
# NLP COM NLTK

Tópicos especiais em gestão de TI.

PROF. DIEGO ROSSI

# NLTK

- NLTK é o acrônimo para Natural Language Toolkit.
- Uma biblioteca bastante importante para trabalhar com processamento de linguagem natural.
- Ampla biblioteca:
  - Tokenization
  - Stemming
  - Tagging
  - Parsing
  - Etc.

# COMO UTILIZAR A BIBLIOTECA

- Para utilizá-la é necessário instalá-la, exceto no colab.
  - !pip install nltk
- Importando a biblioteca.
  - import nltk
- Download de dados, modelos, etc.
  - nltk.download('stopwords')

# TOKENIZAÇÃO

- O NLTK permite realizar a tokenização de sentenças e palavras.

```
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
nltk.download ("punkt")
Frase = "Vamos nos aventurar no universo de processamento de linguagem natural, porém agora iremos utilizar a biblioteca NLTK. A princípio, a maneira de utilizá-la é um pouco diferente.“
sentencas = sent_tokenize(Frase, language='portuguese’)

tokens = word_tokenize(Frase, language="portuguese")
print (tokens)
print(len(tokens))
```

# REMOVENDO STOPWORDS

```
from nltk.corpus import stopwords
nltk.download("stopwords")

stops = stopwords.words ("portuguese")
print(len (stops))
print (stops)
palavras_sem_stopwords = [p for p in tokens if p not in stops]
print(len(palavras_sem_stopwords))
print(Frase)
print(palavras_sem_stopwords)
```

# REMOVENDO PONTUAÇÃO

- Podemos remover a pontuação das frases, reduzindo ainda mais a sua dimensionalidade.

```
import string

palavras_sem_pontuacao = [p for p in palavras_sem_stopwords if p not in string.punctuation]
print(len(palavras_sem_pontuacao))
```

# FREQUÊNCIA

- Distribuição de frequências

```
frequencia = nltk. FreqDist (palavras_sem_pontuacao)
frequencia
```

```
mais_comuns = frequencia.most_common(5)
mais_comuns
```

# STEMMING

- A ferramenta de stemming é responsável por extrair o radical das palavras.

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()
stem = [stemmer.stem (word) for word in palavras_sem_pontuacao]
print (palavras_sem_pontuacao)
print (stem)
```

# POS

- Processo responsável por anotar a classe gramatical de cada palavra presente no texto

```
nltk.download('averaged_perceptron_tagger')
nltk.help.upenn_tagset()


from nltk.tag import pos_tag, pos_tag_sents
Texto = '''
pos= nltk.pos_tag(palavras_sem_pontuacao)
print (pos)
```

# RECONHECIMENTO DE ENTIDADES

- É possível identificar entidades nomeadas (Nomes próprios) no texto.

```
nltk.download('words')
nltk.download ("maxent_ne_chunker")

texto_en = "O campus muriaé fica localizado na monteiro de castro“
token3 = word_tokenize(texto_en, language='portuguese')
tags = pos_tag (token3)
en = nltk.ne_chunk(tags)
print(en)
```

# EXTRA: NORMALIZADOR

- O normalizador "arruma" as abreviações presentes nas sentenças. Muito importante quando o texto trabalhado não é oriundo de ambiente formais. Ex.: Chat
- Essa biblioteca deve ser instalada.
- Não faz parte do NLTK

```
!pip install enelvo
from enelvo.normaliser import Normaliser
norm = Normaliser(tokenizer='readable')
msg = 'Até hj vc n me respondeu. Oq aconteceu?'
resposta = norm.normalise(msg)
print(resposta)
```