Pesquisa e Publicação de Informação

Indexação e Procura

Nuno D. Mendes

Licenciatura em Sistemas e Tecnologias de Informação

11 Mai 2012

ISEGI – UNL

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão
- Nos casos em que o repositório de documentos é muito volátil ou a economia de espaço é crucial, pode ser a única alternativa viável

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão
- Nos casos em que o repositório de documentos é muito volátil ou a economia de espaço é crucial, pode ser a única alternativa viável
- A alternativa é construir **índices** para acelerar a pesquisa de termos. A construção e manutenção de índices é desejável para colecções de documentos *semi-estáticas* (*i.e.* poucas actualizações face ao número de *queries*)

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão
- Nos casos em que o repositório de documentos é muito volátil ou a economia de espaço é crucial, pode ser a única alternativa viável
- A alternativa é construir **índices** para acelerar a pesquisa de termos. A construção e manutenção de índices é desejável para colecções de documentos *semi-estáticas* (*i.e.* poucas actualizações face ao número de *queries*)
- Técnicas mais sofisticadas implementam uma combinação de índices e pesquisa sequencial

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão
- Nos casos em que o repositório de documentos é muito volátil ou a economia de espaço é crucial, pode ser a única alternativa viável
- A alternativa é construir **índices** para acelerar a pesquisa de termos. A construção e manutenção de índices é desejável para colecções de documentos *semi-estáticas* (*i.e.* poucas actualizações face ao número de *queries*)
- Técnicas mais sofisticadas implementam uma combinação de índices e pesquisa sequencial

Tipos de índices

1. Ficheiros invertidos

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão
- Nos casos em que o repositório de documentos é muito volátil ou a economia de espaço é crucial, pode ser a única alternativa viável
- A alternativa é construir **índices** para acelerar a pesquisa de termos. A construção e manutenção de índices é desejável para colecções de documentos *semi-estáticas* (*i.e.* poucas actualizações face ao número de *queries*)
- Técnicas mais sofisticadas implementam uma combinação de índices e pesquisa sequencial

Tipos de índices

1. Ficheiros invertidos
2. Árvores/Arrays de sufixos

Motivação

- A implementação de operações sobre *queries* requer a adopção de técnicas de indexação e procura de termos em documentos
- A abordagem trivial consiste em percorrer o texto dos documentos para cada *query* (pesquisa *online* ou sequencial), mas não é prática para repositórios com alguma dimensão
- Nos casos em que o repositório de documentos é muito volátil ou a economia de espaço é crucial, pode ser a única alternativa viável
- A alternativa é construir **índices** para acelerar a pesquisa de termos. A construção e manutenção de índices é desejável para colecções de documentos *semi-estáticas* (*i.e.* poucas actualizações face ao número de *queries*)
- Técnicas mais sofisticadas implementam uma combinação de índices e pesquisa sequencial

Tipos de índices

1. Ficheiros invertidos
2. Árvores/Arrays de sufixos
3. Tabelas de Dispersão / Ficheiros de assinaturas (em desuso)

Exemplo

^{1}Este ^{6}é ^{8}um ^{11}texto. ^{18}Um ^{21}texto ^{27}tem ^{31}muitas ^{38}palavras. ^{48}As ^{51}palavras ^{60}são ^{64}feitas ^{71}de ^{74}múltiplas ^{84}letras.

Vocabulário	Ocorrências
texto	$11, 21, \ldots$
muitas	$31, \ldots$
palavras	$38, 51, \ldots$
feitas	$64, \ldots$
multiplas	$74, \ldots$
letras	$84, \ldots$

- Neste exemplo, todas as palavras são convertidas para minúsculas, a acentuação gráfica é ignorada e as *stopwords* são eliminadas. As posições referem-se à posição de cada caracter
- O endereçamento por caracter facilita o acesso directo ao texto pesquisado, mas pode usar-se igualmente a posição da palavra.
- Os ficheiros invertidos podem ter diferente granularidade de endereçamento: caracter/palavra ou bloco (que pode corresponder a um documento inteiro)
- Alguns autores reservam o termo *ficheiro invertido* para a lista de ocorrências de termos em documentos, referindo-se aos casos de ocorrência por posição de palavra/caracter como *listas invertidas*

Exemplo

[1] Este é um texto. [2] Um texto tem muitas [3] palavras. As palavras são [4] feitas de múltiplas letras.

