

Primeiro trabalho de Organização e Recuperação da Informação 2018-02

Descrição

Este trabalho consiste em:

1. Implementação de uma rotina para a geração de um índice invertido para uma base de documentos no contexto de um sistema de Recuperação da Informação (RI);

Deve ser entregue apenas um **único** programa desenvolvido em Python que realize a tarefa descrita. O trabalho deve ser feito em grupo de **um ou dois alunos**, e o código gerado deve ser entregue por e-mail ao professor (wendelmelo@ufu.br) até a data 07/10/2018.

Aviso importante: se for detectado cópia ou qualquer tipo de trapaça entre diferentes grupos, todos os grupos serão punidos com a nota zero. Portanto, pense bem antes de pedir para copiar o trabalho do seu coleguinha, pois ele poderá ser punido também!

Não é permitido o uso de nenhuma biblioteca fora da distribuição padrão Python, com exceção do pacote nltk para a extração de radicais dos termos do vocabulário e obtenção de uma lista válida de *stopwords*. Os detalhes sobre a geração do índice invertido são descritos a seguir. **É importante ler com atenção e seguir todos os detalhes da especificação sob pena de perda de pontos na nota do trabalho!**

A base de documentos

A base de documentos é composta por um conjunto arbitrário de arquivos de texto puro. Assuma que nesses arquivos texto, palavras são separadas por um ou mais dos seguintes caracteres: espaço em branco (), ponto (.), vírgula (,), exclamação (!), interrogação (?) ou enter (\n). Seu programa deve tratar caracteres maiúsculos e minúsculos como sendo equivalentes.

As *stopwords*

As *stopwords* são termos que, tomados isoladamente, não contribuem para o entendimento do significado de um documento. Note então que, as *stopwords* **não** devem ser levadas em conta na geração do índice invertido! Seu programa deve considerar a lista de stopwords para a língua portuguesa disponível no pacote nltk, conforme visto em aula. Adicionalmente, seu programa deve utilizar o pacote nltk para considerar qualquer palavra considerada, como preposição, conjunção ou artigo como sendo *stopword* (veja os exemplos da aula do dia 18/09/2018).

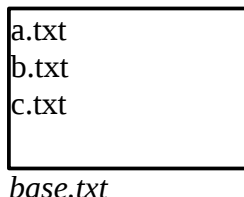
A entrada do programa

Seu programa deverá receber um argumento como entrada **pela linha de comando**. Este 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.

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 *indice_invertido.py*. Assim, chamaríamos nosso programa pela linha de comando fazendo:

```
> python indice_invertido.py base.txt
```

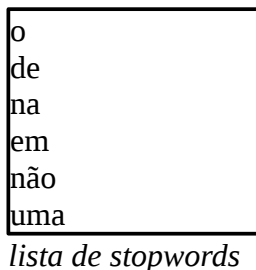
onde o arquivo *base.txt* contém os caminhos para os arquivos que compõe a base de documentos, conforme a seguir:



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

base.txt

Vamos supor que a nossa lista de *stopwords*, obtida do pacote nltk, contenha os seguintes termos:



```
o
de
na
em
não
uma
```

lista de stopwords

A saída do programa

O programa deverá gerar o seguinte arquivo de saída, seguindo exatamente a descrição a seguir:

- *indice.txt* : arquivo que contém o índice invertido gerado a partir dos documentos da base

O arquivo *indice.txt*:

O programa deve gerar um arquivo chamado *indice.txt*, que contém o índice invertido gerado a partir dos documentos da base. Para a geração do índice invertido, é preciso considerar cada palavra **não *stopword*** que apareça em algum dos documentos da base e **extrair seu radical usando o pacote nltk**.

Para cada um desses radicais no índice, é preciso apontar o número do arquivo em que o mesmo aparece, e a quantidade de vezes em que o mesmo 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 engraçada. Não tinha teto, não tinha nada.

a.txt

quem casa quer casa.
QUEM não mora em casa, também quer casa!

b.txt

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

c.txt

Apenas para facilitar o entendimento, desconsiderando a extração dos radiciais, 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
nada: 1,1
quem: 2,2
quer: 2,2 3,2
tambem: 2,1
teto: 1,1
tinha: 1,2

indice.txt (sem extração de radicais)

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!** Não é estritamente necessário que o índice invertido seja gerado em ordem alfabética. Observe que o índice invertido deve ser gerado com os termos em caracteres minúsculos, e que **é importante que o índice gerado siga exatamente o padrão aqui indicado!** Lembre-se, todavia, que o seu programa deve gerar o índice a partir dos radicais dos termos do vocabulário. Assim, considerando a extração dos radicais, o índice seria composto por:

```
am: 3,1
cas: 1,1 2,4 3,3
comig: 3,2
engraç: 1,1
era: 1,1
faç: 3,1
favor: 3,1
minh: 3,1
mor: 2,1 3,1
muit: 1,1
nad: 1,1
qu: 2,4 3,2
tamb: 2,1
tet: 1,1
tinh: 1,2
```

indice.txt (com extração de radicais)

Note que palavras que possuem o mesmo radical serão contadas como sendo equivalentes, como no caso de casa e casar.