Universidade Presbiteriana Mackenzie

Tarefa 1 - J Renato M

Escolhi o texto sobre "Argumentação" pois trata-se de um texto didático e de fácil acesso. De acordo com a lei Zipf " termos muito frequentes, como artigos, preposições etc. que parecem contribuir muito pouco para explicar o conteúdo de um texto."

Aqui, criamos um dicionário de têrmos (palavras) sobre texto exemplo.

1. A primeira célula obtém o texto exemplo na variável `texto`.
2. Na segunda célula cria o dicionário de têrmos com a estrutura:

```
mydict = { 'word1': qty, 'word2': qty, 'word3': qty, ...}
```

3. A terceira célula apresenta um gráfico de distribuição dos têrmos do seu dicionário para confirmar a lei de Zipf

A sua tarefa pode empregar outros textos de seu interesse, inclusive em inglês e outras línguas de mesmas características (francês, alemão, espanhol etc.), e você também pode querer empregar arquivos locais. Existem inúmeros pré- tratamentos possíveis nos dados e diferentes formas de exibir os dados. Para o pré tratamento você pode incluir outras transformações para melhorar a qualidade do seu dicionário. Para exibição dos dados você pode empregar o mesmo código ou buscar uma outra forma de sua preferência.

Aquisição dos Dados

```
import urllib.request

texto = []

for line in urllib.request.urlopen('http://educacao.globo.com/portugues/assunto/texto-argumentativo/argumentacao.html'):
    texto.append(line.decode('utf-8'))

# f = open('/kate_beckinsale.txt','r')    # para arquivos locais
# for line in f:
#    texto.append(line)

for i in range(len(texto)):
    texto[i] = texto[i].lower() # para unicidade
    texto[i] = texto[i].replace('\n','')
    texto[i] = texto[i].replace('.','')
    texto[i] = texto[i].replace(',','')
    texto[i] = texto[i].replace('(','')
    texto[i] = texto[i].replace(')','')
    texto[i] = texto[i].replace('?','')
    texto[i] = texto[i].replace('\'','') # elimina ' e "
```

Construção do Dicionário

```
mydict = {}                           # crie um dicionário vazio

for line in texto:

  line = line.lower()                 # converte para lower
  words = line.split()                # separa cada palavra
#  print(words)

  for word in words:                    # para cada palavra em words
```

```
    if word not in mydict.keys():         # se palavra não está no dicionário
      mydict[word] = 1                    # acrescenta a word com o valor 1
    else:                                 # se a entrada já existe
      mydict[word] =  mydict[word] + 1    # apenas soma 1 ao valor já existente

print(mydict)
```

```
    parentnodeinsertbeforepos;};</script>': 1, '</body>': 1, 'página': 1, 'gerada': 1, '10/07/2015': 1, '12:03:39': 1, '</html>': 1}
```

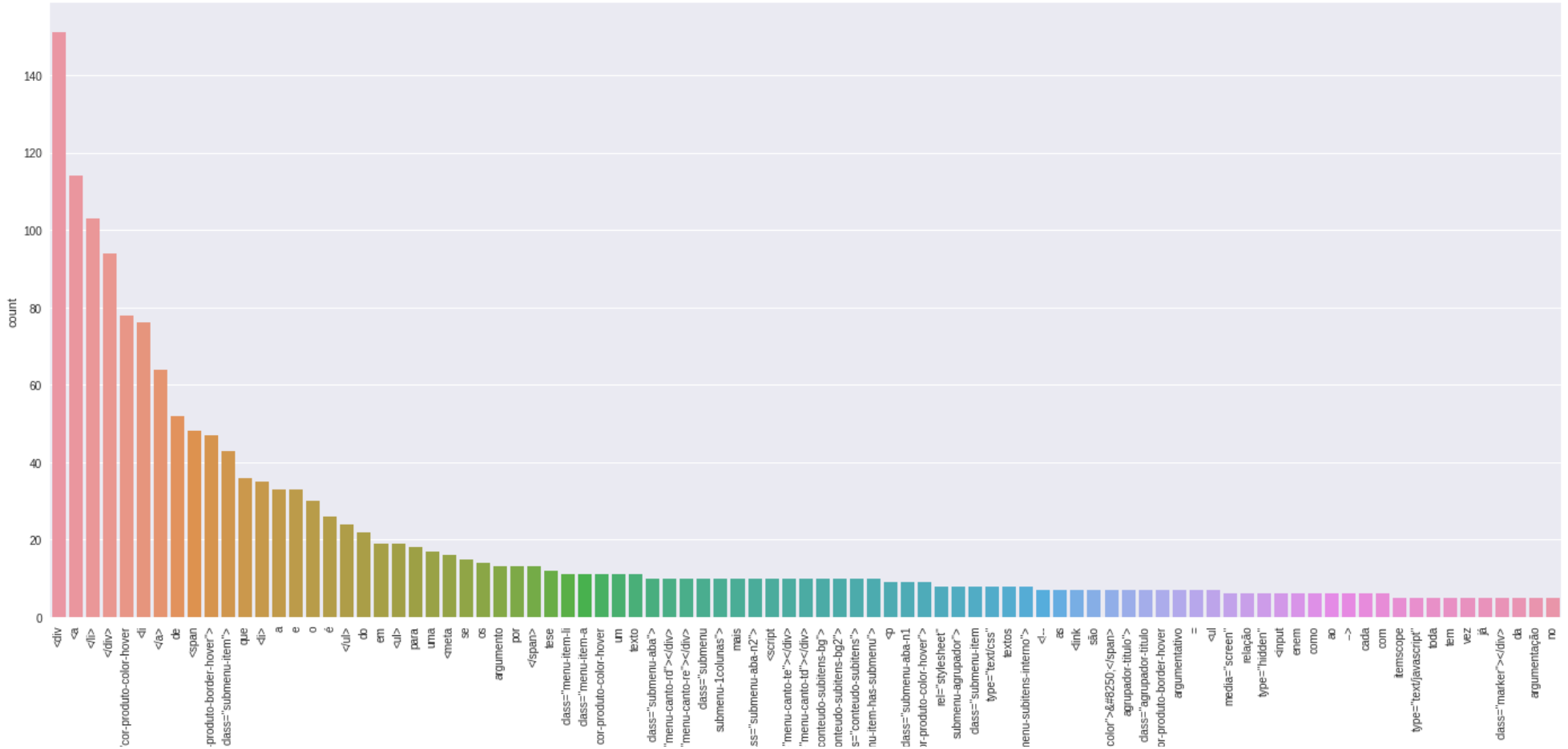
Exibição dos Resultados

```
import pandas as pd
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt

df = pd.DataFrame(mydict.items(),  columns=['word', 'count']).sort_values('count',ascending=False)
df = df[df['count'] > 4] # somente termos com mais de 4 ocorrências
# df = df.iloc[ np.int(len(df)/2) - 10 : np.int(len(df)/2) + 10 ] # para livros ou textos com muitos termos limita a um número mínimo

plt.figure(figsize=(24,10))
mpl.style.use(['seaborn'])
sns.barplot(x=df.word,y=df['count'])
plt.xticks(rotation=90)

plt.show()
```



Conclusão: De acordo com a lei Zipf " termos muito frequentes, como artigos, preposições etc. que parecem contribuir muito pouco para explicar o conteúdo de um texto."

✓ 1s conclusão: 07:32