Recuperação da Informação

Esdras Lins Bispo Jr.
bispojr@ufg.br

Inteligência Artificial
Bacharelado em Ciência da Computação

26 de outubro de 2016

UFG
Campus Jataí

Plano de Aula

1. Revisão
   - Matriz de incidência termo-documento
2. Matriz de incidência termo-documento
   - Índice Invertido
   - Processamento de consultas em RI

UFG
Campus Jataí

Sumário

UFG
Campus Jataí

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

Características...

Normalmente é permitido realizar consultas exatas (através de texto)

Exemplo: Salário $<$ 60000 AND Gerente $=$ Smith

UFG
Campus Jataí

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.

UFG
Campus Jataí

Pressupostos básicos em RI

Coleção

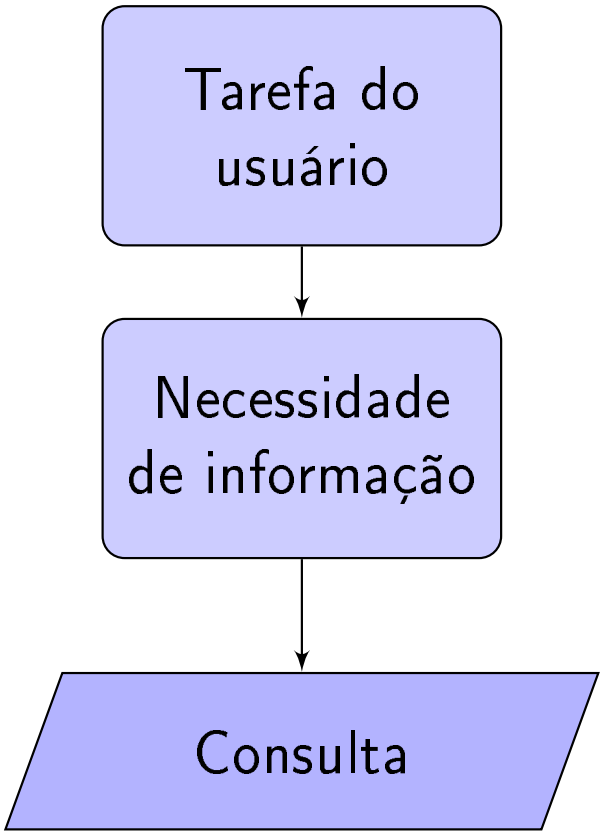
Um conjunto de documentos
(assumimos ser estático, neste momento).

