Relatório - Daniel da Silva Costa

Introdução

O presente relatório foi escrito como um dos requisitos do Trabalho 2 - POS e Transfer Learning, da disciplina de Processamento de Linguagem Natural (2022.1) da UFF.

Tendo em vista a tarefa de *Part-of-Speech Tagging (POS Tagging)*, foram avaliadas 4 técnicas em um corpus de língua portuguesa.

A atividade envolveu construir um *POS tagger* para a língua portuguesa, usando *fine-tuning* a partir de um modelo de *Transformers* pré-treinado em português.

O modelo de Transformers escolhido foi o BERTimbau[1] na sua versão *Base*[2] e foi acessado através do Hugging Face[3], que é um repositório de modelos pré-treinados de Aprendizado Profundo.

O modelo BERTimbau pré-treinado passou por um fine-tuning e os resultados obtidos foram comparados com outros três modelos, a saber:

1 - Um modelo LSTM sem um vetor pré-treinado;

2 - Um modelo GRU sem um vetor pré-treinado; e

3 - Uma ferramenta *off-the-shelf*: spaCy[4]. O spaCy é uma ferramenta voltada para a utilização produtiva no mercado industrial e, sendo assim, basta utilizar as ferramentas presentes em seu ecossistema.

[1] BERTimbau: https://huggingface.co/neuralmind
[2] BERTimbau Base: https://huggingface.co/neuralmind/bert-base-portuguese-cased
[3] Hugging Face: https://huggingface.co/
[4] spaCy: https://spacy.io/

Dataset

Em todas as técnicas foi utilizado o dataset Universal Dependencies (UD) Portuguese Bosque[1] nas suas versões de treinamento e de desenvolvimento. Este dataset é um *treebank* (floresta sintática) em português e traz, além de sentenças, a categoria (POS) de cada *token* (palavra) em cada sentença. Este dataset é composto de três *sub-datasets*: treinamento, desenvolvimento e teste. Neste estudo, foram utilizados os datasets de treinamento e de desenvolvimento. Os modelos LSTM, GRU e BERTimbau foram treinados no dataset de treinamento e avaliados no dataset de desenvolvimento.

Um exemplo extraído do dataset UD pode ser visto abaixo. Foram destacados em negrito: a sentença, os tokens e o POS de cada token.

Exemplo de sentença extraída do dataset UD

 # newdoc_id = CF1
 # text = PT no governo
 # sent_id = CF1-1
 # source = CETENFolha n=1 cad=Opinião sec=opi sem=94a
1 ***PT*** *PT* ***PROPN*** *_ Gender=Masc|Number=Sing 0 root _ _*

2-3 no _ _ _ _ _ _ _ _
*2 **em** em **ADP** _ _ 4 case _ _*
*3 **o** o **DET** _ Definite=Def|Gender=Masc|Number=Sing|PronType=Art 4 det _ _*
*4 **governo** governo **NOUN** _ Gender=Masc|Number=Sing 1 nmod _ _*

Neste exemplo, observa-se:

(i) a sentença: “PT no governo”;

(ii) seus tokens: *PT*, *em*, *o*, *governo* e;

(iii) as categorias (POS) de cada token: *PROPN*, *ADP*, *DET* e *NOUN*.

O spaCy dispensa treinamento, mas precisamos indicar para ele o idioma a ser utilizado na ferramenta. Isso é feito mediante o carregamento de um *pipeline,* do idioma desejado, construído previamente por terceiros. Neste caso, foi utilizado o pt_core_news_lg[2].

Após o carregamento do pipeline em português, o spaCy foi aplicado no dataset UD de desenvolvimento. Os resultados estão disponíveis na seção Resultados.

[1] Universal Dependencies (UD) Portuguese Bosque: https://universaldependencies.org/treebanks/pt_bosque/index.html

[2] spaCy pt_core_news_lg: https://spacy.io/models/pt#pt_core_news_lg

Resultados

Os modelos GRU e LSTM foram treinados durante as seguintes quantidades de épocas: 1, 50 e 100 e o modelo BERTimbau por: 1, 5 e 10.

Os modelos GRU, LSTM e BERTimbau, foram treinados usando-se um *batch size* de tamanho 16, e para os três modelos os resultados foram extraídos apenas na última época. Todos os três modelos possuem arquitetura que permite o aprendizado bidirecional da representação dos tokens.

