

# Ex1

## Pré-processamento

Pré-processamento é uma etapa importante que envolve a limpeza e transformação dos textos brutos dos documentos para facilitar a análise. Tendo como objetivo filtrar as informações, removendo palavras não significativas e transformando o texto em uma representação mais adequada para a análise.

## *StopWords*

*Stopwords* são palavras que são comuns e que geralmente não contribuem para o significado do texto como, por exemplo: 'as', 'e' , 'os', 'de' e etc.

## *Stemming e Lemmatization*

As duas são técnicas para reduzir palavras para a sua forma base/raiz. A diferença é que *stemming* apenas remove prefixos e sufixos, já *lemmatization* envolve regras gramaticais. Por exemplo, 'processando' em *stemming* ficaria 'process' já em *lemmatization* ficaria 'processar'.

## Explicação passo a passo:

```
[6] nltk_id = 'machado'

[7] nltk.download(nltk_id)
```

Primeiramente, é baixado da biblioteca NLTK recursos relacionados a Machado de Assis.

```
miscelanea/mams08.txt: A Paixão de Jesus (1904)
miscelanea/mams09.txt: Gonçalves Dias (1906)
miscelanea/mams10.txt: A Estátua de José de Alencar (1906)

[9] dom_casmurro = nltk.corpus.machado.raw('romance/marm08.txt')

print(dom_casmurro)
os dois amigos da universidade ine levantaram um tumulo com esta inscri
tirada do profeta Ezequiel, em grego: 'Tu eras perfeito nos teus
caminhos'. Mandaram-me ambos os textos, grego e latino, o desenho da se
```

Corpus acessa a coleção 'machado', carrega o texto de 'marm08.txt' presente no diretório 'romance' e armazena em 'dom\_casmurro'

```
[11] dom_casmurro_letras_min = re.findall(r'\b[A-zÀ-úü]+\b', dom_casmurro.lower())

print(dom_casmurro_letras_min)

['romance', 'dom', 'casmurro', 'dom', 'casmurro', 'texto', 'de', 'referência', 'obras', 'completa']
```

Agora vamos remover as stopwords, ou seja, as palavras que não possuem valor semântico para a busca (exer

É então extraído todas as palavras em letras minúsculas de 'dom\_casmurro' e armazenadas em 'dom\_casmurro\_letras\_min'

```
[13] nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
True

stopwords = nltk.corpus.stopwords.words('portuguese')

print(stopwords)

['a', 'à', 'ao', 'aos', 'aquela', 'aquelas', 'aquele', 'aqueles', 'aquilo', ...]
```

A coleção de stopwords é baixada do pacote NLTK e logo é armazenado as *stopwords* da língua portuguesa para 'stopwords'

```
[15] list_stopwords_portugues = set(stopwords)

[16] dom_casmurro_letras_min_semstop = [w for w in dom_casmurro_letras_min if w not in list_stopwords_portugues]

print(dom_casmurro_letras_min_semstop)

['romance', 'dom', 'casmurro', 'dom', 'casmurro', 'texto', 'referência', 'obras', 'completas', 'machado', 'assis', ...]
```

É criado uma lista cujo recebe todas as palavras da lista 'dom\_casmurro\_letras\_min' exceto aquelas presentes na lista de *stopwords*

```
[18] porter = nltk.PorterStemmer()

dom_casmurro_letras_min_semstop_stem = [porter.stem(t) for t in dom_casmurro_letras_min_semstop]

[ ] print(dom_casmurro_letras_min_semstop_stem)

['romanc', 'dom', 'casmurro', 'dom', 'casmurro', 'texto', 'referência', 'obra', 'completa', 'machado']
```

Vamos ver a frequência de ocorrência dos termos e tentar enxergar diferenças entre o texto processado com e sem

É realizado o *stemming* usando o algoritmo de Porter para normalizar as palavras (reduzi-las a uma forma raiz, removendo prefixos e sufixos). Estas são colocadas em `dom_casmurro_letras_min_semstop_stem`

```
[20] freq_sem_stem = FreqDist(dom_casmurro_letras_min_semstop)
     freq_com_stem = FreqDist(dom_casmurro_letras_min_semstop_stem)

[21] print("20 palavras mais frequentes sem stem:")
     print(freq_sem_stem.most_common(20))

20 palavras mais frequentes sem stem:
[('capitu', 341), ('mãe', 229), ('dias', 192), ('tudo', 189), ('capítulo', 189), ('outra', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189)]

[ ] print("20 palavras mais frequentes com stem:")
     print(freq_com_stem.most_common(20))

20 palavras mais frequentes com stem:
[('capitu', 341), ('dia', 302), ('mãe', 230), ('capítulo', 191), ('outra', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189), ('capit', 189), ('mã', 189), ('dia', 189)]
```

Usa-se *FreqDist* da NLTK para calcular a frequência de cada palavra e então printa-se as 20 palavras mais frequentes

Por fim, é exibido os gráficos de frequência das palavras sem e com *stemming*

