Relatório 1 - Prática: Fundamentos de NLP (I)

Vitor Eduardo de Lima Kenor

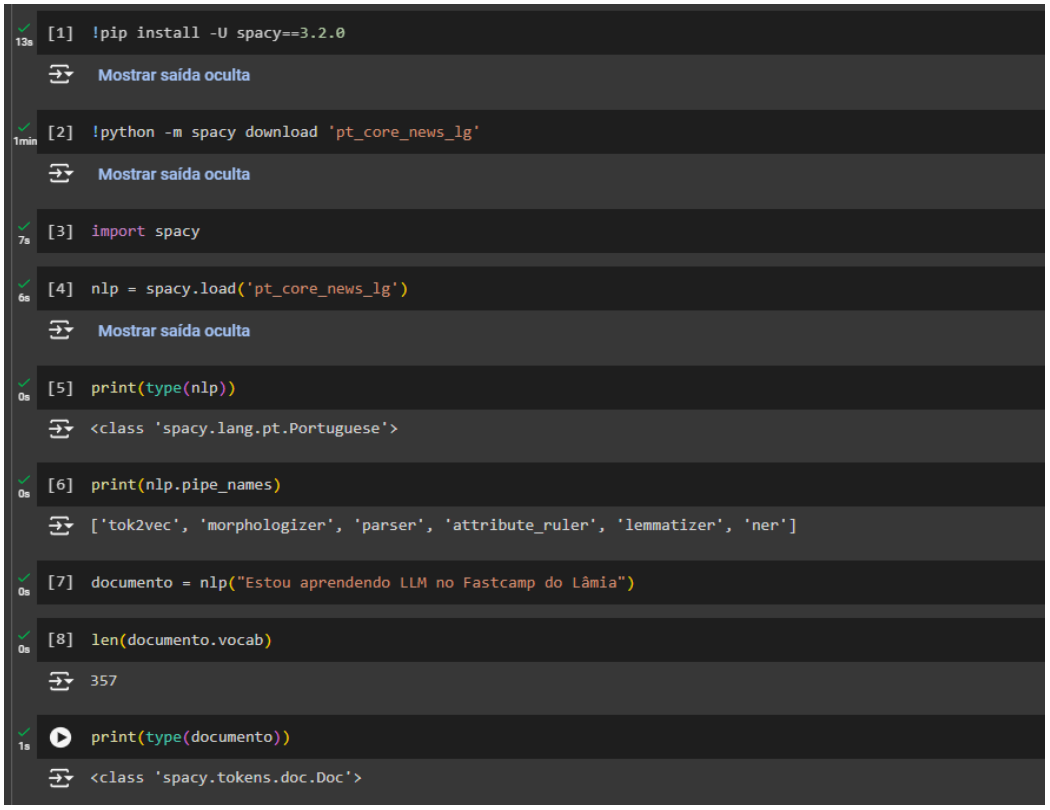
Descrição da atividade

O card primeiramente nos apresenta a um vídeo que conta de maneira simples e rápida o que é, e como funciona o processamento de linguagem natural (Natural Processing Language - NLP). Nele é passado rapidamente uma explicação dos métodos utilizados para fazer o modelo compreender e processar o texto com maior precisão. São eles: Segmentação, Tokenização, Par de Palavras, Lematização, Marcação de Classe Gramatical e Marcação de Entidade Nomeada.

O artigo presente neste card nos conta um pouco da história dos modelos de aprendizado e como foram usados para o NLP. Podemos ver as evoluções e as datas em que cada um dos modelos foi utilizado até chegar em 2018 com o modelo pré-treinado.

No card também temos 4 seções de um curso da Udemy para analisarmos. A primeira seção é a introdução de como o curso é estruturado e quais conteúdos serão abordados. Na segunda seção, somos apresentados a conceitos básicos. São eles: Corpus, Annotations, Tokenization, Parts-of-Speech Tagging (POS), Lemmatizing, Stemming, Dependency Parsing, Ngram, Modelo. É abordado um pouco sobre o que é Word Embedding e alguns tipos de se fazer. A seção dois finaliza explicando como é uma pipeline de um projeto de uma maneira geral para fins de entender a estrutura.