Objetivo

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

UFG
Campus Jataí

Teste



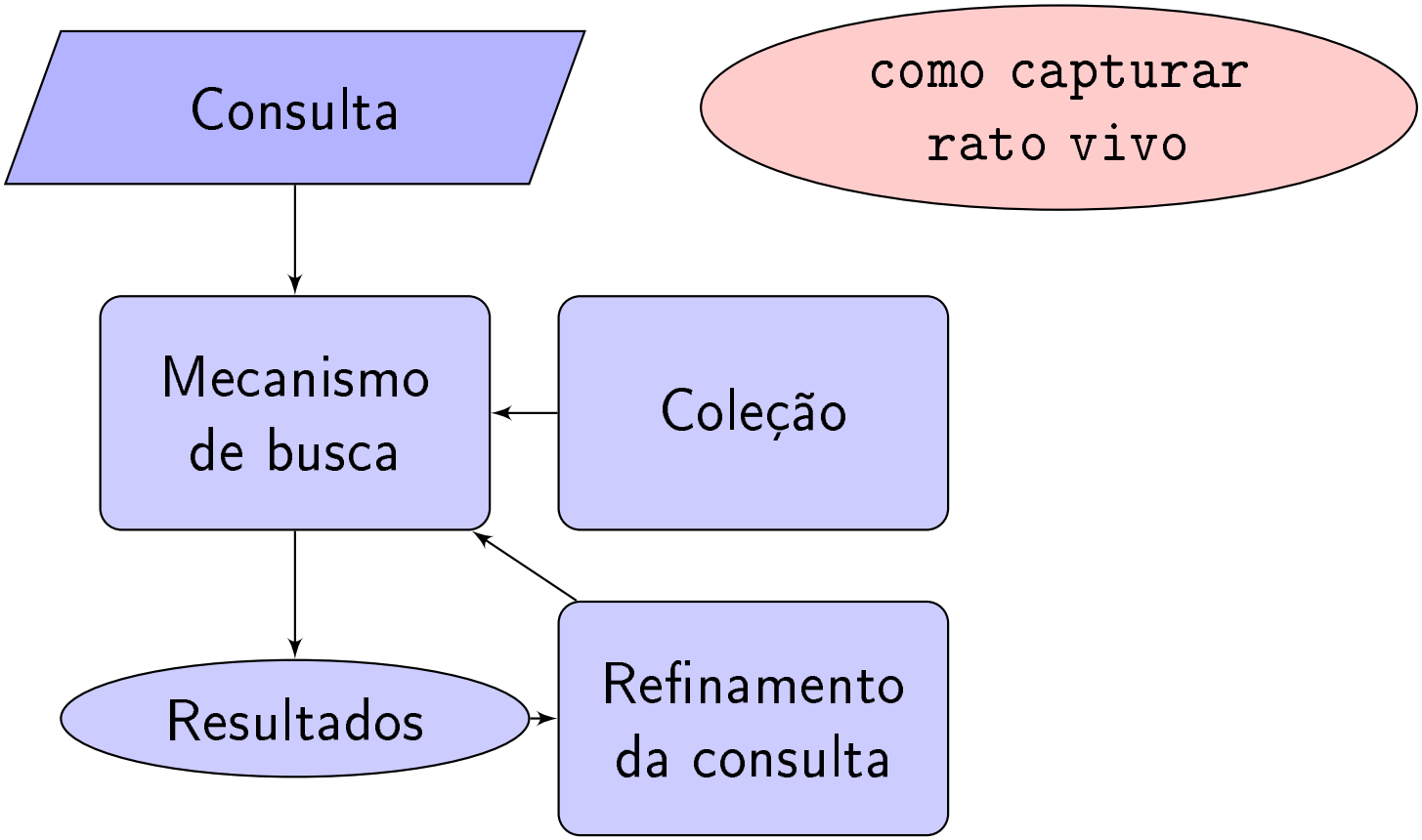
Capturar um rato de forma correta (politicamente)

Info sobre como capturar ratos

como capturar rato vivo

UFG
Campus Jataí

Teste


Consulta
como capturar rato vivo
Mecanismo de busca
Coleção
Resultados
Refinamento da consulta

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'?
- 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.

Pergunta...

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

Sumário

UFG
Campus Jataí

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):

UFG
Campus Jataí

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 AND 110111 AND 101111 = 100100.

UFG
Campus Jataí

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);

UFG
Campus Jataí

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;

UFG
Campus Jataí

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;

UFG
Campus Jataí

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!!!

UFG
Campus Jataí

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)?

UFG
Campus Jataí

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;

UFG
Campus Jataí

Coleções muito grandes

Problema...

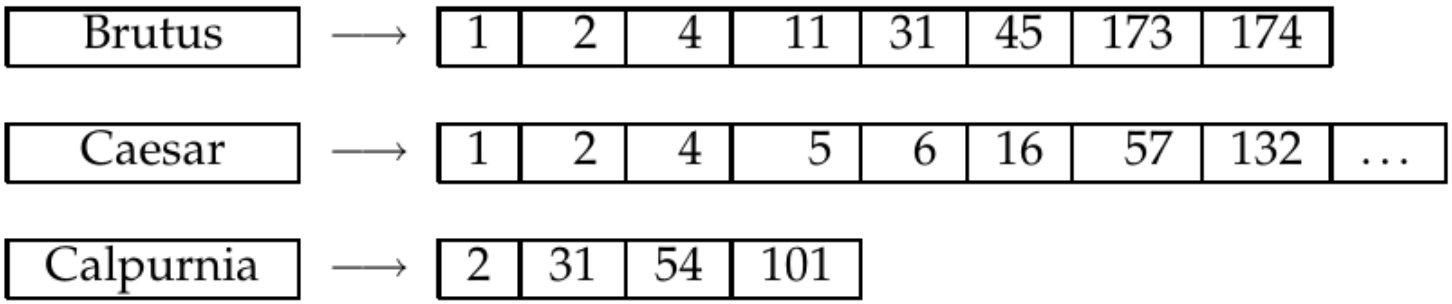
- Considere $N = 1$ milhão de documentos, cada um com 1000 palavras em média;
- Considere que cada palavra tenha, em média, 6 bytes (incluindo espaços e pontuação);
- Temos 6GB de dados em documentos;
- Suponha que haja 500K termos distintos entre si;
- 500K $\times$ 1M tem meio trilhão de 0's e 1's!!!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

