

Relatório sobre o código: 'pre-processamento.ipynb'

```
1 import nltk
2 import re
3 from nltk.probability import FreqDist
4 import matplotlib.pyplot as plt
5
6 nltk_id = 'machado'
7 nltk.download(nltk_id)
8
9 print(nltk.corpus.machado.readme())
10 dom_casmurro = nltk.corpus.machado.raw('romance/marm08.txt')
11
12 print(dom_casmurro)
```

- Realizada as importações das bibliotecas necessárias. L1-4
- Feito o download do material relacionado a Machado De Assis através de seu ID: 'machado'. L6-7
- Print do readme em relação aos contos e histórias disponíveis do autor escolhido. L9
- Seleção da história/conto em formato texto a partir dos códigos disponíveis através do readme. L10
- Print da história escolhida. L12

```
14 dom_casmurro_letras_min = re.findall(r'\b[A-ZÁ-úü]+\b', dom_casmurro.lower())
15 print(dom_casmurro_letras_min)
16
17 nltk.download('stopwords')
18 stopwords = nltk.corpus.stopwords.words('portuguese')
19
20 print(stopwords)
21
22 list_stopwords_portugues = set(stopwords)
23 dom_casmurro_letras_min_semstop = [w for w in dom_casmurro_letras_min if w not in
    list_stopwords_portugues]
24 print(dom_casmurro_letras_min_semstop)
```

- É utilizada a lib 're', onde o método findAll() como primeiro argumento recebe um regex (para retirada de caracteres especiais, pontuações e números) e o

contéudo a ser aplicado o método em si, que já foi passado com caracteres minúsculos através do método lower(). L14

- Print da ação acima. L15
- Download do material de 'stopwords' através da lib 'nltk'. L17
- Utilização dos stopwords da língua portuguesa. L18

```
['a', 'à', 'ao', 'aos', 'aquela', 'aquelas', 'aquele', 'aqueles', 'aquilo', 'as', 'às', 'até', 'com', 'como', 'da', 'das', 'de',  
, 'dela', 'delas', 'dele', 'deles', 'depois', 'do', 'dos', 'e', 'é', 'ela', 'elas', 'ele', 'eles', 'em', 'entre', 'era', 'eram',  
, 'éramos', 'essa', 'essas', 'esse', 'esses', 'esta', 'está', 'estamos', 'estão', 'estar', 'estas', 'estava', 'estavam', 'estáv  
amos', 'este', 'esteja', 'estejam', 'estejamos', 'estes', 'esteve', 'estive', 'estivemos', 'estiver', 'estivera', 'estiveram',  
, 'estivéramos', 'estiverem', 'estivermos', 'estivesse', 'estivessem', 'estivéssemos', 'estou', 'eu', 'foi', 'fomos', 'for', 'for  
a', 'foram', 'fôramos', 'forem', 'formos', 'fosse', 'fossem', 'fôssemos', 'fui', 'há', 'haja', 'hajam', 'hajamos', 'hão', 'have  
mos', 'haver', 'hei', 'houve', 'houvemos', 'houver', 'houvera', 'houverá', 'houveram', 'houvéramos', 'houverão', 'houverei', 'h  
ouverem', 'houveremos', 'houveria', 'houveriam', 'houveríamos', 'houvermos', 'houvesse', 'houvessem', 'houvéssemos', 'isso', 'i  
sto', 'já', 'lhe', 'lhes', 'mais', 'mas', 'me', 'mesmo', 'meu', 'meus', 'minha', 'minhas', 'muito', 'na', 'não', 'nas', 'nem',  
, 'no', 'nos', 'nós', 'nossa', 'nossas', 'nosso', 'nossos', 'num', 'numa', 'o', 'os', 'ou', 'para', 'pela', 'pelas', 'pelo', 'pel  
os', 'por', 'qual', 'quando', 'que', 'quem', 'são', 'se', 'seja', 'sejam', 'sejamos', 'sem', 'ser', 'será', 'serão', 'serei', 's  
eremos', 'seria', 'seriam', 'seríamos', 'seu', 'seus', 'só', 'somos', 'sou', 'sua', 'suas', 'também', 'te', 'tem', 'têm', 'tem  
os', 'tenha', 'tenham', 'tenhamos', 'tenho', 'terá', 'terão', 'terei', 'teremos', 'teria', 'teriam', 'teríamos', 'teu', 'teus',  
, 'teve', 'tinha', 'tinham', 'tínhamos', 'tive', 'tivemos', 'tiver', 'tivera', 'tiveram', 'tivéramos', 'tiverem', 'tivermos', 't  
ivesse', 'tivessem', 'tivéssemos', 'tu', 'tua', 'tuas', 'um', 'uma', 'você', 'vocês', 'vos']
```

