Classificação e Pesquisa de Dados

Aulas 18-19

Organização de Arquivos:
Arquivos Invertidos, Árvores TRIE e Patricias

UFRGS INF01124

Resumo da aula

- Estudar as estruturas de **Arquivo Invertido**, Patricias e TRIE
- Conhecer suas aplicações mais comuns (índices textuais, *Search Engines*, *Recuperação de Informação*) e o processo geral de indexação de documentos

Arquivo Invertido

- Caracterização:
  - Em vez de serem coletados os valores dos atributos para cada registro, são identificados os registros que possuem um dado valor do atributo considerado
  - À cada valor de chave corresponde uma lista de endereços de registros
  - O conjunto de listas invertidas associado a uma chave de acesso é chamado ***inversão***
  - Um arquivo invertido pode possuir uma ou mais inversões
- Aplicação: índices textuais, motores de busca na web, índices secundários em BD

Arquivo Invertido

Idade	Endereços (*postings*)			
20	5	6		
22	7	11		
23	3	8		
25	1	4	13	15
26	10	12		
27	2			
28	9	14	16	

Inversões

Salário	Endereços (*postings*)				
500	1	12			
550	4	9			
600	11	3	15		
650	5	6			
700	2	10			
750	7	8	13	14	16

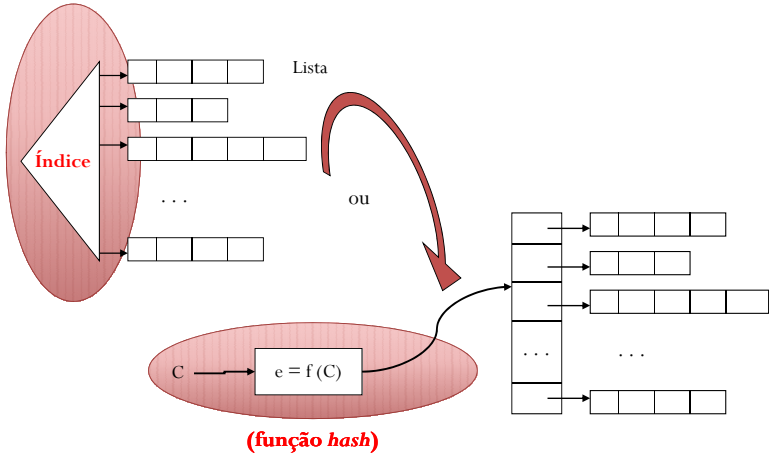
Arquivo principal (BD):

#	ID	Nome	Idade	Salário
1	1000	Ademar	25	500
2	1050	Afonso	27	700
3	2400	Iara	23	600
4	1850	Edmundo	25	550
5	1440	Cristiano	20	650
6	3150	Tatiana	20	650
7	2000	Gerson	22	750
8	1900	Ênio	23	750
9	2430	Ivan	28	550
10	2600	Miguel	26	700
11	1075	Ângela	22	600
12	1400	Cláudia	26	500
13	2200	Helena	25	750
14	2700	Ramon	28	750
15	2950	Flávio	25	600
16	3100	Sônia	28	750
:	:	:	:	:

Arquivo Invertido

- Decisões importantes:
  1. Como estruturar o acesso às listas?
  2. Como estruturar as listas?

1. Como estruturar o acesso às listas?



2. Como estruturar as listas?

- Qualquer solução estudada para representação de listas lineares:
  - **Contigüidade física** (registros de tamanho variável, normalmente);
  - **Encadeamento**;
  - **Mapa de bits** (quando a gama de valores possíveis é pequena).
- Considerar que, via de regra, tais listas são armazenadas em disco, não sendo recomendável o simples encadeamento item a item

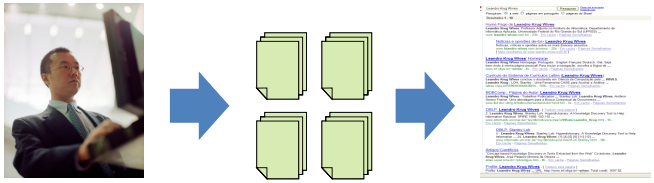
2. Como estruturar as listas?

