MATE34 – PLN

Atividade 01 – REN/Expressões Regulares

Agosto, 2022

Aluno: Anderson Boa Morte

PGCOMP
Universidade Federal da Bahia

Escopo

- No contexto de PLN (Processamento de Linguagem Natural), realizar um experimento de REN (Reconhecimento de Entidades Nomeadas).
- O experimento consiste em reconhecer a entidade **Pessoa** em textos de contexto geral escritos na Língua Portuguesa.

Descrição do experimento

- Utilizamos a biblioteca NLP **spaCy** para realização do experimento.
  - O spaCy possui 80 modelos pré-treinados para 24 idiomas, incluindo 3 modelos para a Língua Portuguesa. Utilizamos o modelo pt_core_news_sm.
- Utilizamos a biblioteca python re (*regular expression*) para combinar o uso de expressões regulares com o NER (*Named Entity Recognition*) do spaCy.
- Apresentamos adiante trechos de código de exemplo.

Modelo pt_core_news_sm

- Características do modelo:
  - **Tipo**: core *(vocabular, syntax, entities, vectors)*
  - **Gênero**: texto escrito (*news*, *media*)
  - **Tamanho**: sm (12 MB)
  - **Componentes**: tok2vec, morphologizer, parser, lemmatizer, ner etc
  - **Vetores**: 500 mil chaves, 500 mil vetores exclusivos (300 dimensões)
  - **Dataset** de treino inclui:
    - *Corpus* Bosque (9.368 frases)

  Fonte: https://spacy.io/models/pt

Código-exemplo 1

```
import spacy

# Carrega o modelo pre-treinado da língua Portuguesa do spaCy
nlp = spacy.load('pt_core_news_sm')

text = 'Foi arrasador quando primeiro descobrimos que nosso manuscrito de Galileu na realidade não é de Galileu,' \
    ' disse em entrevista a diretora interina das bibliotecas da universidade, Donna L. Hayward. Mas como a fina' \
    ' lidade... E esse é um teste para verificar se o spaCy consegue captuar o título do Dr. Jucelino. ' \
    'e um pouco mais de texto para identificar o reconhecimento de entidade localidade João Pessoa-PB, Salvador,' \
    'Ilhéus e Itaparica.' \

doc = nlp(text)

for ent in doc.ents:
    print(ent.text, ent.label_)
```

Saída:

```
Galileu PER
Galileu PER
Donna L. PER
Dr. Jucelino MISC
João Pessoa-PB, Salvador PER
Ilhéus LOC
Itaparica LOC
```

Colab:

https://colab.research.google.com/drive/1j8VjP-QL12v9l8e3Gz9-jXzNW6JNnY_P?usp=sharing

Código-exemplo 2

```
import spacy

# Carrega o modelo pre-treinado da língua Portuguesa do spaCy
nlp = spacy.load('pt_core_news_sm')

text = 'Foi arrasador quando primeiro descobrimos que nosso manuscrito de Galileu na realidade não é de Galileu,' \
    ' disse em entrevista a diretora interina das bibliotecas da universidade, Donna L. Hayward. Mas como a fina' \
    ' lidade... E esse é um teste para verificar se o spaCy consegue captuar o título do Dr. Jucelino. ' \
    'e um pouco mais de texto para identificar o reconhecimento de entidade localidade João Pessoa-PB, Salvador,' \
    'Ilhéus e Itaparica.' \

doc = nlp(text)

for ent in doc.ents:
    print(ent.text, ent.label_)

options = {"compact": True, "bg": "#09a3d5",
           "color": "white", "font": "Source Sans Pro"}

spacy.displacy.serve(doc, style='ent', options=options)
```

Saída:

Colab: https://colab.research.google.com/drive/1j8VjP-QL12v9l8e3Gz9-jXzNW6JNnY_P?usp=sharing

Código-exemplo 3

```python
import spacy
from spacy.language import Language
from spacy.tokens import Span

nlp = spacy.load("pt_core_news_sm")

@Language.component("expand_person_entities")
def expand_person_entities(doc):
    new_ents = []
    for ent in doc.ents:
        if (ent.label_ == "PERSON" or ent.label_ == "MISC") and ent.start != 0:
            prev_token = doc[ent.start - 1]
            if prev_token.text in ("Dr", "Dr.", "Mr", "Mr.", "Ms", "Ms."):
                new_ent = Span(doc, ent.start - 1, ent.end, label=ent.label)
                new_ents.append(new_ent)
        else:
            new_ents.append(ent)
    doc.ents = new_ents
    return doc

# Add the component after the named entity recognizer
nlp.add_pipe("expand_person_entities", after="ner")

doc = nlp("Dr. Alex Smith chaired first board meeting of Acme Corp Inc.")
print([(ent.text, ent.label_) for ent in doc.ents])
```

Colab: https://colab.research.google.com/drive/1j8VjP-QL12v9l8e3Gz9-jXzNW6JNnY_P?usp=sharing

Código-exemplo 4

```
import spacy

nlp = spacy.load('pt_core_news_sm')
ruler = nlp.add_pipe("entity_ruler")

patterns = [{"label": "PERSONA",
             "pattern": [{"TEXT": {"REGEX": r"\d{3}"}}]
            }]

ruler.add_patterns(patterns)

doc = nlp("This is Fred and his number is 123 to get an apple  pie")
for ent in doc.ents:
    print(ent.text, ent.label_)
```

Colab: https://colab.research.google.com/drive/1j8VjP-QL12v9l8e3Gz9-jXzNW6JNnY_P?usp=sharing

Código-exemplo 5

Usando Expressões Regulares para o reconhecimento de entidade nomeadas Pessoa e Localidade

In [1]:
```
import re
```

In [2]:
```
def multi_re_find(patterns, phrase):
    '''
    Pega uma lista de padrões regex
    Imprime a lista de todos os matches
    '''
    for pattern in patterns:
        print('Procurando a frase usando o re check: %r' %pattern)
        print(re.findall(pattern, phrase))
        print('\n')
```

In [42]:
```
test_phrase = 'Antonio Carlos da Silva Jobim'
'''
Regex tips, vide https://regex101.com/ :
- ^: início da string
- [a-zA-Z ]: caracteres na faixa a-z ou A-Z e espaço
- {2,30}: mínimo de 2 e máximo de 30 caracteres
- $: fim da string
'''

test_patterns=[ r'^[a-zA-Z ]{2,30}$']

multi_re_find(test_patterns,test_phrase)
```

```
Procurando a frase usando o re check: '^[a-zA-Z ]{2,30}$'
['Antonio Carlos da Silva Jobim']
```

Github: https://github.com/abmorte/MATE34/blob/main/atividade1_regex.ipynb

Referências

- https://spacy.io/
- https://ner.pythonhumanities.com/02_02_intro_to_regex.html

Perguntas?

Contatos:
Anderson Boa Morte
andersonmorte@ufba.br