Sobre o processo de tokenização, não foi preciso ser feito nenhum procedimento específico, pois os datasets UDs já trazem os tokens separados, bastando percorrer o arquivo para construção do vocabulário e da lista de POS dos datasets de treinamento e de desenvolvimento. A exceção ficou com o modelo BERTimbau que necessita de um tokenizador específico, pois ele foi pré-treinado usando-se sub-palavras. Assim, o tokenizador pré-construído do BERTimbau também foi carregado durante a execução do programa.

No caso dos modelos GRU, LSTM e BERTimbau, as sentenças passaram ainda pelo processo de *padding*, onde foram acrescentados tokens com a significação de preenchimento (<pad>) para permitir que as sentenças fossem tratadas com o mesmo tamanho no momento do treinamento. Isso se deve à forma como as arquiteturas de redes neurais profundas são tradicionalmente construídas.

No caso do BERTimbau ainda foi preciso colocar um token [CLS] no início de cada sentença e um token [SEP] no final de cada sentença por ser a maneira como os dados de treinamento do BERTimbau foram pré-processados originalmente. O primeiro delimita o início da sentença e o último, o final.

Ainda foi utilizado um outro token especial chamado *unknown* (<unk>) durante a inferência dos dados do dataset de desenvolvimento. Isso é necessário porque, tendo em vista que os datasets de treinamento e de desenvolvimento são diferentes, podem haver palavras neste último

que não existem no primeiro e, portanto, o modelo pode não ter visto aquelas palavras durante o seu treinamento.

No caso da ferramenta spaCy, não foi necessário o tratamento com tokens especiais como o de padding (<pad>), pois a ferramenta não passou por treinamento.

Ainda no caso do spaCy, como ele não precisa de treinamento, os resultados se referem à aplicação desta ferramenta no dataset de desenvolvimento UD.

A seguir, são apresentados os resultados obtidos por cada modelo e o tempo de treinamento (*Running Time*). As acurácias (*Accuracy*) foram destacadas em negrito. As métricas foram obtidas usando-se a função classification_report() do Scikit-Learn[1].

GRU

Épocas: 1

	precision	recall	f1-score	support
<pad>	0.00	0.00	0.00	33153
ADJ	0.36	0.12	0.18	1157
ADP	0.89	0.93	0.91	3549
ADV	0.01	0.52	0.03	844
AUX	0.87	0.54	0.67	581
CCONJ	0.46	0.92	0.61	542
DET	0.90	0.92	0.91	3702
INTJ	0.00	0.00	0.00	3
NOUN	0.55	0.83	0.66	4415
NUM	0.96	0.12	0.21	461
PRON	0.79	0.65	0.71	835
PROPN	0.41	0.23	0.29	2143
PUNCT	0.95	0.99	0.97	3267
SCONJ	0.73	0.33	0.45	542
SYM	1.00	0.81	0.89	36
VERB	0.51	0.44	0.47	2166
X	0.00	0.00	0.00	19
_	0.97	0.92	0.94	1753
accuracy			0.32	59168
macro avg	0.58	0.52	0.50	59168
weighted avg	0.31	0.32	0.30	59168

F1: 0.3

Accuracy: 0.32

==========

Finished

Running Time: 0.76 minutes

Épocas: 50

	precision	recall	f1-score	support
<pad>	0.00	0.00	0.00	33153
ADJ	0.53	0.82	0.64	1157

ADP	0.96	0.97	0.96	3549
ADV	0.54	0.89	0.67	844
AUX	0.95	0.91	0.93	581
CCONJ	0.99	0.99	0.99	542
DET	0.96	0.98	0.97	3702
INTJ	0.00	0.00	0.00	3
NOUN	0.81	0.91	0.86	4415
NUM	0.94	0.74	0.83	461
PRON	0.92	0.88	0.90	835
PROPN	0.03	0.45	0.05	2143
PUNCT	1.00	1.00	1.00	3267
SCONJ	0.76	0.74	0.75	542
SYM	1.00	1.00	1.00	36
VERB	0.86	0.66	0.75	2166
X	0.00	0.00	0.00	19
_	0.99	0.97	0.98	1753
accuracy			0.38	59168
macro avg	0.68	0.72	0.68	59168
weighted avg	0.36	0.38	0.36	59168