Na seção de número três do card, somos apresentados a como usar o NLP com a biblioteca Spacy do python. A parte do código é ensinada no google colab, a primeira aula da seção começamos baixando a versão 3.2.0 da biblioteca spacy no colab. Também baixamos um modelo pré-treinado de tamanho grande, e depois disso importamos a biblioteca Spacy. Depois disso criamos o objeto NLP com o modelo pré-treinado instalado anteriormente e começamos a fazer testes e entender como ele funciona.



```
[1] !pip install -U spacy==3.2.0
    Mostrar saída oculta

[2] !python -m spacy download 'pt_core_news_lg'
    Mostrar saída oculta

[3] import spacy

[4] nlp = spacy.load('pt_core_news_lg')
    Mostrar saída oculta

[5] print(type(nlp))
    <class 'spacy.lang.pt.Portuguese'>

[6] print(nlp.pipe_names)
    ['tok2vec', 'morphologizer', 'parser', 'attribute_ruler', 'lemmatizer', 'ner']

[7] documento = nlp("Estou aprendendo LLM no Fastcamp do Lâmia")

[8] len(documento.vocab)
    357

    print(type(documento))
    <class 'spacy.tokens.doc.Doc'>
```

Na segunda aula da seção colocamos a mão na massa para entender sobre tokens, inicialmente na aula temos a explicação e depois a parte prática de tokens no Spacy.

```
print(documento[6])
```
```
Lâmia
```
```
print(documento[3:8])
```
```
no Fastcamp do Lâmia.
```
```
print(len(documento))
```
```
8
```
```
print('tokens: ', [tokens.text for tokens in documento])
print('Stop Word: ', [tokens.is_stop for tokens in documento])
print('Alfanumérico: ', [tokens.is_alpha for tokens in documento])
print('Maísculo: ', [tokens.is_upper for tokens in documento])
print('Pontuação: ', [tokens.is_punct for tokens in documento])
print('Número: ', [tokens.like_num for tokens in documento])
print('Sentença Inicial: ', [tokens.is_sent_start for tokens in documento])
print('Formato: ', [tokens.shape_ for tokens in documento])
```
```
tokens:  ['Estou', 'aprendendo', 'LLM', 'no', 'Fastcamp', 'do', 'Lâmia', '.']
Stop Word:  [True, False, False, True, False, True, False, False]
Alfanumérico:  [True, True, True, True, True, True, True, False]
Maísculo:  [False, False, True, False, False, False, False, False]
Pontuação:  [False, False, False, False, False, False, False, True]
Número:  [False, False, False, False, False, False, False, False]
Sentença Inicial:  [True, False, False, False, False, False, False, False]
Formato:  ['Xxxxx', 'xxxx', 'XXX', 'xx', 'Xxxxx', 'xx', 'Xxxxx', '.']
```
```
for token in documento:
  if token.is_punct:
    print('Puntuação encontrada:', token.text)
  if token.is_stop:
    print('Stop word encontrado:', token.text)
```
```
Stop word encontrado: Estou
Stop word encontrado: no
Stop word encontrado: do
Puntuação encontrada: .
```

A terceira aula é sobre POS Tagging e Dependências, novamente com uma explicação e uma prática no código.

```
for token in documento:
    print(token.text, '-', token.pos_, '-', token.dep_, '-', token.lemma_, '-', token.shape_)
```
```
Estou - AUX - aux - Estou - Xxxxx
aprendendo - VERB - ROOT - aprender - xxxx
LLM - PROPN - obj - LLM - XXX
no - ADP - case - o - xx
Fastcamp - PROPN - obl - Fastcamp - Xxxxx
do - ADP - case - do - xx
Lâmia - PROPN - nmod - Lâmia - Xxxxx
. - PUNCT - punct - . - .
```
```
for token in documento:
  print(token.text, '-', token.morph)
```
```
Estou - Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin
aprendendo - VerbForm=Ger
LLM - Gender=Fem|Number=Sing
no - Definite=Def|Gender=Masc|Number=Sing|PronType=Art
Fastcamp - Gender=Masc|Number=Sing
do - Definite=Def|Gender=Masc|Number=Sing|PronType=Art
Lâmia - Number=Sing
. -
```

A quarta aula é breve e fala sobre as entidades nomeadas, temos a explicação e vimos que o modelo baixado já faz essa classificação e temos uma tabela mostrando o que cada sigla quer dizer. Por exemplo, a sigla GPE se trata de entidades nomeadas como países, cidades e estados. Já na quinta aula aprendemos do que se tratam as stop words e como retirá-las de seu texto pois atrapalham os modelos.

```
token_lista = []
for token in documento:
  token_lista.append(token.text)

stop_lista = []
for words in nlp.Defaults.stop_words:
  stop_lista.append(words)

semstop = [word for word in token_lista if not word in stop_lista]

print(documento.text)
print(semstop)
```

```
Estou aprendendo LLM no Fastcamp do Lâmia.
['Estou', 'aprendendo', 'LLM', 'Fastcamp', 'Lâmia', '.']
```

