FACOM- UFU
Professor: Wendel Melo

Primeiro trabalho de Organização e Recuperação da Informação 2023-2

Descrição

Objetivo do trabalho:

1. Implementação de um gerador de índice invertido ;

Deve ser entregue apenas um **único** programa desenvolvido em Python que realize a tarefa descrita. O trabalho deve ser feito de forma individual e o código gerado deve ser entregue através de um formulário apropriado na plataforma Teams.

Aviso importante: se for detectado cópia ou qualquer tipo de trapaça entre diferentes alunos, todos os alunos 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!

É proibido usar o pacote nltk! Trabalhos que importarem qualquer parte da biblioteca nltk receberão a nota zero!

É obrigatório o uso do pacote SpaCy do Python para a lematização dos termos do vocabulário e obtenção de uma lista válida de *stopwords*. Você deve usar o modelo de língua portuguesa apresentado em aula para fazer o seu programa. Os detalhes sobre a geração do índice 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 (.), reticências(...) 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 filtrar as *stopwords* pela classificação obtida com o modelo usado no pacote SpaCy

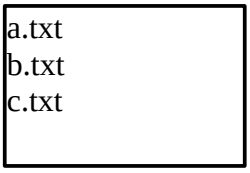
A entrada do programa

Seu programa deverá receber um argumento como entrada **pela linha de comando**. Esse argumento especifica o caminho de um arquivo texto que contém os caminhos de todos os arquivos que compõem a base de documentos, 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.py*. Assim, chamaríamos nosso programa pela linha de comando fazendo:

> python indice.py base.txt

onde o arquivo *base.txt* contém os caminhos para os arquivos que compõem a base de documentos (ressalta-se que o arquivo *base.txt* pode conter um número arbitrário de caminhos para os arquivos que compõem a base de documentos, não necessariamente 3), conforme a seguir:



base.txt

A saída do programa

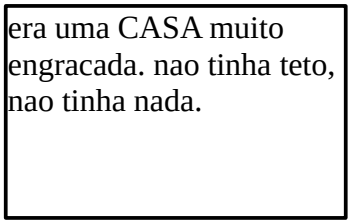
O programa deverá gerar um arquivo de saída, com nomes e conteúdo **exatamente** como a seguir:

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

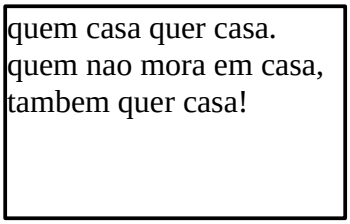
O arquivo *indice.txt*:

O programa deve gerar um arquivo chamado *indice.txt*, que contem o índice invertido gerado a partir dos documentos da base.

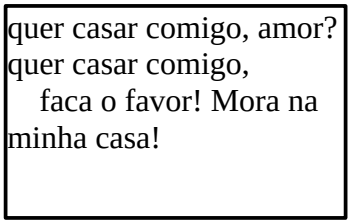
Para cada um dos termos no índice, é preciso apontar o número do arquivo em que o mesmo aparece, e a quantidade de vezes em que o mesmo aparece no arquivo. Os arquivos são numerados segundo a ordem em que aparecem no arquivo que indica os documentos da base, que, para o nosso exemplo, foi denominado 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:



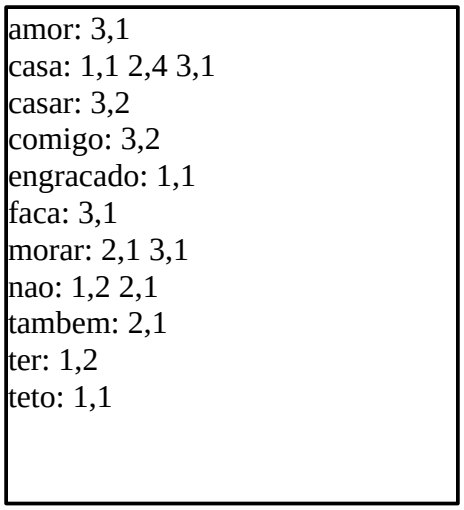
a.txt



b.txt



c.txt



indice.txt (com extração de radicais)

Não deixe de testar seu código. Você pode usar a ferramenta de teste disponibilizada pelo professor :

Para rodar o corretor, baixe e descompacte o arquivo corretor_indice.zip . Mova os aqruivos *.pyc para a pasta onde seu código está salvo. Abra um terminal do sistema operacional nessa mesma pasta (sim, o do sistema operacional e não o do python), e execute o comando:

python3 waxm_corretor_indice.pyc <ARQUIVO DA BASE> <ARQUIVO COM SEU CÓDIGO>

Se o seu sistema for Windows, talvez o comando seja esse:

py waxm_corretor_indice.pyc <ARQUIVO DA BASE> <ARQUIVO COM SEU CÓDIGO>

Por exemplo, supondo que o arquivo que especifica a base se chame base.txt e seu código esteja em um arquivo chamado indice.py, faça:

python3 waxm_corretor_indice.pyc base.txt indice.py

ou

py waxm_corretor_indice.pyc base.txt indice.py

Você também pode baixar as bases de exemplo para testar seu código.

Atenção: se o seu código não passar satisfatoriamente pelo corretor automático, seu trabalho já começa a ser corrigido com desconto de pontuação. USE O CORRETOR AUTOMÁTICO!