F1: 0.36

Accuracy: 0.38

==========

Finished

Running Time: 33.0 minutes

Épocas: 100

	precision	recall	f1-score	support
<pad>	0.00	0.00	0.00	33153
ADJ	0.86	0.54	0.66	1157
ADP	0.96	0.95	0.96	3549
ADV	0.91	0.87	0.89	844
AUX	0.91	0.92	0.92	581
CCONJ	0.99	0.99	0.99	542
DET	0.96	0.96	0.96	3702
INTJ	0.00	0.00	0.00	3
NOUN	0.92	0.74	0.82	4415
NUM	0.90	0.74	0.81	461
PRON	0.90	0.86	0.88	835
PROPN	0.05	0.87	0.10	2143
PUNCT	1.00	1.00	1.00	3267
SCONJ	0.67	0.73	0.70	542
SYM	1.00	1.00	1.00	36
VERB	0.71	0.84	0.77	2166
X	0.00	0.00	0.00	19
_	0.99	0.97	0.98	1753
accuracy			0.38	59168
macro avg	0.71	0.72	0.69	59168

weighted avg	0.37	0.38	0.36	59168

F1: 0.36

Accuracy: 0.38

==========

Finished

Running Time: 67.0 minutes

LSTM

Épocas: 1

	precision	recall	f1-score	support
<pad>	0.00	0.00	0.00	33153
ADJ	0.36	0.13	0.19	1157
ADP	0.92	0.94	0.93	3549
ADV	0.74	0.47	0.58	844
AUX	0.85	0.62	0.72	581
CCONJ	0.99	0.92	0.95	542
DET	0.88	0.93	0.90	3702
INTJ	0.00	0.00	0.00	3
NOUN	0.57	0.83	0.68	4415
NUM	0.91	0.02	0.04	461
PRON	0.58	0.61	0.59	835
PROPN	0.47	0.22	0.30	2143
PUNCT	0.98	0.99	0.98	3267
SCONJ	0.65	0.27	0.38	542
SYM	1.00	0.50	0.67	36
VERB	0.04	0.67	0.08	2166
X	0.00	0.00	0.00	19
_	0.95	0.93	0.94	1753
accuracy			0.33	59168
macro avg	0.60	0.50	0.50	59168
weighted avg	0.31	0.33	0.30	59168

F1: 0.3

Accuracy: 0.33

==========

Finished

Running Time: 0.77 minutes

Épocas: 50

	precision	recall	f1-score	support
<pad>	0.00	0.00	0.00	33153
ADJ	0.61	0.73	0.66	1157
ADP	0.95	0.97	0.96	3549
ADV	0.90	0.87	0.88	844

AUX	0.84	0.92	0.88	581
CCONJ	0.99	0.98	0.99	542
DET	0.96	0.97	0.97	3702
INTJ	0.00	0.00	0.00	3
NOUN	0.90	0.84	0.87	4415
NUM	0.58	0.84	0.68	461
PRON	0.92	0.84	0.88	835
PROPN	0.04	0.60	0.07	2143
PUNCT	1.00	1.00	1.00	3267
SCONJ	0.67	0.70	0.69	542
SYM	0.97	1.00	0.99	36
VERB	0.73	0.84	0.78	2166
X	0.00	0.00	0.00	19
_	1.00	0.97	0.98	1753
accuracy			0.39	59168
macro avg	0.67	0.73	0.68	59168
weighted avg	0.36	0.39	0.37	59168

F1: 0.37

Accuracy: 0.39

==========

Finished

Running Time: 34.0 minutes

Épocas: 100

	precision	recall	f1-score	support
<pad>	0.00	0.00	0.00	33153
ADJ	0.73	0.71	0.72	1157
ADP	0.96	0.97	0.97	3549
ADV	0.73	0.89	0.80	844
AUX	0.92	0.94	0.93	581
CCONJ	0.99	0.99	0.99	542
DET	0.96	0.98	0.97	3702
INTJ	0.00	0.00	0.00	3
NOUN	0.89	0.86	0.87	4415
NUM	0.96	0.78	0.86	461
PRON	0.88	0.88	0.88	835
PROPN	0.66	0.81	0.73	2143
PUNCT	1.00	1.00	1.00	3267
SCONJ	0.79	0.70	0.74	542
SYM	1.00	1.00	1.00	36
VERB	0.87	0.72	0.79	2166
X	0.00	0.00	0.00	19
_	0.99	0.96	0.98	1753
accuracy			0.39	59168
macro avg	0.74	0.73	0.73	59168
weighted avg	0.40	0.39	0.39	59168