Depois aprendemos a buscar similaridade entre as palavras, quanto mais próximo de 1 mais semelhante e quanto mais perto de 0 menos semelhante. Logo adiante aprendemos o Matching, que se trata de buscar padrões dentro de um Doc.

```
[22] documento1 = nlp('Estou terminando o Fastcamp')
     documento2 = nlp('Não estou terminando o Fastcamp')
     print(documento1.similarity(documento2))
```

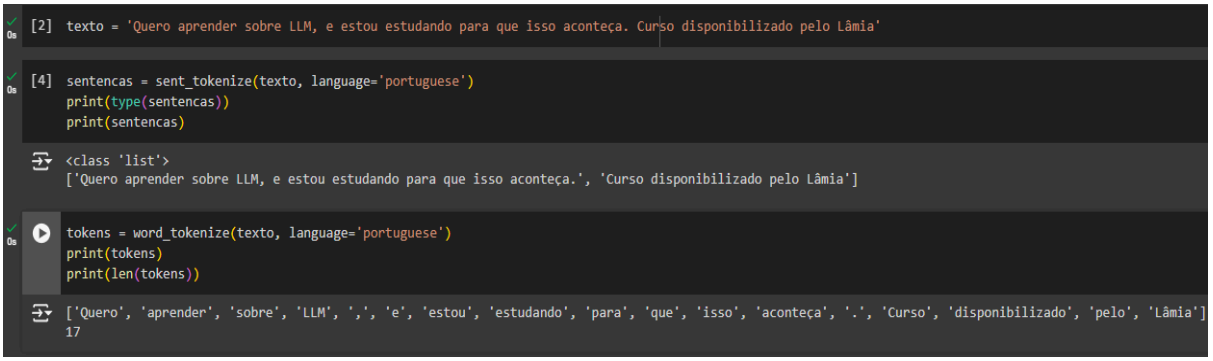
```
0.8416532925870895
```

```
from typing import Match
from spacy.matcher import Matcher

documento3 = nlp("Você quer ligar (44) - 9983229751 ou (32) 6581250275")

matcher = Matcher(nlp.vocab)
padrao = [{'ORTH': '('}, {'SHAPE': 'dd'}, {'ORTH': ')'}, {'ORTH': '-', 'OP': '?'}, {'IS_DIGIT': True}]
matcher.add('telefone', [padrao])
matches = matcher(documento3)

for id, inicio, fim in matches:
  print(documento3[inicio:fim])
```

```
(44) - 9983229751
(32) 6581250275
```

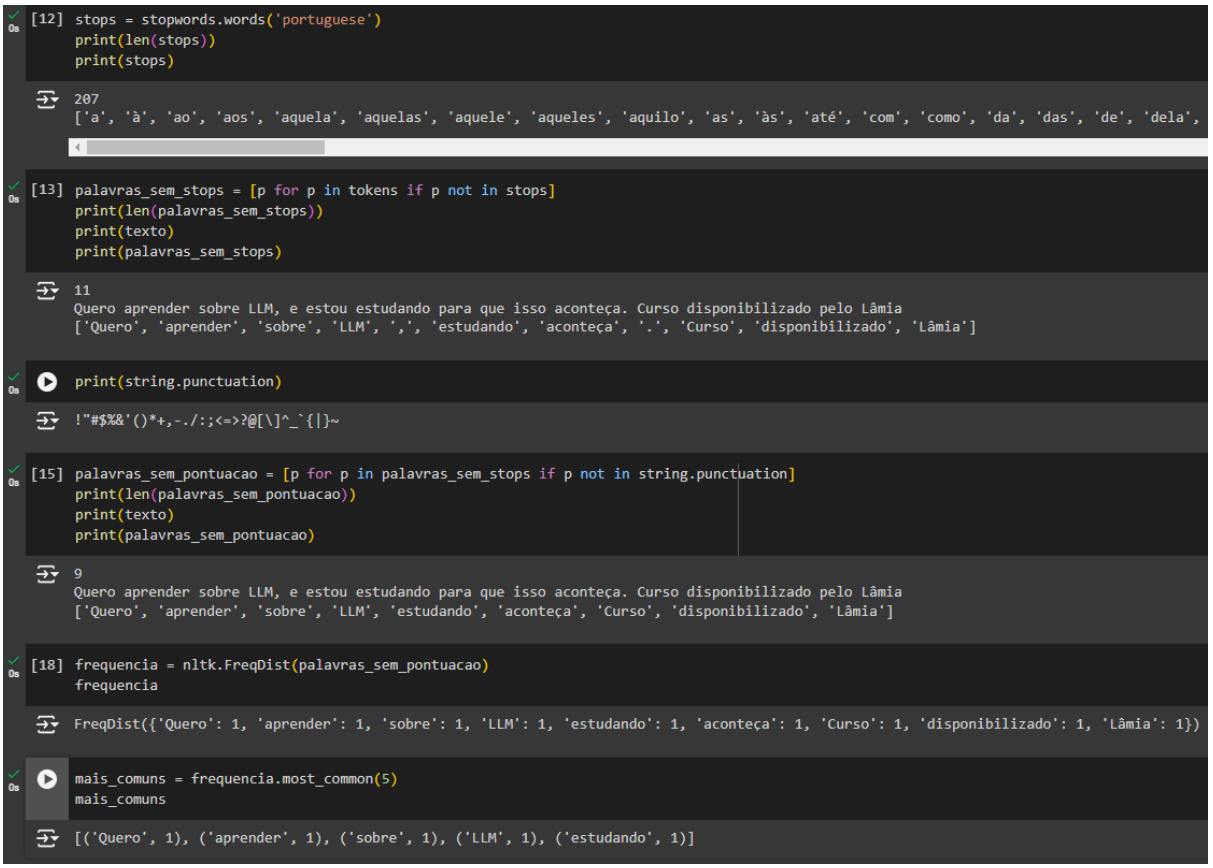
Nas últimas aulas da seção é ensinado sobre o display, que se trata de um módulo do Spacy para visualização. Os dois tipos de visualização deste módulo é para entidades nomeadas e dependência. Finaliza-se a seção falando sobre pipeline, apresenta como é a pipeline padrão do Spacy e como remover e adicionar etapas na pipeline.