UFG
Campus Jataí

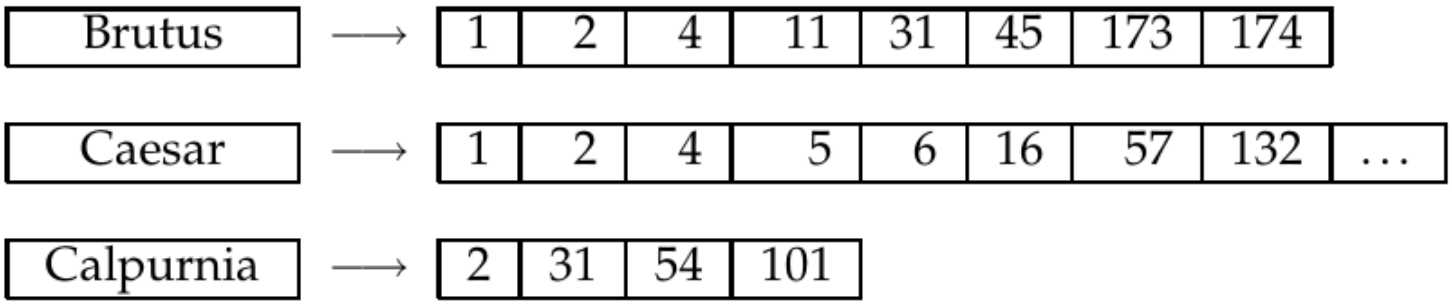
Índice Invertido



Como fazer...

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

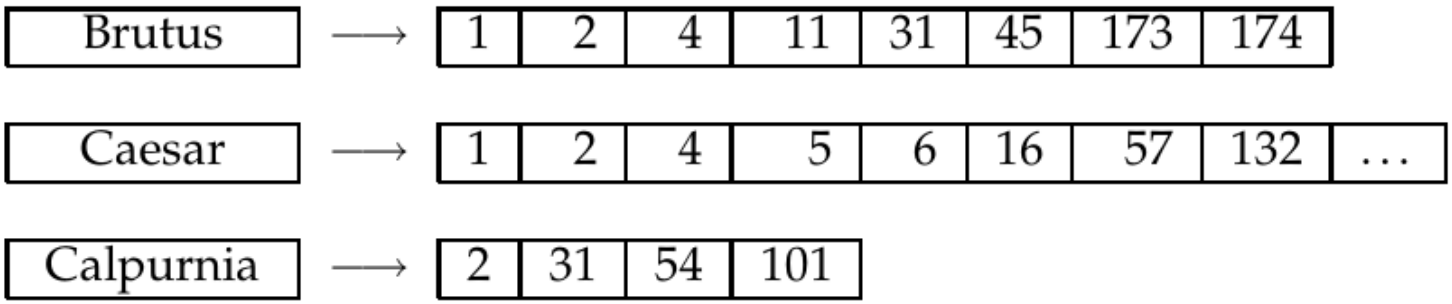
Índice Invertido



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);