- Importante lembrar/considerar:
  - fazer com que cada lista seja composta por uma lista encadeada de zero ou mais blocos, cada um contendo vários endereços de registros
  - Fazer com que as listas estejam ordenadas com o mesmo critério
  - Fazer com que os registros sejam identificados da mesma maneira

Aplicação-exemplo:

Information Retrieval

- *Information Retrieval* cobre atualmente qualquer forma de documento (informações semi-estruturadas, textos, vídeos, imagens, sons, seqüências de DNA, etc.)
- Vamos nos focar em:
  - Indexação de textos
  - Coleções estáticas
  - Recuperação *ad-hoc* (recuperação de informações em resposta à consultas do usuário)

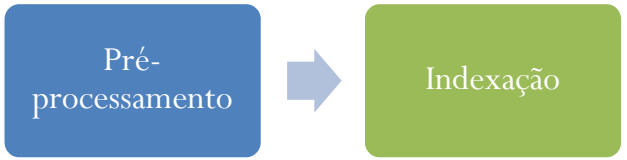


Consulta ad hoc

Coleção estática de documentos

Resultado "ranqueado"

Visão geral do processo de indexação



Pré-processamento

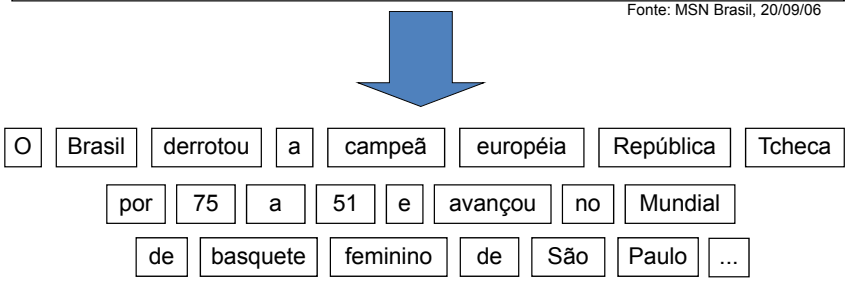
- Objetivo: extrair palavras candidatas ao índice
- Envolve:
  - Identificar termos (tokenização)
  - Normalizar termos (eliminar erros e variações morfológicas)
  - Calcular/identificar freqüências e pesos dos termos
  - Identificar termos discriminantes

Pré-processamento:

Exemplo simples de tokenização

O Brasil derrotou a campeã européia República Tcheca por 75 a 51 e avançou no Mundial de basquete feminino de São Paulo...

Fonte: MSN Brasil, 20/09/06



Cada *token* é um candidato para o índice!

Pré-processamento:

Seleção de termos discriminantes

o	Brasil	derrotou	a	campeã	européia	República
Tcheca	por	a	e	avançou	no	mundial
de	basquete	feminino	de	São	Paulo	...

Eliminação de "Stopwords"

Brasil	derrotou	campeã	européia	República	Tcheca	
avançou	mundial	basquete	feminino	São	Paulo	...

Pré-processamento:

Exemplo simples de normalização

Brasil	derrotou	campeã	européia	República	Tcheca	
avançou	mundial	basquete	feminino	São	Paulo	...

Case folding (lowercase)
Eliminação de Acentos
Minimização de erros ortográficos…

brasil	derrotou	campea	europeia	republica	tcheca	
avancou	mundial	basquete	feminino	sao	paulo	...

Pré-processamento:

Exemplo simples de *term weighting*

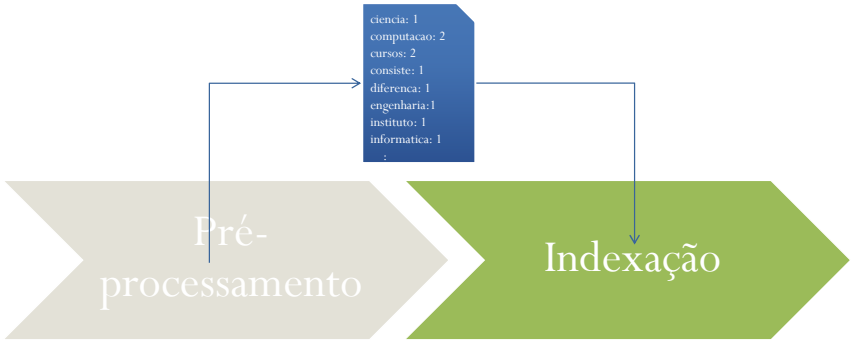
brasil	derrotou	campea	europeia	republica	tcheca	
avancou	mundial	basquete	feminino	sao	paulo	...