Na quinta e última seção do card usaremos a biblioteca NLTK para o processamento de linguagem natural. Primeira aula iniciamos o ambiente no Colab, baixando tudo que será utilizado durante a seção. Vários pré processamentos que aconteciam automaticamente no Spacy não acontecem da mesma maneira com o NLTK, fazemos alguns processos manualmente.



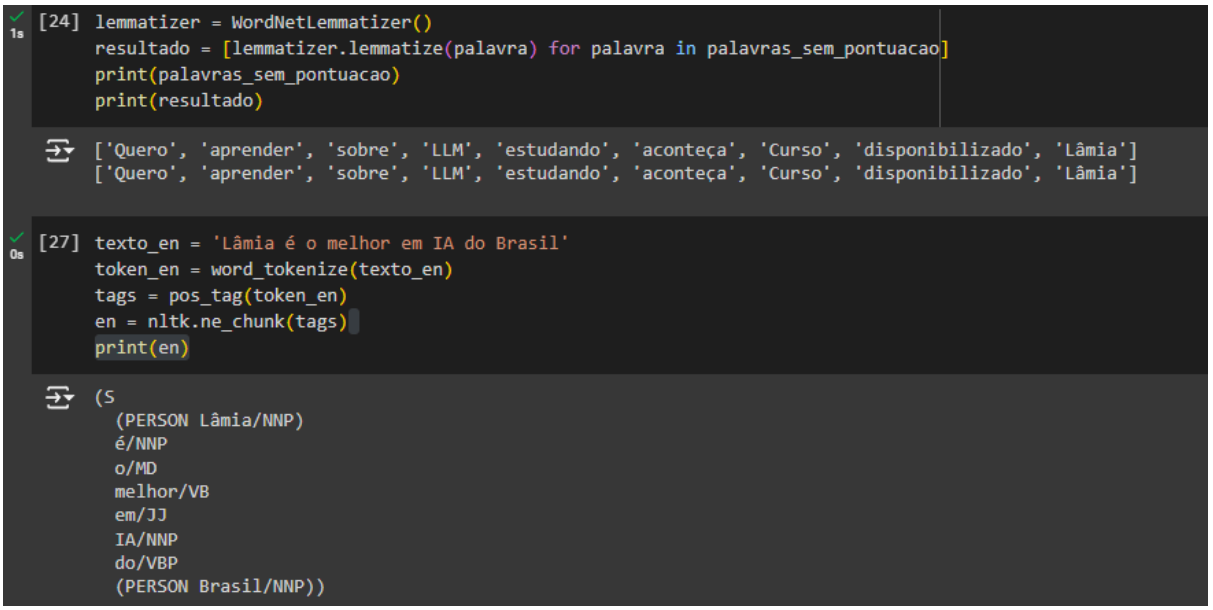
```
[2] texto = 'Quero aprender sobre LLM, e estou estudando para que isso aconteça. Curso disponibilizado pelo Lâmia'

[4] sentencas = sent_tokenize(texto, language='portuguese')
    print(type(sentencas))
    print(sentencas)

    <class 'list'>
    ['Quero aprender sobre LLM, e estou estudando para que isso aconteça.', 'Curso disponibilizado pelo Lâmia']

    tokens = word_tokenize(texto, language='portuguese')
    print(tokens)
    print(len(tokens))

    ['Quero', 'aprender', 'sobre', 'LLM', ',', 'e', 'estou', 'estudando', 'para', 'que', 'isso', 'aconteça', '.', 'Curso', 'disponibilizado', 'pelo', 'Lâmia']
    17
```

