IA – 1° SEMESTRE DE 2022

12. EXERCÍCIO PRÁTICO – PROCESSAMENTO LINGUAGEM NATURAL

Nome: Luiz Gustavo Alves Assis da Silva
RA: 149115

1) Considere o seguinte arquivo textual: https://www.kaggle.com/crawford/20-newsgroups

   a) Escolha 2 tópicos dentre os 20 disponíveis no dataset e faça download dos mesmos.
   b) Gere um *Bag of words*: Leia o texto e armazene cada palavra em uma posição em um vetor. Faça a contagem da frequência de cada palavra gerando uma matriz termo-frequência. Na última coluna armazene o rótulo do texto (‘1’ = textos do tópico 1 e ‘2’ = textos do tópico 2)

Story	based	premise	congress	...	work	versions	make	sure	Rótulo
1	2	0	3		0	0	0	0	1
0	0	0	0		1	5	0	1	2
...									? ?

   c) Remova os *stop words* (palavras irrelevantes): https://gist.github.com/sebleier/554280
   d) Escolha **dois** algoritmos de classificação vistos anteriormente (knn, naive bayes, arv. Decisão, svm, etc) e classifique os textos (separar 70% para treino e 30% para teste). Anexar a saída e % de acerto de cada algoritmo.

(OBS: Por decorrência do tempo e dificuldade de implementação, não consegui fazer o item d para classificar as palavras).

```
'''

#
#   AULA 12
#   PROCESSAMENTO DE LINGUAGEM NATURAL (PLN)
#   LINGUAGEM DE PROGRAMAÇÃO: Python
#
#   NOME: LUIZ GUSTAVO ALVES ASSIS DA SILVA
#   RA: 149115
#

'''

# BIBLIOTECAS UTILIZADAS

import pandas as pd
import numpy as np
import csv

stop_words = open("stopwords.txt", "r")
stop_words_linhas = stop_words.readlines()
lst_stop_words = []

for linha in stop_words_linhas:
    lst_stop_words.append(linha.strip())
```

Temos um programa que realiza o processamento do texto e gera um BOW (Bag-Of-Words).
Em primeira instância, é feita a remoção das "stop-words" (palavras irrelevantes) dos arquivos de texto para facilitar a contagem de frequência dos termos.

```
def caractere_invalido(linha):

    caracteres_escpeciais = ['!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.',
                             '/', ':', ';', '<', '=', '>', '@', '[', ']', '|', '{', '}', '~', '?',
                             '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '_', '^', '\\', '`']

    for caractere in caracteres_escpeciais:
        linha = linha.replace(caractere, ' ')

    return linha
```

A função *caractere_invalido* verifica a existência de caracteres especiais em cada linha de texto e, em seguida, aplica a remoção destes caracteres, substituindo por um caractere de espaço.

```python
def processamento_texto(texto):

    lst_data = []

    for linha in texto:
        palavras = caractere_invalido(linha.decode(errors='ignore')).split()

        for palavra in palavras:
            if palavra.lower() not in lst_stop_words:
                lst_data.append(palavra.lower())

    palavra_count = {}

    for palavra in lst_data:
        if palavra not in palavra_count.keys():
            palavra_count[palavra] = 1
        else:
            palavra_count[palavra] += 1

    return palavra_count
```

A função *processamento_texto* armazena cada palavra do texto em uma lista de palavras (não permitindo caracteres especiais e repetição de palavras), em seguida é feita a contagem de ocorrência de todas as palavras no texto. Após o término da contagem, é retornado um dicionário que contém as palavras como chave e o número de ocorrências como valor.

```python
def termo_freq(text_unique, data):

    text = {}

    for palavra in text_unique:
        if palavra not in data:
            text[palavra] = 0
        else:
            text[palavra] = data[palavra]

    return text
```

A função *termo_freq*, como nome diz, calcula a frequência de cada um dos termos nos textos dos topicos escolhidos em questão.

```python
def gerar_BOW(data1, data2):

    text = list(data1.keys()) + list(data2.keys())
    text_unique = list(np.unique(text))
    text_unique.insert(len(text_unique), 'ROTULO')

    t1 = termo_freq(text_unique, data1)
    t1['ROTULO'] = 1

    t2 = termo_freq(text_unique, data2)
    t2['ROTULO'] = 2

    BOW = open('BOW.csv', mode='w', newline='')
    with BOW as csv_file:
        writer = csv.DictWriter(csv_file, fieldnames=text_unique)

        writer.writeheader()
        writer.writerow(t1)
        writer.writerow(t2)

    return text_unique
```

A função *gerar_BOW* realiza a geração do BOW (Bag of Words) concatenando as duas listas em um texto só e, em seguida, as palavras dos textos são filtradas, por meio da função np.unique, permitindo apenas a existência de palavras distintas no texto. Além disso, é criada a coluna "RÓTULO" na última posição do vetor.

Após isso, é calculado a frequência dos termos e é feita a formatação das palavras de acordo com o rótulo ("1" para textos do tópico 1 e "2" para textos do tópico 2). Por fim, é criado um arquivo do tipo csv para armazenar a BOW de acordo com a frequência das palavras e o rótulo atribuído.

```python
if __name__ == '__main__':

    t1 = open("soc.religion.christian.txt", "rb")
    data1 = processamento_texto(t1)

    t2 = open("talk.politics.guns.txt", "rb")
    data2 = processamento_texto(t2)

    gerar_BOW(data1, data2)
    BOW = pd.read_csv("BOW.csv")
    print(BOW)
```

Na função main, os 2 tópicos escolhidos dentre os 20 disponíveis foram: "soc.religion.christian" e "talk.politics.guns". Em seguida, é feito o processamento dos textos e a geração do arquivo csv (BOW) e a impressão deste arquivo é feita no terminal do usuário.

```
   aa  aaa  aaah  aacc  aad  aaef  ...  zxqi  zy  zyg  zyhszg  zz  ROTULO
0  48    0     0     2    0     0  ...     0   0    0       0   0       1
1  22    2     4     0    6     2  ...     2   4    2       2   6       2

[2 rows x 26385 columns]
(2, 26385)

Process finished with exit code 0
```