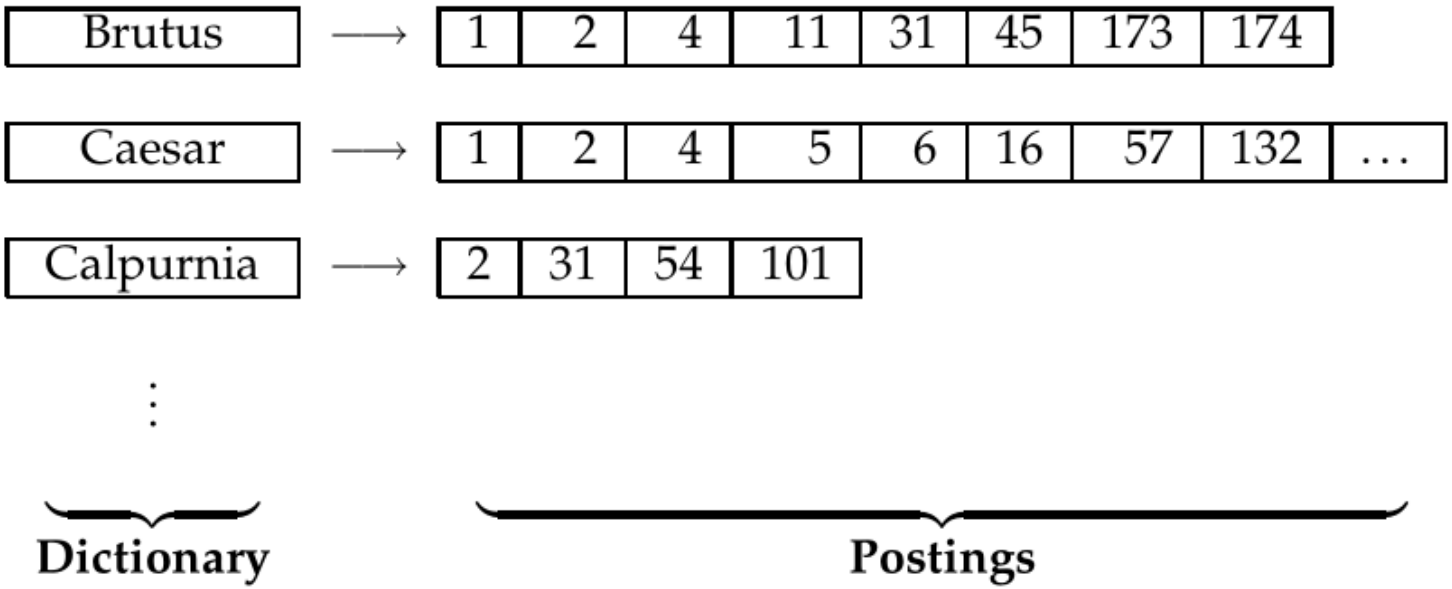
Índice Invertido



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?