Vocabulário	Ocorrências
texto	1, 2, ...
muitas	2, ...
palavras	3, ...
feitas	4, ...
multiplas	4, ...
letras	4, ...

- Neste exemplo, todas as palavras são convertidas para minúsculas, a acentuação gráfica é ignorada e as *stopwords* são eliminadas. As posições referem-se à posição de cada bloco. Note que as duas ocorrências do termo "palavras" coalescem.
- O uso de endereçamento por bloco reduz significativamente o espaço necessário para armazenas as listas de ocorrências, que dominam o tamanho do índice. (O tamanho do vocabulário cresce lentamente com o tamanho de textos em língua natural, especialmente com a eliminação de *stopwords*, o uso de *stemming* e outras operações sobre o texto)

Tamanho do Índice por método de endereçamento

Tabela: Tamanho do ficheiro invertido como proporção do tamanho da colecção de documentos

Índice	**Colecção Pequena** (1 MB)		**Colecção Média** (200 MB)		**Colecção Grande** (2 GB)	
	c/ stopwords	**s/ stopwords**	**c/ stopwords**	**s/ stopwords**	**c/ stopwords**	**s/ stopwords**
Palavras	0.45	0.73	0.36	0.64	0.35	0.63
Documentos	0.19	0.26	0.18	0.32	0.26	0.47
Blocos de 64KB	0.27	0.41	0.18	0.32	0.05	0.09
Blocos de 256KB	0.18	0.25	0.017	0.024	0.005	0.007

Procura

- As operações de procura no índice de modo a satisfazer as necessidades do sistema de Pesquisa de Informação envolvem três fases:

Procura

- As operações de procura no índice de modo a satisfazer as necessidades do sistema de Pesquisa de Informação envolvem três fases:
  1. **Pesquisa no vocabulário** : Cada termo da *query* é pesquisado no vocabulário individualmente. A lista de entradas no vocabulário pode estar organizada em estruturas de dados eficientes para optimizar esta operação.

Procura

- As operações de procura no índice de modo a satisfazer as necessidades do sistema de Pesquisa de Informação envolvem três fases:
  1. **Pesquisa no vocabulário** : Cada termo da *query* é pesquisado no vocabulário individualmente. A lista de entradas no vocabulário pode estar organizada em estruturas de dados eficientes para optimizar esta operação.
  2. **Obtenção das ocorrências** : A lista de ocorrências para cada termo de pesquisa encontrado é obtida e guardada para futura manipulação.

Procura

- As operações de procura no índice de modo a satisfazer as necessidades do sistema de Pesquisa de Informação envolvem três fases:
  1. **Pesquisa no vocabulário** : Cada termo da *query* é pesquisado no vocabulário individualmente. A lista de entradas no vocabulário pode estar organizada em estruturas de dados eficientes para optimizar esta operação.
  2. **Obtenção das ocorrências** : A lista de ocorrências para cada termo de pesquisa encontrado é obtida e guardada para futura manipulação.
  3. **Manipulação das ocorrências** : As ocorrências de cada termo são manipuladas de modo a resolver, designadamente, pesquisas de termos compostos (*e.g.* frases), calcular a proximidade das ocorrências no texto (relevante para algumas operações sobre *queries*). Dependendo do tipo de índice usado, poderá ser necessário efectuar uma pesquisa directa da informação em falta (*e.g.* no caso do endereçamento em bloco não é possível determinar frases ou distâncias apenas com o índice do bloco)

Procura

- As operações de procura no índice de modo a satisfazer as necessidades do sistema de Pesquisa de Informação envolvem três fases:
  1. **Pesquisa no vocabulário** : Cada termo da *query* é pesquisado no vocabulário individualmente. A lista de entradas no vocabulário pode estar organizada em estruturas de dados eficientes para optimizar esta operação.
  2. **Obtenção das ocorrências** : A lista de ocorrências para cada termo de pesquisa encontrado é obtida e guardada para futura manipulação.
  3. **Manipulação das ocorrências** : As ocorrências de cada termo são manipuladas de modo a resolver, designadamente, pesquisas de termos compostos (*e.g.* frases), calcular a proximidade das ocorrências no texto (relevante para algumas operações sobre *queries*). Dependendo do tipo de índice usado, poderá ser necessário efectuar uma pesquisa directa da informação em falta (*e.g.* no caso do endereçamento em bloco não é possível determinar frases ou distâncias apenas com o índice do bloco)
- Pesquisa binária sobre um vocabulário em ordem alfabética obtém performances muito competitivas – $O(\log n)$