Figura 1 Coletânea de stopwords da língua portuguesa. Retorno da L20

- Transforma o retorno dos stopwords do nltk em um set, para retirada de possíveis termos repetidos na coleção. L20
- Retirada de todos os termos presentes de stopwords gerados na L20 da lista que foi gerada do livro de Machado de Assis já pré-processada na L14. L23
- Print da nova lista gerada já pré-processada e sem stopwords. L24

Pontos importantes a serem ressaltados:

Até o momento duas etapas foram realizadas de extrema importância, sendo elas o seu pré-processamento, utilizado para o tratamento do texto. Necessário para sua padronização e posteriormente a construção do BoW.

Retirada de stopwords da língua portuguesa, stopwords são consideradas palavras sem relevância para a consulta que fazem apenas conexões entre termos semânticos, por isso sua retirada da coleção é muito importante, afim de diminuir o tamanho total do array gerado para o BoW posteriormente.

```
26 porter = nltk.PorterStemmer()  
27 dom_casmurro_letras_min_semstop_stem = [porter.stem(t) for t in  
    dom_casmurro_letras_min_semstop]  
28 print(dom_casmurro_letras_min_semstop_stem)  
29  
30 freq_sem_stem = FreqDist(dom_casmurro_letras_min_semstop)  
31 freq_com_stem = FreqDist(dom_casmurro_letras_min_semstop_stem)
```

- Atribuição do Stemming para a variável 'porter'. L26
- Aplicação do Stemming para a coleção que estamos trabalhando. L27
- Print do resultado do algoritmo. L28
- Aplicação de distribuição de frequência para as coleções COM e SEM Stemming. L30 e L31

Pontos importantes a serem ressaltados:

Note que foi abordado o Stemming, essa função processa uma palavra e trabalha retornando apenas sua raiz. Exemplo: 'Amando' → Se transforma em 'Ama'.

Existe também a função Lemmatization, a qual não foi utilizada no algoritmo por questões de performance. Por ser uma função mais complexa textos mais longos necessitariam de um poder de processamento alto e conseqüentemente levariam um tempo de execução elevado também.

De maneira semelhante a função Lemmatization transforma uma palavra em sua raiz verdadeira, ou seja, verbo no infinitivo. Exemplo: 'Amando' -> Se transforma em 'Amar'.

As duas funções estão presentes no NLTK, Stemming: PorterStemmer e LancasterStemmer. Assim como um 'lemmatizador' WordNetLemmatizer.

```

33 print("20 palavras mais frequentes sem stem:")
34 print(freq_sem_stem.most_common(20))
35
36 print("20 palavras mais frequentes com stem:")
37 print(freq_com_stem.most_common(20))
38
39 plt.figure(figsize = (13, 8))
40 freq_sem_stem.plot(25, title = "Frequência de Palavras - Sem Stemming")
41
42 plt.figure(figsize = (13, 8))
43 freq_com_stem.plot(25, title = "Frequência de Palavras - Com Stemming")

```

- Apresenta os 20 termos da coleção com e sem Stemming. L33 a L37
- Através da lib 'matplotlib' plota no gráfico a frequência das palavras COM e SEM Stemming. L39 a L43