

Descrição

O trabalho consiste na implementação de um programa que, a partir da construção do índice invertido implementada no primeiro trabalho, realize as seguintes tarefas:

- 1 – Construa uma representação de cada documento da base de documentos como um vetor de índices referentes aos termos de indexação;
- 2 – Responda à uma consulta de entrada segundo o modelo booleano de recuperação da informação.

A base de documentos

A base de documentos é composta por um conjunto arbitrário de arquivos de texto. Assuma que nesses arquivos texto, palavras são separadas por um ou mais dos seguintes caracteres: espaço em branco (), ponto (.), vírgula (,), exclamação (!) ou interrogação (?). Assuma também que todo o conteúdo dos arquivos na base está em caracteres minúsculos e sem caracteres acentuados.

A entrada do programa

Seu programa deverá receber três argumentos como entrada **pela linha de comando**. O primeiro argumento especifica o caminho de um arquivo texto que contém os caminhos de todos os arquivos que compõem a base, cada um em uma linha. O segundo argumento especifica o caminho de um arquivo texto que traz uma lista com *stopwords*, com uma *stopword* por linha. As *stopwords* são palavras que **não** possuem significado semântico para um sistema de RI, e portanto, **não devem ser consideradas na construção do índice invertido**. O terceiro argumento é o caminho de um arquivo contendo uma consulta a ser respondida.

Exemplo: Vamos supor que nossa base é composta pelos arquivos a.txt, b.txt e c.txt. Vamos supor também que nosso programa se chama *trab2ori.exe*. Assim, chamaríamos nosso programa pela linha de comando fazendo:

```
> trab2ori.exe base.txt stopwords.txt consulta.txt
```

onde o arquivo *base.txt* contém os caminhos para os arquivos que compõem a base de documentos, conforme a seguir:

```
a.txt  
b.txt  
c.txt  
  
base.txt
```

, e o arquivo *stopwords.txt* possui uma lista de palavras que deverão ser tratados pelo sistema como *stopwords*, isto é, não deverão ser consideradas na construção do índice:

```
o
de
na
em
nao
uma
```

stopwords.txt

, e o arquivo *consulta.txt* contém o conteúdo de uma consulta. Assuma que a consulta é composta por termos do vocabulário da base de documentos, com um operador *AND* (E) entre eles. Cada termo pode ser precedido pelo caracter ‘-’, indicando o uso de um operador *MINUS* (MENOS). Por exemplo:

```
casa mora -favor
```

consulta.txt

. No exemplo acima, a consulta requisita documentos com os termos “casa” e “mora”, mas sem a presença do termo “favor”.

O programa deverá gerar três arquivos de saída:

- 1 – *indice.txt*: arquivo com o índice invertido;
- 2 – *repdocs.txt*: arquivo com a representação dos documentos da base
- 3 – *resposta.txt*: arquivo com a resposta a consulta de entrada

1 - O arquivo *indice.txt*

O arquivo *indice.txt* contém o índice invertido gerado a partir dos documentos da base. Para a geração do índice invertido, é preciso considerar cada palavra **não stopwords** que apareça em algum dos documentos da base. **Aqui, é preciso gerar o índice invertido em ordem alfabética.** Para cada uma dessas palavras no índice, é preciso apontar o número do arquivo em que a mesma aparece, e a quantidade de vezes em que a mesma aparece no arquivo. Os arquivos são numerados segundo a ordem em que aparecem no arquivo que indica os documentos da base, que, para o nosso exemplo, foi denominado como *base.txt*. Assim, o arquivo *a.txt* é o arquivo 1, o arquivo *b.txt* é o arquivo 2 e, por fim, o arquivo *c.txt* é o arquivo 3. Suponha que estes arquivos estejam preenchidos conforme abaixo:

```
era uma casa muito
engracada. nao tinha teto,
nao tinha nada.
```

a.txt

```
quem casa quer casa
muito. quem nao mora
em casa, tambem quer
casa!
```