Análise de frequencia

brasil	3
campea	1
derrota	2
europeia	1
Republica	1
:	

Indexação

- Criação do índice (lista invertida)
- Um documento por vez (pré-processamento + indexação)



Constituição do índice

a) **Dicionário de Termos:** índice de termos indexados
b) **Lista de *postings:*** lista de documentos por termo
c) **Lista de documentos:** informações sobre documentos

Visão geral do arquivo de índice

Dicionário

Palavra	Apontador
Aluno	0100
Barraca	**0010**
Carro	0012
:	:
Zoo	0002

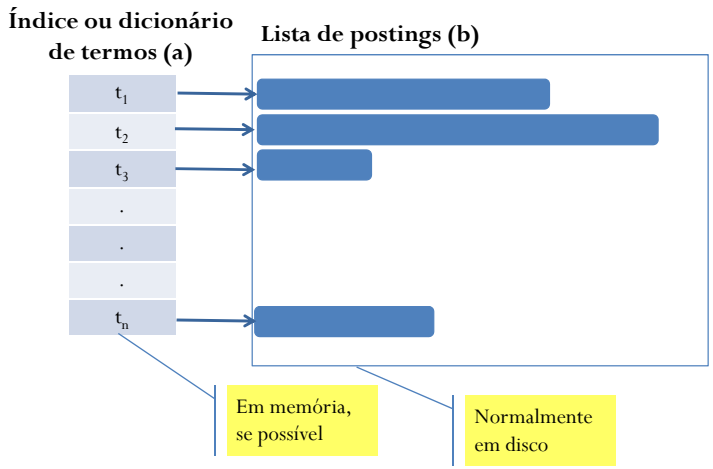
Lista de Postings

0001	001, 003, 100
0002	-
:	:
0010	002
:	:
nnnn	002, 004, 98,…

Lista de Docs

Id	Nome	Path
001	A	
002	**B.txt**	**C:\...**
003	C	
:	D	:
nnn	E	

Arquivo (lista) invertido



Estrutura do dicionário de termos (a)

Aponta para o arquivo de postings que contém a lista de documentos onde cada termo aparece!

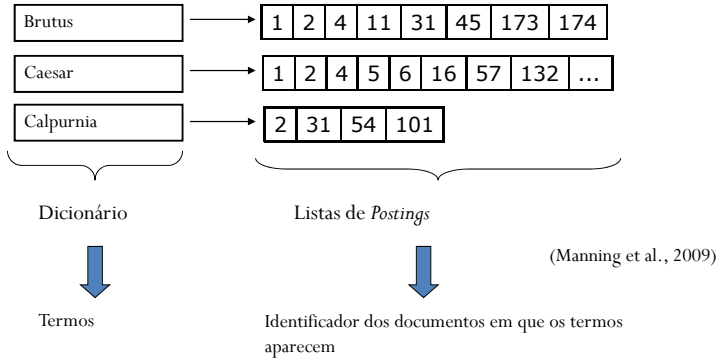
Palavra	Freq. (total)	df	Apontador (postings list file entry)
Aluno	20	10	0100
Barraca	5	2	0010
Carro	506	23	0012
:	:	:	:
Zoo	3	1	0002

