Universidade Federal de Pelotas
Disciplina de Processamento da Linguagem Natural

Grupo: Geovana Silveira e Alessandra Galvao

Trabalho de Entidades Nomeadas, Relações e Correferência

Parte 1: CRF para NER

1. **Dataset Leis:**

   Para a realização desta parte do trabalho 1 utilizando o dataset das leis, foi preciso primeiramente organizar o dataset txt fornecido para o formato do arquivo (ner_dataset.csv) do código base. Como o dataset original possuía apenas a Word e a Tag, foi necessário utilizar o modelo em português da biblioteca do Spacy para obter o part-of-speech(POS) das palavras do dataset. Na figura 1 é possível observar as alterações realizadas no dataset original.

	Sentence #	Word	POS	Tag
0	Sentence: 1	EMENTA	VERB	O
1	Sentence: 1	:	PUNCT	O
2	Sentence: 1	APELAÇÃO	PROPN	O
3	Sentence: 1	CÍVEL	PROPN	O
4	Sentence: 1	-	PUNCT	O
5	Sentence: 1	AÇÃO	PROPN	O
6	Sentence: 1	DE	ADP	O
7	Sentence: 1	INDENIZAÇÃO	PROPN	O
8	Sentence: 1	POR	ADP	O
9	Sentence: 1	DANOS	NOUN	O

Figura 1: Dataset das leis após as alterações realizadas.

Em relação às features adicionadas para este dataset, escolhemos adicionar 3 features novas: ***isArtigo***, que representa um termo muito utilizado em textos relacionados às leis, ***isNumRomano***, que representa os números romanos presentes no dataset e que são utilizados para numerar artigos e a feature ***isSymbol***, que representa os símbolos utilizados no dataset.

Com a adição destas features foi possível observar uma melhora nos resultados em comparação ao resultados obtidos pelo arquivo do código base. As figuras 2 e 3 ilustram os resultados obtidos após a adição das features.

```
[158] f1_score = flat_f1_score(y_test, y_pred, average = 'weighted')
      print(f1_score)

      0.9847088625463212
```

Figura 2: f1_score obtido após a adição das features criadas

	precision	recall	f1-score	support
B-JURISPRUDENCIA	0.93	0.79	0.85	127
B-LEGISLACAO	0.93	0.90	0.92	265
B-LOCAL	0.87	0.77	0.81	43
B-ORGANIZACAO	0.92	0.89	0.91	260
B-PESSOA	0.96	0.91	0.93	213
B-TEMPO	0.94	0.88	0.91	165
I-JURISPRUDENCIA	0.93	0.84	0.89	301
I-LEGISLACAO	0.97	0.94	0.96	1244
I-LOCAL	0.87	0.88	0.87	51
I-ORGANIZACAO	0.94	0.92	0.93	456
I-PESSOA	0.98	0.95	0.96	431
I-TEMPO	0.93	0.91	0.92	139
O	0.99	1.00	0.99	23912
accuracy			0.98	27607
macro avg	0.94	0.89	0.91	27607
weighted avg	0.98	0.98	0.98	27607

Figura 3: Relatório de classificação obtido pelo dataset de Leis

Link do código - Dataset Leis :
https://colab.research.google.com/drive/1qtx1EYWE2mUCFnUf31Ej-5IqRWffWg8o

2. **Dataset Tweets:**

Para obter as mensagens do tweets foi preciso criar uma aplicação no twitter (https://developer.twitter.com/apps) para obter as chaves de acesso para a extração das mensagens. Foi utilizado também a biblioteca Tweepy para a busca dos tweets.

Assim como no dataset das leis, o dataset dos tweets também passou por modificações para ficar com o mesmo formato do arquivo do código base. A biblioteca do spacy foi utilizada novamente para conseguir o POS.

	Sentence #	Word	POS	Tag
0	Sentence 1	@ForeverPlayerG	PROPN	O
1	Sentence 1	tem	AUX	O
2	Sentence 1	que	SCONJ	O
3	Sentence 1	ser	VERB	O
4	Sentence 1	,	PUNCT	O
5	Sentence 1	essa	DET	O
6	Sentence 1	o	DET	O
7	Sentence 1	áudio	NOUN	O
8	Sentence 1	tava	VERB	O
9	Sentence 1	ruim	NOUN	O

Figura 4: Dataset dos tweets após as alterações realizadas.

