Atividade Avaliativa 2

Fundamentos Matemáticos

Turmas 10A e 14A

Neste segundo REO, como Atividade Avaliativa, você deverá abordar a tarefa de **buscar por termos em um conjunto de textos e retornar o texto que contém a maior frequência dos termos buscados**. Então imagine que você busque por "terremoto tornado" em um conjunto de notícias jornalísticas (documentos de textos simples), a busca deve retornar a notícia que contém a maior frequência dos termos da busca. Será explicado melhor a seguir.

Esta é uma atividade bem simples para ilustrar como uma abordagem puramente frequentista tem suas limitações, e que técnicas com maior conhecimento linguístico podem obter resultados muito melhores.

Problema

Dado uma **frase de busca** (string) e um **conjunto de textos** (arquivos .txt em codificação UTF-8, armazenados em um diretório). Você deve criar um programa que:

1) Leia, **pré-processe** e **tokenize** os textos e a string de busca

   O **pré-processamento** é composto dos seguintes passos:

   - colocar tudo em letras minúsculas
   - remover pontuações (! " # $ % & \ ' ( ) * + , - . / : ; < = > ? @ [\ \ ] ^ _ ` { | } ~)

Hoje, um dia especial, é a data do meu Aniversário!
após pré-processamento
hoje um dia especial é a data do meu aniversário

A **tokenização** deve ser feita quebrando a string de busca e os textos em seus espaços, retornando uma lista.

hoje um dia especial é a data do meu aniversário
após a tokenização
['hoje', 'um', 'dia', 'especial', 'é', 'a', 'data', 'do', 'meu', 'aniversário']

2) Calcule a **sobreposição (equação *overlap* a seguir)** de tokens entre a string de busca e cada um dos textos em busca dos textos que contêm a maior sobreposição de tokens com a string de busca.

$$overlap(T1,\ T2) = \frac{\sum tokens\ comuns\ em\ T1\ e\ T2}{tamanho\ da\ maior\ lista(T1, T2)}$$

Em que T1 e T2 são duas listas de tokens (por exemplo, T1 contém os tokens da string de busca e T2 contém os tokens de um texto)

busca = ['hoje', 'um', 'dia', 'especial', 'é', 'a', 'data', 'do', 'meu', 'aniversário']
T2 = ['aniversário', 'é', 'a', 'comemoração', 'mais', 'legal']
T3 = ['atividade', 'de', 'PLN']

overlap(busca, T2) = 0.3
overlap(busca, T3) = 0.0

3) Exibir o texto com o maior valor de *overlap*. No caso de empate, retornar qualquer um dos textos com maior overlap.

Aniversário é a comemoração mais legal

Execução

O código enviado será executado da seguinte maneira:

```
python freq_search.py directory_path search_string
```

- `freq_search.py` é o código-fonte (seu programa) que receberá os dois argumentos a seguir e realizará a busca
- `directory_path` é o caminho para o diretório em que, em sua raiz, terá um número arbitrário de textos a serem lidos
- `search_string` é uma string contendo os termos a serem buscados

Por exemplo:

```
python freq_search.py noticias "furação tornado"
```

Envio

Linguagem de programação: Python 3.*. Não é permitido o uso de bibliotecas que não sejam as nativas do Python, ou seja, não será realizado nenhum tipo de instalação (Pip, Conda ou outros).

Enviar um arquivo chamado `freq_search.py` com o código criado.

Data limite de envio (05/09/2021, às 23h59) pelo link do Campus Virtual.

Não plagiem! Detector de plágio será utilizado!