus no dicionário → recupere os seus *postings*;

Processamento de consultas: AND

Brutus AND Caesar

- Localize Brutus no dicionário → recupere os seus *postings*;
- Localize Caesar no dicionário → recupere os seus *postings*;

Processamento de consultas: AND

Brutus AND Caesar

- Localize Brutus no dicionário → recupere os seus *postings*;
- Localize Caesar no dicionário → recupere os seus *postings*;
- “Funda” as duas listas de *postings* →
faça a interseção dos conjuntos de documentos;

Processamento de consultas: AND

Brutus AND Caesar

- Localize Brutus no dicionário → recupere os seus *postings*;
- Localize Caesar no dicionário → recupere os seus *postings*;
- “Funda” as duas listas de *postings* →
faça a interseção dos conjuntos de documentos;

Brutus →

1	2	4	11	31	45	173	174
---	---	---	----	----	----	-----	-----

Caesar →

1	2	4	5	6	16	57	132	...
---	---	---	---	---	----	----	-----	-----

A fusão (*merge*)

Caminhe através das duas listas em tempo linear em relação ao tamanho das listas.

Brutus →

1	2	4	11	31	45	173	174
---	---	---	----	----	----	-----	-----

Caesar →

1	2	4	5	6	16	57	132	...
---	---	---	---	---	----	----	-----	-----

A fusão (*merge*)

Caminhe através das duas listas em tempo linear em relação ao tamanho das listas.

Brutus →

1	2	4	11	31	45	173	174
---	---	---	----	----	----	-----	-----

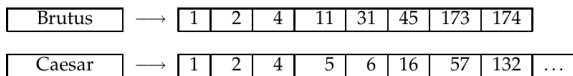
Caesar →

1	2	4	5	6	16	57	132	...
---	---	---	---	---	----	----	-----	-----

Se os tamanhos das listas forem x e y ,
a fusão levará $O(x + y)$ operações.

A fusão (*merge*)

Caminhe através das duas listas em tempo linear em relação ao tamanho das listas.



Se os tamanhos das listas forem x e y ,
a fusão levará $O(x + y)$ operações.

Importante!!!

As listas de *postings* precisam estar ordenadas pelo docID.

Algoritmo de fusão

```
INTERSECT( $p_1, p_2$ )  
1   $answer \leftarrow \langle \rangle$   
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$   
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$   
4      then  $\text{ADD}(answer, \text{docID}(p_1))$   
5           $p_1 \leftarrow \text{next}(p_1)$   
6           $p_2 \leftarrow \text{next}(p_2)$   
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$   
8          then  $p_1 \leftarrow \text{next}(p_1)$   
9          else  $p_2 \leftarrow \text{next}(p_2)$   
10 return  $answer$ 
```

Consultas por frases

- Responder consultas como “stanford university” (como uma frase);

Consultas por frases

- Responder consultas como “stanford university” (como uma frase);
- Então a sentença “I went to university at Stanford? não é uma ocorrência;



Consultas por frases

- Responder consultas como “stanford university” (como uma frase);
- Então a sentença “I went to university at Stanford? não é uma ocorrência;
- O conceito de consultas por frases tem sido facilmente compreendido pelos usuários;

Consultas por frases

- Responder consultas como “stanford university” (como uma frase);
- Então a sentença “I went to university at Stanford? não é uma ocorrência;
- O conceito de consultas por frases tem sido facilmente compreendido pelos usuários;
- Uma das poucas ideias de “busca avançada” que funciona na prática;



Consultas por frases

- Responder consultas como “stanford university” (como uma frase);
- Então a sentença “I went to university at Stanford? não é uma ocorrência;
- O conceito de consultas por frases tem sido facilmente compreendido pelos usuários;
- Uma das poucas ideias de “busca avançada” que funciona na prática;
- Muitas das consultas são implicitamente consultas por frases;



Consultas por frases

- Responder consultas como “stanford university” (como uma frase);
- Então a sentença “I went to university at Stanford? não é uma ocorrência;
- O conceito de consultas por frases tem sido facilmente compreendido pelos usuários;
- Uma das poucas ideias de “busca avançada” que funciona na prática;
- Muitas das consultas são implicitamente consultas por frases;
- Para isto, não é suficiente armazenar apenas entradas
< termo : docs >.

Consultas por frases

<*be*: 993427;
1: 7, 18, 33, 72, 86, 231;
2: 3, 149;
4: 17, 191, 291, 430, 434;
5: 363, 367, ...>

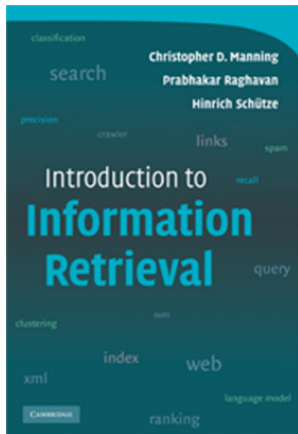
Consultas por frases

<*be*: 993427;
1: 7, 18, 33, 72, 86, 231;
2: 3, 149;
4: 17, 191, 291, 430, 434;
5: 363, 367, ...>

Pergunta?

Quais dos documentos 1, 2, 4 e 5 pode conter “be or not to be”?

Referência bibliográfica



Livro

Information Retrieval,
Stanford University,
Christopher Manning *et al.*

Link

Acesso em
<http://nlp.stanford.edu/IR-book/>

Recuperação da Informação

Esdras Lins Bispo Jr.
bispojr@ufg.br

Inteligência Artificial
Bacharelado em Ciência da Computação

24 de outubro de 2016