Refinando a tokenização

Transcrição

O processo de transformar um arquivo de texto em pequenos tokens é chamado de **Tokenização**, o qual é recorrente no pré-processamento ou na análise estatística de dados textuais. Então, como é bastante utilizado, temos ferramentas que facilitam a sua implementação.

Uma bem conhecida na área de NLP é o `nltk` ou *Natural Language Tool Kit*, que é um conjunto de ferramentas que implementa diversos métodos e algoritmos para análise textual.

Apenas relembrando nosso problema, separamos os tokens da nossa frase com `split()`, mas as palavras ainda estão junto com as pontuações. Então separaremos em tokens de palavras e tokens de pontuação.

No Google Colab, criaremos uma nova lista chamada de `palavras_separadas` como já havíamos chamado anteriormente, mas agora teremos palavras separadas de pontuação.

Esta será igual ao `nltk` com a classe **`tokenize`**, a qual possui métodos de tokenização implementados, como o **`word_tokenize()`** que chamaremos em seguida.

Este fará justamente o que precisamos: separará as palavras das pontuações, e retornará uma lista com esses tokens.

Passaremos nosso corpus como parâmetro. Por enquanto, passaremos apenas o corpus `texto_exemplo` para compararmos.

```
palavras_separadas = nltk.tokenize.word_tokenize(texto_exemplo)
```

COPIAR CÓDIGO

Ao rodarmos a célula, receberemos uma mensagem de erro.

Isso aconteceu porque não fizemos os downloads necessários de alguns pacotes quando instalamos o `nltk`. O próprio sistema já indica que a solução é executar o comando `nltk.download('punkt')` antes de realizarmos a tokenização de fato.

Na célula seguinte, imprimiremos `palavras_separadas` para termos acesso às palavras.

```
import nltk
nltk.download('punkt')
palavras_separadas = nltk.tokenize.word_tokenize(texto_exemplo)
```

COPIAR CÓDIGO

```
print(palavras_separadas)
```

COPIAR CÓDIGO

Como retorno, teremos uma lista com as palavras e pontuações separadas, exatamente como queríamos: `['Olá', ',', 'tudo', 'bem', '?']`.