Tokenização

Transcrição

Imaginemos que dois estudantes estão aprendendo um novo idioma, o russo.

Ao final de um ano de estudos, o primeiro leu apenas um livro de 100 páginas, enquanto o outro leu toda a coleção do Game of Thrones em russo, sendo que cada parte possui quase mil páginas. Neste caso, a segunda pessoa possui muito mais informações sobre a língua, pois teve acesso à um **vocabulário** muito maior, inclusive estando apta a fazer correções.

Portanto, a **quantidade** de vocábulos é interessante para este aprendizado, o que também vale para o nosso corretor.

Para sabermos se o nosso corpus é ideal para o trabalho, deveremos saber quantas palavras possui. Afinal, um texto com dez palavras faz no máximo dez correções por exemplo. Então precisaremos ter um corpus com um grande volume de termos.

Neste caso, utilizar apenas a função `len()` recebendo `artigos` não nos ajudará, pois nos retornará apenas a quantidade de caracteres presentes no corpus, e não a de palavras que queremos.

Criaremos uma variável `texto_exemplo` sendo igual a frase `"Olá, tudo bem?"` para entendermos melhor o que fazer.

Em seguida, geraremos outra variável chamada `palavras_separadas`, onde usaremos o método `.split()` de Python em `texto_exemplo`.

Na célula seguinte, *printaremos* a `palavras_separadas` com `print()`.

```
texto_exemplo = "Olá, tudo bem?"
palavras_separadas = texto_exemplo.split()
```

COPIAR CÓDIGO

```
print(palavras_separadas)
```

COPIAR CÓDIGO

Executando este comando, receberemos a frase separada desta forma: `['Olá,','tudo','bem?']`

Porém, nem a vírgula da primeira parte em `'Olá,'` e nem o ponto de interrogação na terceira em `'bem?'` fazem parte dessas palavras em si.

De qualquer forma o método `split()` pode nos ajudar neste caso. Se imprimirmos o `len()` de `palavras_separadas`, teremos o tamanho do vetor com os três itens retornados no comando anterior.

```
print(len(palavras_separadas))
```

COPIAR CÓDIGO

Mas queremos os vocábulos separados, e não concatenados com pontuação.

Em Processamento de Linguagem Natural, cada um dos elementos separados desse vetor é conhecido como **token**. Ou seja, ao pegarmos as `strings` e as separarmos em pequenos pedaços, estamos *tokenizando* a frase.

Portanto, ainda temos apenas tokens separados, e não palavras separadas propriamente ditas.

Para a nomenclatura não nos confundir, mudaremos o nome da variável `apalavras_separadas` para apenas `tokens`.

Em seguida, imprimiremos o `tokens` e o `len()` de `tokens`.

```
texto_exemplo = "Olá, tudo bem?"
tokens = texto_exemplo.split()
```

COPIAR CÓDIGO

```
print(len(tokens))
```

COPIAR CÓDIGO

```
print(tokens)
```

COPIAR CÓDIGO

Mas ainda precisamos das palavras separadas de fato, pois queremos saber a quantidade que nosso corpus possui.

Uma forma de fazer isso seria percorrer cada um dos caracteres dos tokens e excluir somente as pontuações, como no caso da vírgula de `'Olá,'` e a interrogação de `'bem?'`.

Porém, fazer isso em um volume muito grande de palavras como o nosso seria extremamente trabalhoso.

A seguir, descobriremos como criar os tokens já separando a pontuação, facilitando bastante o trabalho.