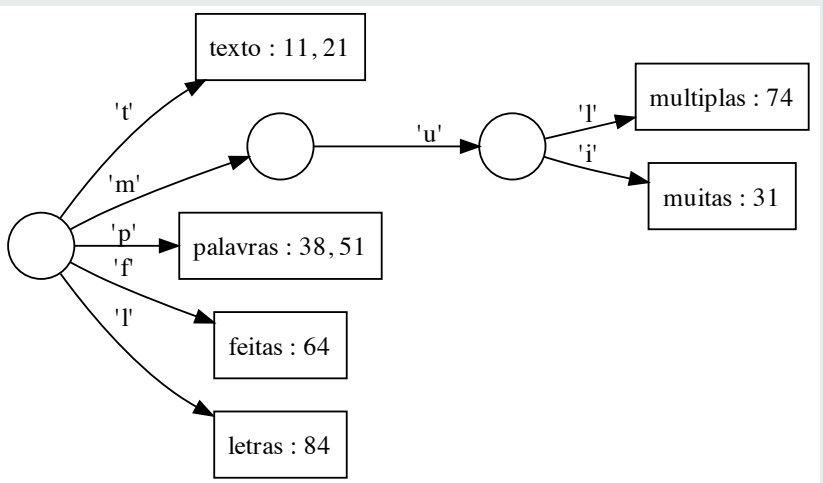
Procura

- As operações de procura no índice de modo a satisfazer as necessidades do sistema de Pesquisa de Informação envolvem três fases:
  1. **Pesquisa no vocabulário** : Cada termo da *query* é pesquisado no vocabulário individualmente. A lista de entradas no vocabulário pode estar organizada em estruturas de dados eficientes para optimizar esta operação.
  2. **Obtenção das ocorrências** : A lista de ocorrências para cada termo de pesquisa encontrado é obtida e guardada para futura manipulação.
  3. **Manipulação das ocorrências** : As ocorrências de cada termo são manipuladas de modo a resolver, designadamente, pesquisas de termos compostos (*e.g.* frases), calcular a proximidade das ocorrências no texto (relevante para algumas operações sobre *queries*). Dependendo do tipo de índice usado, poderá ser necessário efectuar uma pesquisa directa da informação em falta (*e.g.* no caso do endereçamento em bloco não é possível determinar frases ou distâncias apenas com o índice do bloco)
- Pesquisa binária sobre um vocabulário em ordem alfabética obtém performances muito competitivas – $O(\log n)$
- Outras estruturas de dados permitem que pesquisas mais complexas (*e.g.* prefixos, combinação de listas, etc) mantenham performances sublineares – $O(n^{\alpha})$, com $\alpha < 1$

Construção

- A construção de um *trie* de termos permite pesquisas sofisticadas com um custo linear de construção – $O(n)$

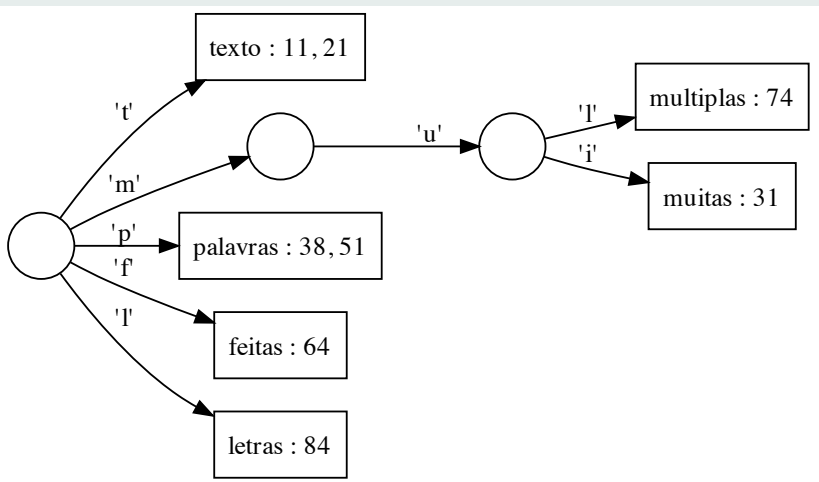
^{1}Este ^{6}é ^{8}um ^{11}texto. ^{18}Um ^{21}texto ^{27}tem ^{31}muitas ^{38}palavras. ^{48}As ^{51}palavras ^{60}são ^{64}feitas ^{71}de ^{74}múltiplas ^{84}letras.