F1: 0.39

Accuracy: 0.39

==========

Finished

Running Time: 68.0 minutes

BERTimbau

Épocas: 1 (Fine-Tuning)

	precision	recall	f1-score	support
0	0.94	0.90	0.92	3122
1	0.91	0.92	0.91	838
3	0.00	0.00	0.00	11
4	0.92	0.96	0.94	503
5	0.87	0.88	0.87	515
6	0.93	0.96	0.95	4637
7	0.98	0.98	0.98	2889
8	0.93	0.89	0.91	348
9	0.00	0.00	0.00	46
10	0.96	0.97	0.97	2700
11	0.94	0.93	0.93	760
12	0.98	0.99	0.98	470
13	0.00	0.00	0.00	4
14	1.00	1.00	1.00	2671
15	0.89	0.90	0.89	1352
16	0.98	0.99	0.99	3082
17	0.00	0.00	0.00	0
accuracy			0.95	23948
macro avg	0.72	0.72	0.72	23948
weighted avg	0.95	0.95	0.95	23948

F1: 0.95

Accuracy: 0.95

==========

Finished

Running Time: 0.75 minutes

Épocas: 5 (Fine-Tuning)

	precision	recall	f1-score	support
0	0.99	0.99	0.99	470
1	0.99	0.99	0.99	2889
2	0.98	0.94	0.96	348
3	0.96	0.98	0.97	503
4	0.92	0.91	0.92	1352

5	0.00	0.00	0.00	4
7	0.99	0.99	0.99	3082
8	1.00	1.00	1.00	2671
9	0.95	0.96	0.96	838
10	1.00	1.00	1.00	11
11	0.96	0.99	0.97	760
12	0.96	0.96	0.96	4637
13	0.87	0.57	0.68	46
14	0.96	0.94	0.95	515
15	0.94	0.94	0.94	3122
16	0.97	0.99	0.98	2700
17	0.00	0.00	0.00	0
accuracy			0.97	23948
macro avg	0.85	0.83	0.84	23948
weighted avg	0.97	0.97	0.97	23948

F1: 0.97

Accuracy: 0.97

==========

Finished

Running Time: 3.7 minutes

Épocas: 10 (Fine-Tuning)

	precision	recall	f1-score	support
0	0.98	0.94	0.96	348
1	0.95	0.93	0.94	3122
2	0.97	0.99	0.98	760
3	1.00	1.00	1.00	11
4	0.00	0.00	0.00	4
5	0.63	0.63	0.63	46
6	0.97	0.98	0.98	2700
7	0.97	0.95	0.96	838
8	0.97	0.97	0.97	503
9	0.99	0.99	0.99	2889
10	0.96	0.97	0.96	4637
11	0.95	0.95	0.95	515
13	0.99	0.99	0.99	3082
14	1.00	1.00	1.00	2671
15	0.92	0.92	0.92	1352
16	1.00	0.99	0.99	470
accuracy			0.97	23948
macro avg	0.89	0.89	0.89	23948
weighted avg	0.97	0.97	0.97	23948

F1: 0.97

Accuracy: 0.97

==========

Finished

Running Time: 6.8 minutes

spaCy

	precision	recall	f1-score	support
ADJ	0.95	0.93	0.94	1345
ADP	0.98	0.98	0.98	4185
ADV	0.96	0.96	0.96	1005
AUX	0.97	0.99	0.98	651
CCONJ	0.98	0.98	0.98	650
DET	0.98	0.98	0.98	4354
INTJ	1.00	0.67	0.80	3
NOUN	0.97	0.96	0.97	5193
NUM	0.98	0.96	0.97	569
PRON	0.96	0.90	0.93	972
PROPN	0.92	0.96	0.94	2439
PUNCT	0.99	0.99	0.99	3844
SCONJ	0.84	0.91	0.87	638
SYM	0.97	0.97	0.97	36
VERB	0.98	0.98	0.98	2535
X	0.85	0.61	0.71	28
accuracy			0.97	28447
macro avg	0.96	0.92	0.93	28447
weighted avg	0.97	0.97	0.97	28447