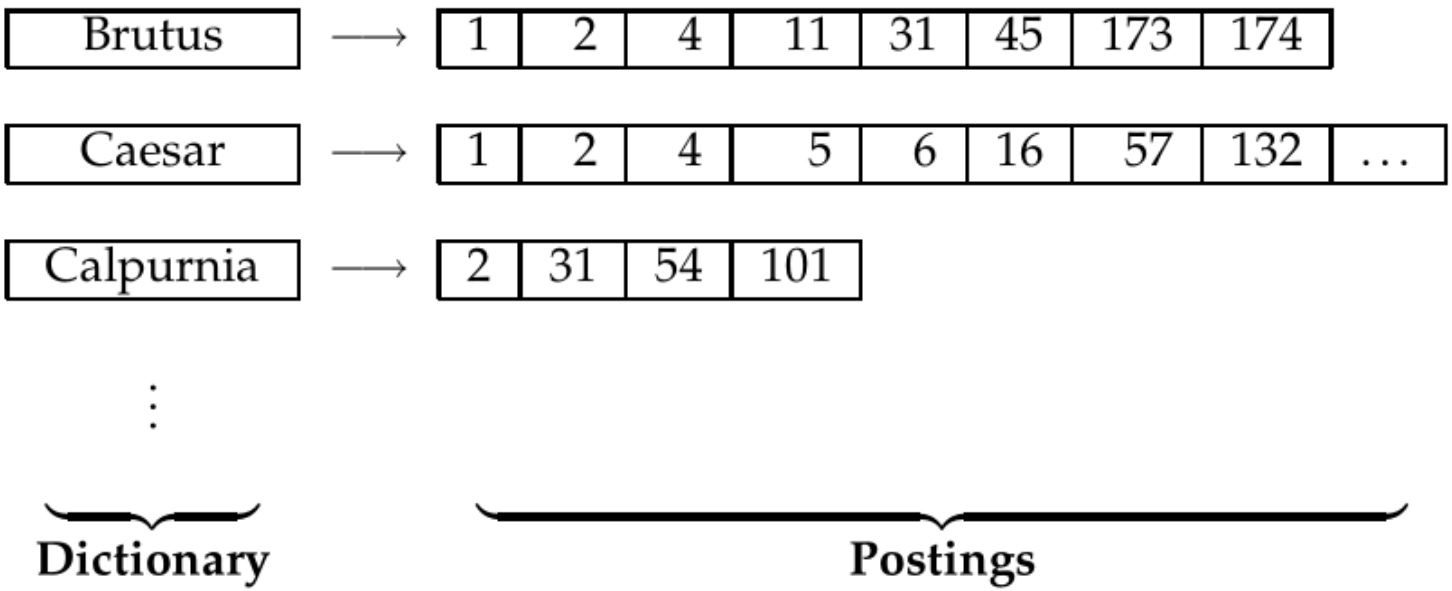
Índice Invertido



Como fazer...

- Precisamos criar uma lista de *postings*;

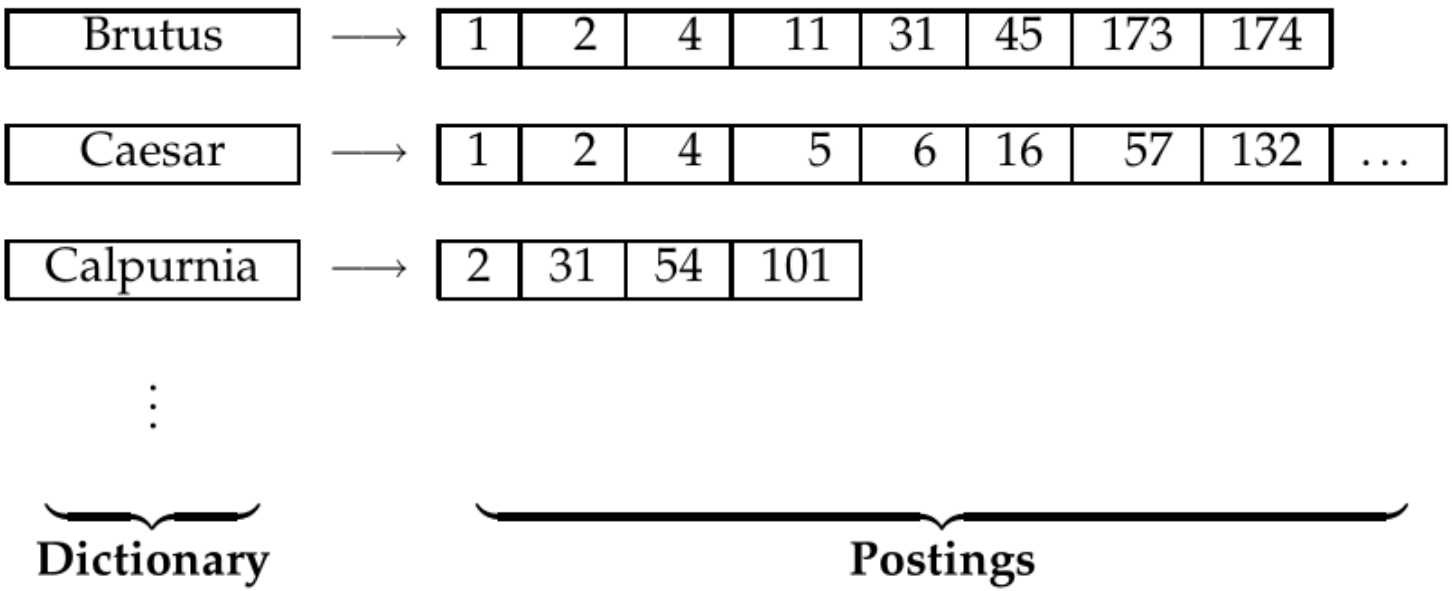
Índice Invertido



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 ...