Construção

- A construção de um *trie* de termos permite pesquisas sofisticadas com um custo linear de construção – $O(n)$

^{1}Este ^{6}é ^{8}um ^{11}texto. ^{18}Um ^{21}texto ^{27}tem ^{31}muitas ^{38}palavras. ^{48}As ^{51}palavras ^{60}são ^{64}feitas ^{71}de ^{74}múltiplas ^{84}letras.



- Para índices muito longos, é comum construirem-se sub-índices para um sub-conjunto de documentos que são hierarquicamente fundidos
- O *trie* propriamente dito e a lista de ocorrências são frequentemente mantidos em ficheiros separados, sendo a lista de ocorrências substituída por uma referência para a entrada correspondente no ficheiro de ocorrências

Sufixos

^{1}Este ^{6}é ^{8}um ^{11}texto. ^{18}Um ^{21}texto ^{27}tem ^{31}muitas ^{38}palavras. ^{48}As ^{51}palavras ^{60}são ^{64}feitas ^{71}de ^{74}múltiplas ^{84}letras.

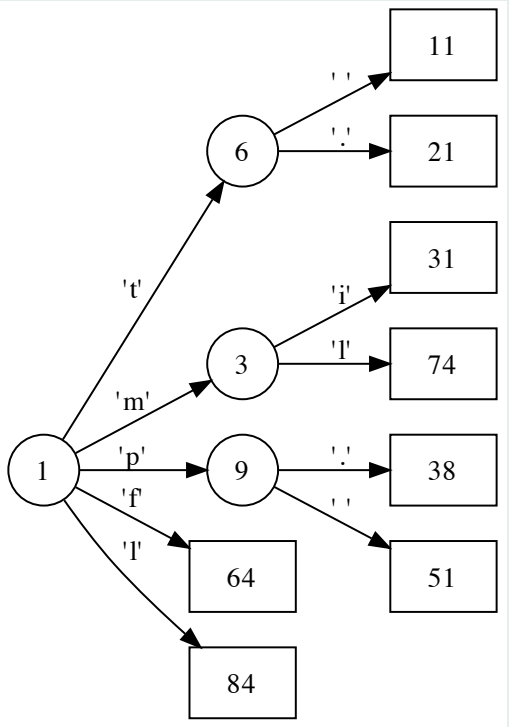
```
texto.  Um texto tem muitas palavras.  As palavras são feitas de multiplas letras.
texto tem muitas palavras.  As palavras são feitas de multiplas letras.
muitas palavras.  As palavras são feitas de multiplas letras.
palavras.  As palavras são feitas de multiplas letras.
palavras são feitas de multiplas letras.
feitas de multiplas letras.
multiplas letras.
letras.
```

Sufixos

^{1}Este ^{6}é ^{8}um ^{11}texto. ^{18}Um ^{21}texto ^{27}tem ^{31}muitas ^{38}palavras. ^{48}As ^{51}palavras ^{60}são ^{64}feitas ^{71}de ^{74}múltiplas ^{84}letras.

Árvore de Sufixos

- Podem ser usadas para indexar apenas palavras, mas permitem a indexação de qualquer sequências de carateres (importante para indexar, por exemplo, informação textual não verbal como sequências genéticas)
- Neste exemplo, consideramos apenas pontos de indexação nas fronteiras das palavras, ignorando *stopwords*
- Exigem mais espaço para guardar o índice do que os ficheiros invertidos, mas facilitam a pesquisa de frases



Sufixos

^{1}Este ^{6}é ^{8}um ^{11}texto. ^{18}Um ^{21}texto ^{27}tem ^{31}muitas ^{38}palavras. ^{48}As ^{51}palavras ^{60}são ^{64}feitas ^{71}de ^{74}múltiplas ^{84}letras.

Array de Sufixos

- Permitem reduzir substancialmente o tamanho do índice relativamente às árvores de sufixos, continuando a possibilitar essencialmente o mesmo tipo de operações de pesquisa
- A tempo de pesquisa de termos deixa, no entanto, de ser linear no tamanho do padrão (como nas árvores de sufixos), e passa a ser $O(\log n)$, porque é necessária uma pesquisa binária do termo
- O array de sufixos é contruído simplesmente visitando as folhas da árvores de sufixos por ordem lexicográfica

64	84	31	74	51	38	11	21