F1: 0.97
Accuracy: 0.97

[1] Scikit-Learn: https://scikit-learn.org/stable/index.html

Análise e Discussão

Treinar os modelos GRU, LSTM e BERTimbau (*fine-tuning*) para uma época, levou aproximadamente o mesmo tempo, respectivamente: 0,76; 0,77 e 0,75. Mas os resultados para a medida F1 e para a acurácia (*Accuracy*) foram bem diferentes, conforme pode ser visto na tabela abaixo (o modelo com melhor resultado foi destacado em negrito):

Modelo	Número de Épocas	F1	Acurácia	Tempo de Treinamento (em minutos)
GRU	1	0,3	0,32	0,76
LSTM	1	0,3	0,33	0,77
BERTimbau	**1**	**0,95**	**0,95**	**0,75**

Como pode-se observar, o modelo BERTimbau obteve os melhores resultados considerando-se esses três modelos.

Quando comparado com o spaCy, o BERTimbau somente obteve resultado compatível quando treinado (*fine-tuning*) por 5 épocas, tendo sido treinado por apenas 3,7 minutos:

Modelo	Número de Épocas	F1	Acurácia	Tempo de Treinamento (em minutos)
BERTimbau	5	0,97	0,97	3,7
spaCy	-	0,97	0,97	-

O BERTimbau ainda foi treinado novamente por 10 épocas durante 6,8 minutos, mas o resultado não foi alterado.

Os modelos GRU e LSTM, foram treinados ainda por 50 e 100 épocas cada um, mas os resultados não mudaram substancialmente, conforme pode ser visto na seção Resultados e na tabela abaixo.

Modelo	Número de Épocas	F1	Acurácia	Tempo de Treinamento (em minutos)
GRU	50	0,36	0,38	33,0
GRU	100	0,36	0,38	67,0
LSTM	50	0,37	0,39	34,0
LSTM	100	0,39	0,39	68,0

Importante destacar que a versão do BERTimbau utilizada foi a *Base* e, possivelmente, se fosse utilizada a versão *Large*, o modelo teria alcançado resultados ainda melhores, porque o modelo *Large* foi pré-treinado com o dobro de camadas e o triplo de parâmetros, conforme pode ser visto na Figura 1.

Model	Arch.	#Layers	#Params
neuralmind/bert-base-portuguese-cased	BERT-Base	12	110M
neuralmind/bert-large-portuguese-cased	BERT-Large	24	335M

Figura 1: Modelos Disponíveis. Fonte: Extraído de https://huggingface.co/neuralmind/bert-large-portuguese-cased

Na avaliação dos resultados do spaCy, a acurácia permaneceu a mesma do indicado no site oficial do pipeline em português[1]: 0,97. Isso pode ser explicado, talvez, porque o pipeline em português carregado e utilizado no spaCy foi treinado em dois datasets, sendo um deles o UD Portuguese Bosque v2.8 conforme indicado em *Sources* na página do pipeline[2].

Assim, a ferramenta voltada para a indústria continua sendo uma excelente opção tendo em vista o caráter de produtividade que o mercado precisa. Contudo, o trabalho sugere que os resultados podem ser ainda melhores, possivelmente, com a utilização da versão *Large* do BERTimbau e com o treinamento durante mais épocas. Pode-se, inclusive, considerar-se a utilização de outros datasets no treinamento (*fine-tuning*) do BERTimbau, por exemplo, pode-se investigar a utilização do outro dataset utilizado na construção do pipeline em português do spaCy.

[1] Accuracy Evaluation: https://spacy.io/models/pt#pt_core_news_lg-accuracy
[2] UD Portuguese Bosque v2.8: https://github.com/explosion/spacy-models/releases/tag/pt_core_news_sm-3.3.0

Repositório do Estudo

Os notebooks foram executados na ferramenta Google Colaboratory[1] e se encontram no repositório Git: https://github.com/danieldasilvacosta/uff--trabalho-2--2022.

[1] Google Colaboratory: https://colab.research.google.com/