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.

Doc 2
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:

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

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

UFG
Campus Jataí

Índice Invertido

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

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

$\Longrightarrow$

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i′	1
it	2

term	docID
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

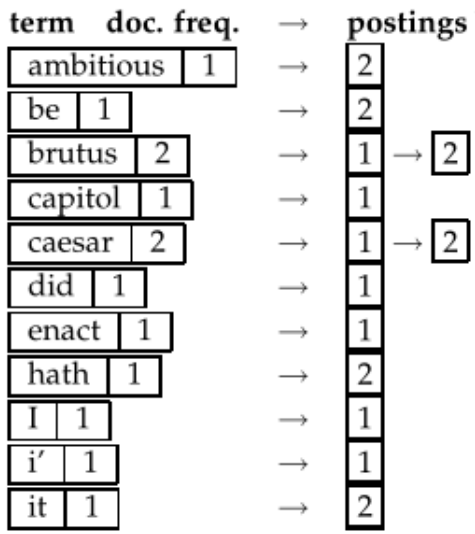
UFG
Campus Jataí

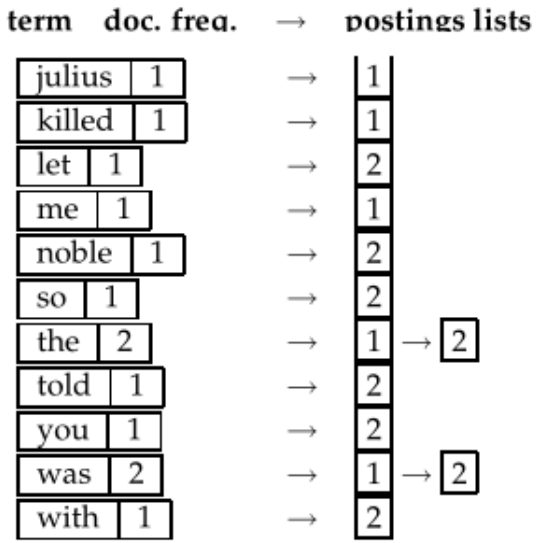
Índice Invertido

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2

term	docID
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

$\Longrightarrow$





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?

UFG
Campus Jataí

Processamento de consultas: AND

Brutus AND Caesar

- Localize Brutus no dicionário $\rightarrow$ 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

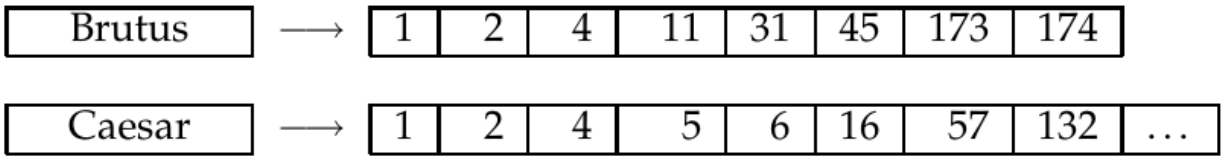
- Localize Brutus no dicionário → recupere os seus *postings*;
- Localize Caesar no dicionário → recupere os seus *postings*;
- "Funda" as duas listas de *postings* →
  faça a interseção dos conjuntos de documentos;

UFG
Campus Jataí

Processamento de consultas: AND

Brutus AND Caesar

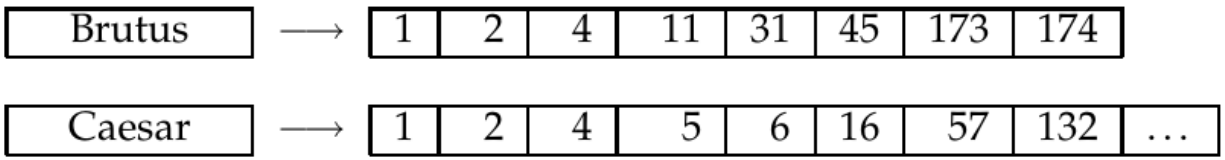
- Localize Brutus no dicionário → recupere os seus *postings*;
- Localize Caesar no dicionário → recupere os seus *postings*;
- "Funda" as duas listas de *postings* →
  faça a interseção dos conjuntos de documentos;