Depois de ver como funciona o sistema de tirar palavras com pontuação e Stopwords no NLTK, aprendemos a como ver as palavras com mais frequência no texto determinado.



```
[12] stops = stopwords.words('portuguese')
     print(len(stops))
     print(stops)

     207
     ['a', 'à', 'ao', 'aos', 'aquela', 'aquelas', 'aquele', 'aqueles', 'aquilo', 'as', 'às', 'até', 'com', 'como', 'da', 'das', 'de', 'dela',

[13] palavras_sem_stops = [p for p in tokens if p not in stops]
     print(len(palavras_sem_stops))
     print(texto)
     print(palavras_sem_stops)

     11
     Quero aprender sobre LLM, e estou estudando para que isso aconteça. Curso disponibilizado pelo Lâmia
     ['Quero', 'aprender', 'sobre', 'LLM', ',', 'estudando', 'aconteça', '.', 'Curso', 'disponibilizado', 'Lâmia']

     print(string.punctuation)

     !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

[15] palavras_sem_pontuacao = [p for p in palavras_sem_stops if p not in string.punctuation]
     print(len(palavras_sem_pontuacao))
     print(texto)
     print(palavras_sem_pontuacao)

     9
     Quero aprender sobre LLM, e estou estudando para que isso aconteça. Curso disponibilizado pelo Lâmia
     ['Quero', 'aprender', 'sobre', 'LLM', 'estudando', 'aconteça', 'Curso', 'disponibilizado', 'Lâmia']

[18] frequencia = nltk.FreqDist(palavras_sem_pontuacao)
     frequencia

     FreqDist({'Quero': 1, 'aprender': 1, 'sobre': 1, 'LLM': 1, 'estudando': 1, 'aconteça': 1, 'Curso': 1, 'disponibilizado': 1, 'Lâmia': 1})

     mais_comuns = frequencia.most_common(5)
     mais_comuns

     [('Quero', 1), ('aprender', 1), ('sobre', 1), ('LLM', 1), ('estudando', 1)]
```

Fechamos a seção aprendendo como fazer lemmatizer, busca de entidades nomeadas e criar pós-tagin com o NLTK.



```
[24] lemmatizer = WordNetLemmatizer()
     resultado = [lemmatizer.lemmatize(palavra) for palavra in palavras_sem_pontuacao]
     print(palavras_sem_pontuacao)
     print(resultado)

['Quero', 'aprender', 'sobre', 'LLM', 'estudando', 'aconteça', 'Curso', 'disponibilizado', 'Lâmia']
['Quero', 'aprender', 'sobre', 'LLM', 'estudando', 'aconteça', 'Curso', 'disponibilizado', 'Lâmia']

[27] texto_en = 'Lâmia é o melhor em IA do Brasil'
     token_en = word_tokenize(texto_en)
     tags = pos_tag(token_en)
     en = nltk.ne_chunk(tags)
     print(en)

(S
  (PERSON Lâmia/NNP)
  é/NNP
  o/MD
  melhor/VB
  em/JJ
  IA/NNP
  do/VBP
  (PERSON Brasil/NNP))
```

Conclusões

A Partir deste card podemos aprender toda a base e conceitos de como funciona o processamento de linguagem natural. Tivemos a explicação de cada etapa de pré processamento e vimos métodos para extrair ou retirar informações dos textos. Também adquirimos conhecimentos práticos nas bibliotecas de NLP Spacy e NLTK.

Referências

Vídeo de introdução ao NLP:
Natural Language Processing In 5 Minutes | What Is NLP And How Does It Work? | Si…

Artigo apresenta os modelos de aprendizagem de máquina para NLP:
https://medium.com/@harishdatalab/machine-learning-models-for-nlp-ff4010e7dd06

Curso da Udemy:
https://www.udemy.com/course/formacao-processamento-de-linguagem-natural-nlp/?couponCode=KEEPLEARNING