Foi criada três features: ***mention***, que verifica palavras que iniciam com arroba**(@)**, ***hashtag***, que verifica a presença de palavras que possuem um hashtag(#) no começo da frase e a feature ***url***, que verifica a presença de palavras que iniciam com uma url (https://'). A adição destas features representou uma melhoria significativa se comparado com o resultado do dataset de leis como se pode observar através das figuras 5 e 6.

```
f1_score = flat_f1_score(y_test, y_pred, average = 'weighted')
print(f1_score)
```

```
0.9958441208980108
```

Figura 5: f1_score obtido após as features criadas para o dataset de tweets

	precision	recall	f1-score	support
B-location	0.56	0.39	0.46	588
B-organization	0.60	0.38	0.47	417
B-person	0.66	0.46	0.54	957
I-location	0.78	0.62	0.69	506
I-organization	0.00	0.00	0.00	70
I-person	0.37	0.20	0.26	256
O	1.00	1.00	1.00	521860
accuracy			1.00	524654
macro avg	0.57	0.43	0.49	524654
weighted avg	1.00	1.00	1.00	524654

Figura 6: Relatório de classificação obtido pelo dataset de tweets

Link do código - Dataset Tweets:
https://colab.research.google.com/drive/1kQzo1q4FnkP71VDhMYCb-loelAuSMRR5

Parte 2: Extração de Relações

Para a extração das relações, foi preciso primeiramente obter a sequência do texto do dataset e identificar as entidades nomeada da sequência, para isso foi utilizado a função ne_chunk da biblioteca nltk, que classifica as palavras em três categorias: PERSON, ORGANIZATION e GPE (geo-political entity).
Depois da identificação das entidades, foi utilizando três expressões regulares:

```
IN = re.compile(r'.*\bde\b(?!\b.+ndo)')
IN2 = re.compile(r'.*\bpor\b(?!\b.+ndo)')
IN3 = re.compile(r'.*\bem\b(?!\b.+ndo)')
```

Figura 7: Expressões regulares criadas para a extração das relações

Estas expressões foram utilizadas para extrair relações entre as categorias de entidades e as expressões “de”, ”por” e “em”. A função *extract_rels()* do nltk foi utilizada para extrair as relações existentes entre o par de categorias de entidades nomeadas e sequência de palavras entre as categorias.

```
for rel in nltk.sem.extract_rels('PERSON','ORGANIZATION', ne, corpus='ace', pattern=IN):
    print(nltk.sem.rtuple(rel))

for rel in nltk.sem.extract_rels('ORGANIZATION','ORGANIZATION', ne, corpus='ace', pattern=IN):
    print(nltk.sem.rtuple(rel))

for rel2 in nltk.sem.extract_rels('GPE','ORGANIZATION', ne, corpus='ace', pattern=IN):
    print(nltk.sem.rtuple(rel2))

for rel in nltk.sem.extract_rels('ORGANIZATION','PERSON', ne, corpus='ace', pattern=IN2):
    print(nltk.sem.rtuple(rel))

for rel in nltk.sem.extract_rels('PERSON','ORGANIZATION', ne, corpus='ace', pattern=IN2):
    print(nltk.sem.rtuple(rel))

for rel in nltk.sem.extract_rels('PERSON','ORGANIZATION', ne, corpus='ace', pattern=IN3):
    print(nltk.sem.rtuple(rel))

for rel in nltk.sem.extract_rels('ORGANIZATION','ORGANIZATION', ne, corpus='ace', pattern=IN3):
    print(nltk.sem.rtuple(rel))

for rel in nltk.sem.extract_rels('PERSON','GPE', ne, corpus='ace', pattern=IN3):
    print(nltk.sem.rtuple(rel))
```

Figura 8: Categorias de entidades nomeadas para extrair as relações

Link do código:
https://colab.research.google.com/drive/1Ov_NZurLWIfPG_07sUjP6QYH_W81X31j