UFG
Campus Jataí

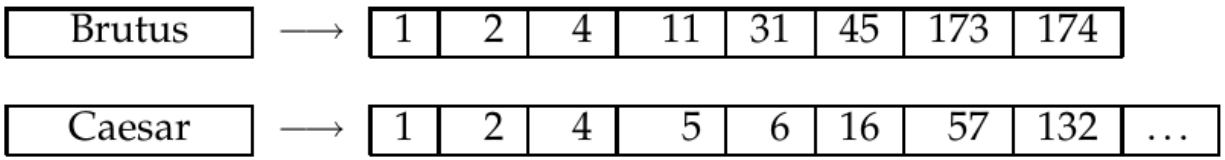
A fusão (*merge*)

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



A fusão (*merge*)

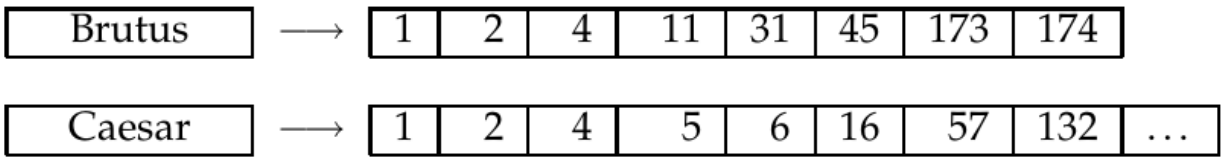
Caminhe através das duas listas em tempo linear em relação ao tamanho das listas.



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

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.

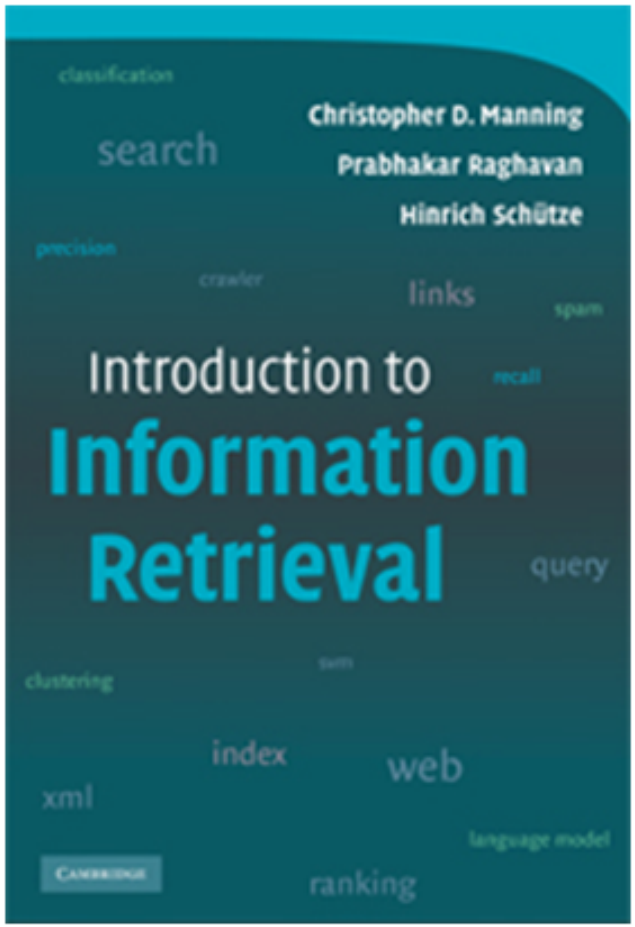
Importante!!!

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

Algoritmo de fusão

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

Referência bibliográfica



Livro

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

Link

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

UFG
Campus Jataí

Recuperação da Informação

Esdras Lins Bispo Jr.
bispojr@ufg.br

Inteligência Artificial
Bacharelado em Ciência da Computação

26 de outubro de 2016

UFG
Campus Jataí