b.txt

```
quer casar comigo, amor?
quer casar comigo,
faca muito favor! mora na
minha casa!
```

c.txt

Nesse caso, o arquivo *indice.txt* com o índice gerado seria composto por:

```
amor: 3,1
casa: 1,1 2,4 3,1
casar: 3,2
comigo: 3,2
engracada: 1,1
era: 1,1
faca: 3,1
favor: 3,1
minha: 3,1
mora: 2,1 3,1
muito: 1,1 2,1 3,1
nada: 1,1
quem: 2,2
quer: 2,2 3,2
tambem: 2,1
teto: 1,1
tinha: 1,2
```

indice.txt

Observe que, para cada palavra no índice invertido, temos uma lista de pares a,q onde a é o número do arquivo em que a palavra aparece, e q é a quantidade de vezes em que a palavra aparece no arquivo. Assim, para a palavra casa, por exemplo, temos o par 1,1, indicando que no arquivo 1, este termo aparece uma vez. Em seguida, temos o par 2,4, indicando que no arquivo 2 este termo apareceu 4 vezes. Por fim, temos o par 3,1, indicando que, no arquivo 3, este termo aparece uma vez. **Note que as stopwords não devem entrar no índice invertido!**

2 – O arquivo repdocs.txt

O arquivo repdocs.txt contém a representação de cada arquivo da base como um vetor de índices dos termos do vocabulário. Primeiramente, é preciso numerar (atribuir índices) cada termo segundo a ordem alfabética:

- 1 - amor
- 2 - casa
- 3 - casar
- 4 - comigo
- 5 - engraçada
- 6 - era
- 7 - faça
- 8 - favor
- 9 - minha
- 10 - mora
- 11 - muito
- 12 - nada
- 13 - quem
- 14 - quer
- 15 - tambem
- 16 - teto
- 17 - tinha

a partir daí, cada documento pode ser rescrito como um vetor com os índices dos respectivos termos que o compõem, **excluindo-se as stopwords**. Por exemplo, o documento a.txt:

era uma casa muito
engracada. nao tinha teto,
nao tinha nada.

a.txt

pode ser representado através do vetor [6, 2, 11, 5, 17, 16, 17, 12]. O documento b.txt:

quem casa quer casa
muito. quem nao mora
em casa, tambem quer
casa!

b.txt

pode ser representado através do vetor [13, 2, 14, 2, 11, 13, 10, 2, 15, 14, 2]. Por fim, o documento c.txt:

quer casar comigo, amor?
quer casar comigo,
faça muito favor! mora na
minha casa!

c.txt

pode ser representado através do vetor [14, 3, 4, 1, 14, 3, 4, 7, 11, 8, 10, 9, 2]. Desse modo, o arquivo repdocs.txt conterá as representações dos documentos da base, na mesma ordem em que aparecem no arquivo base.txt. Note que o arquivo repdocs.txt contém apenas os elementos dos vetores de representação, com um vetor de representação por linha:

```
6 2 11 5 17 16 17 12
13 2 14 2 11 13 10 2 15 14 2
14 3 4 1 14 3 4 7 11 8 10 9 2
```

repdocs.txt

3 – O arquivo resposta.txt

O arquivo resposta.txt contém a resposta à consulta contida no arquivo de consulta, no nosso exemplo consulta.txt. A primeira linha desse arquivo deve conter a quantidade de documentos que satisfazem a consulta. As demais linhas contém os arquivos da base que atendem a consulta, conforme o exemplo a seguir:

```
1
b.txt
```

resposta.txt

Linguagens permitidas

O programa que gera o índice invertido pode ser desenvolvido em qualquer uma das seguintes linguagens:

- C
- C++
- Java
- Python
- Haskell

Entrega

O código fonte pode ser desenvolvido em grupo de até 3 pessoas, na realidade, os mesmos grupos formados para o trabalho 1 devem ser mantidos. A entrega, até o dia ~~21/11/2017~~ 30/11/2017, por e-mail ao professor: wendelmelo@ufu.br .