2 bytes
4 bytes

Dicionário de Termos: observações

- O número de entradas é relativamente pequeno e tende a estabilizar, devido ao número de palavras
- Uma vez pronto, tende a não variar (principalmente em coleções estáticas)
- Normalmente armazenado como uma lista ordenada de palavras (vetor), NA MEMÓRIA PRINCIPAL!
  - Uma estimativa de Grossman e Frieder (2004), 2 milhões de termos ocupam menos de 32MB (sem compressão)
  - Busca binária, com complexidade relativamente baixa: $O(\log n)$; ou
  - Funções hash com lista de colisão; ou
  - **Em coleções dinâmicas, usar outras estruturas (TRIE, PATRICIA, B-TREES)**

Exemplo



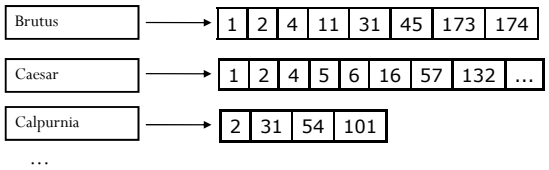
(Manning et al., 2009)

Exemplo

- Resolução:

Brutus AND Calpurnia

1. Localizar Brutus no Dicionário;
2. Recuperar sua lista de *postings*;
3. Localizar Calpurnia no Dicionário;
4. Recuperar sua lista de *postings*;
5. Calcular a intersecção entre as duas listas de *postings*.

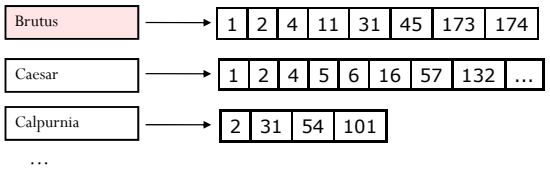


Exemplo

- Resolução:

Brutus AND Calpurnia

1. **Localizar Brutus no Dicionário;**
2. Recuperar sua lista de *postings*;
3. Localizar Calpurnia no Dicionário;
4. Recuperar sua lista de *postings*;
5. Calcular a intersecção entre as duas listas de *postings*.

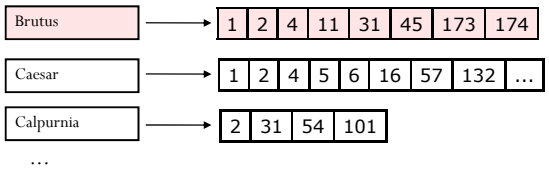


Exemplo

- Resolução:

Brutus AND Calpurnia

1. Localizar Brutus no Dicionário;
2. **Recuperar sua lista de *postings*;**
3. Localizar Calpurnia no Dicionário;
4. Recuperar sua lista de *postings*;
5. Calcular a intersecção entre as duas listas de *postings*.

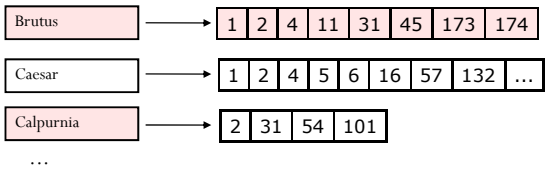


Exemplo

- Resolução:

Brutus AND Calpurnia

1. Localizar Brutus no Dicionário;
2. Recuperar sua lista de postings;
3. **Localizar Calpurnia no Dicionário;**
4. Recuperar sua lista de *postings*;
5. Calcular a intersecção entre as duas listas de *postings*.

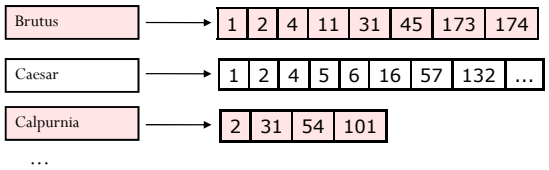


Exemplo

- Resolução:

Brutus AND Calpurnia

1. Localizar Brutus no Dicionário;
2. Recuperar sua lista de postings;
3. Localizar Calpurnia no Dicionário;
4. **Recuperar sua lista de postings;**
5. Calcular a intersecção entre as duas listas de *postings*.



Exemplo

- Resolução:

Brutus AND Calpurnia

1. Localizar Brutus no Dicionário;
2. Recuperar sua lista de postings;
3. Localizar Calpurnia no Dicionário;
4. Recuperar sua lista de postings;
5. **Calcular a intersecção entre as duas listas de postings.**