UFRJ

INSTITUTO DE COMPUTAÇÃO UFRJ

Recuperação da Informação

Introdução às implementações

Profa. Giseli Rabello Lopes

Exercício

- Considere as seguintes entradas:
  - **D:** Coleção de documentos (composta pelo texto de cada documento)
  - **`stopwords`:** Conjunto de termos que serão considerados *stopwords*
  - **`q`:** Consulta (*string* composta por termos e separadores)
  - **`separadores`:** Conjunto de caracteres que devem ser usados como separadores para a tokenização
- A estruturação a seguir é apenas um exemplo. Pode-se também utilizar arquivos texto para representar os documentos (onde o conteúdo de cada arquivo representa o texto de um documento)

*Seu programa deve funcionar para quaisquer entradas, as entradas nos slides seguintes são apenas exemplos

Exercício

- Exemplo 1 de entradas (em Python):

```
D=[['O peã e o caval são pec de xadrez. O caval é
o melhor do jog.'],
['A jog envolv a torr, o peã e o rei.'],
['O peã lac o boi'],
['Caval de rodei!'],
['Polic o jog no xadrez.']] #conjunto de
documentos
stopwords=['a', 'o', 'e', 'é', 'de', 'do', 'no',
'são'] #lista de stopwords
q='xadrez peã caval torr' #consulta
separadores=[' ',',','.','!','?'] #separadores
para tokenizacao
```

Exercício

- Exemplo 2 de entradas (em Python):

```
D=[['Parasita é o grande vencedor do Oscar 2020, com quatro prêmios'],
['Green Book, Roma e Bohemian Rhapsody são os principais vencedores do Oscar 2019'],
['Oscar 2020: Confira lista completa de vencedores. Parasita e 1917 foram os grandes vencedores
da noite'],
['Em boa fase, Oscar sonha em jogar a Copa do Mundo da Rússia'],
['Conheça os indicados ao Oscar 2020; Cerimônia de premiação acontece em fevereiro'],
['Oscar Schmidt receberá Troféu no Prêmio Brasil Olímpico 2019. Jogador de basquete com mais
pontos em Jogos Olímpicos.'],
['Seleção brasileira vai observar de 35 a 40 jogadores para definir lista da Copa América'],
['Oscar 2020: saiba como é a escolha dos jurados e como eles votam'],
['Bem, Amigos! discute lista da Seleção, e Galvão dá recado a Tite: Cadê o Luan?'],
['IFAL–Maceió convoca aprovados em lista de espera do SISU para chamada oral'],
['Arrascaeta e Matías Viña são convocados pelo Uruguai para eliminatórias da Copa. Além deles,
há outros destaques na lista.'],
['Oscar do Vinho: confira os rótulos de destaque da safra 2018'],
['Parasita é o vencedor da Palma de Ouro no Festival de Cannes'],
['Estatísticas. Brasileirão Série A: Os artilheiros e garçons da temporada 2020'],
['Setembro chegou! Confira o calendário da temporada 2020/2021 do futebol europeu']] #conjunto
de documentos
stopwords=['a', 'o', 'e', 'é', 'de', 'do', 'da', 'no', 'na', 'são', 'dos', 'com','como',
'eles', 'em', 'os', 'ao', 'para', 'pelo'] #lista de stopwords
q='oscar 2020' #consulta
separadores=[' ',',','.','!','?',':',';','/'] #separadores para tokenizacao
```

Exercício

- Implemente as etapas a seguir para o pré-processamento de documentos e consulta:
  - Tokenizar utilizando os separadores especificados na entrada
  - Normalizar termos (ex. caixa-baixa)
  - Eliminar *stopwords* especificadas na entrada
- Crie uma estrutura de indexação para apoiar a recuperação da informação que guarde a frequência de aparecimento dos termos em cada documento:
  - Ex.: índice de arquivo invertido ou “variante” da matriz de incidências que, ao invés de apenas 0s e 1s, armazena a frequência de aparecimento dos termos em cada documento

*Pode ser implementado em qualquer linguagem de programação; sugestões: Python; Scilab

Dicas: Python

- Google Colaboratory de exemplo:
  - https://colab.research.google.com/drive/176SlOR6lkcaJ1jaKtxWrKx3sNULI5QAe

Dicas: Python

- `if`
- `for`
- `range()`
- `len()`
- `.split()`
- `.lower()`
- `.append()`
- `set()`
- `sorted()`
- `&`
- `|`
- `True`
- `False`

- `np.array(a,dtype=bool)`
- `np.sort()`
- `v[::-1]`
- `v[:,c]`
- `v[l,:]`
- `np.argsort()`
- `np.zeros()`
- `np.ones()`
- `np.linalg.norm()`
- `np.dot()`
- `m.log(a,base)`
- `Matriz.T`

*np – Numpy e m – math

Dicas: Python

- Introdução sobre estrutura de dados `DataFrame` na biblioteca *pandas* (não é obrigatório utilizar, mas para quem desejar):
  - https://colab.research.google.com/notebooks/mlcc/intro_to_pandas.ipynb

	0	1	2	3	4
peã	1.0	1.0	1.0	0.0	0.0
caval	2.0	0.0	0.0	1.0	0.0
pec	1.0	0.0	0.0	0.0	0.0
xadrez	1.0	0.0	0.0	0.0	1.0
melhor	1.0	0.0	0.0	0.0	0.0
jog	1.0	1.0	0.0	0.0	1.0
envolv	0.0	1.0	0.0	0.0	0.0
torr	0.0	1.0	0.0	0.0	0.0
rei	0.0	1.0	0.0	0.0	0.0
lac	0.0	0.0	1.0	0.0	0.0
boi	0.0	0.0	1.0	0.0	0.0
rodei	0.0	0.0	0.0	1.0	0.0
polic	0.0	0.0	0.0	0.0	1.0