Towards Portparser – a highly accurate parsing system for Brazilian Portuguese following the Universal Dependencies framework

Lucelene Lopes and **Thiago Alexandre Salgueiro Pardo**
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
lucelene@gmail.com taspardo@icmc.usp.br

Abstract

This paper presents a parsing model – whose corresponding system is named Portparser – for Brazilian Portuguese, which outperforms current systems for news texts in this language. Following the Universal Dependencies (UD) framework, we build our model by using a recently released manually annotated corpus (Porttinari-base) for training. We test different parsing methods and explore parameter settings in order to propose a highly accurate model, encompassing not only the dependency annotation, but also the Part of Speech tagging and the identification of lemmas and the related morphological features. Our experiments show that our best model achieves around 99% accuracy for Part of Speech tagging, lemma, and morphological features, with around 95% for dependency annotation, surpassing known systems for Portuguese by up to 7% accuracy. Furthermore, we conduct an error analysis of the proposed model to show the current limitations and challenges for future works.

1 Introduction

Parsers are useful for several Natural Language Processing (NLP) tasks, as machine translation, text simplification, and information extraction, among many others, whether such tasks take their language processing decisions directly over explicit syntactical representations, or use them as complementary information to improve statistical and neural model results.

Building highly accurate parsing systems is a classical challenge for NLP. In particular, for Portuguese, there has been several initiatives, for both constituency and dependency-based analysis styles, but with limited performances. As an example, the widely known parser UDPipe 2 (Straka, 2018), trained on the datasets of the international Universal Dependencies (UD) framework (de Marneffe et al., 2021), achieves 87.04% for news texts[1] according to the well-known Labeled Attachment Score (LAS)[2]. When we consider that the techniques underlying the NLP applications have their own limitations, using such parsed data as input information will cause cumulative errors, which may significantly hinder the system performance.

Advancing parsing results is costfull, as it requires dealing with difficult linguistic decisions, and many linguistic phenomena are not fully formalized in Linguistics for NLP purposes (see, *e.g.*, Duran et al. (2021a,b, 2022)). Producing bigger annotated datasets (treebanks) for training parsing systems requires annotation that may be a long and hard process. It is also necessary to create appropriate computational models for the task, which may be computationally expensive, specially considering current deep learning strategies. Despite such difficulties, facing this challenge is a necessary and relevant endeavor in NLP.

This paper addresses this challenge. We present an in-depth investigation on dependency parsing for the Brazilian Portuguese language in order to produce Portparser (which stands for "PORTuguese PARSER"). Following the UD framework, we use a manually annotated corpus – the Porttinari-base (Duran et al., 2023a) – for training. We test different parsing methods and explore parameter settings in order to propose a highly accurate model, encompassing not only the dependency annotation, but also the Part of Speech (PoS) tagging, the identification of lemmas, and the morphological features.

To illustrate the annotation of morphosyntactic information in Portuguese using UD standards, Figure 1 presents the annotation of the sentence "*Esse*

[1] It is worthy to note that more recent trained models – using UD version 2.12 – achieve near 90% LAS for news texts in Portuguese, as reported at https://ufal.mff.cuni.cz/udpipe/2/models (accessed on January 2024).

[2] As defined by Nivre and Fang (2017), the Labeled Attachment Score "evaluates the output of a parser by considering how many words have been assigned both the correct syntactic head and the correct label" of the relation, being the main evaluation metric in the area.

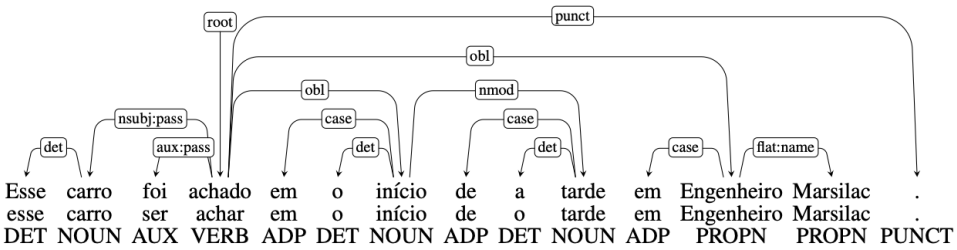

Figure 1: Example of UD morphosyntactic annotation (PoS tag, lemma, and dependency relations) – reproduced from (Rademaker et al., 2017).

carro foi achado no início da tarde em Engenheiro Marsilac.", where the PoS tags, lemmas and dependency relations are shown. Note that the morphological features are not included in this figure.

Our experiments show that, for news texts, our best model achieves around 99% accuracy for PoS tagging, lemma, and morphological features, with around 95% for dependency annotation, surpassing known systems for Portuguese by up to 7% accuracy. More than this, we conduct an error analysis of the proposed model to show the current limitations and challenges for future works.

This paper is organized as follows: next section briefly introduces the main related work; the third section presents the experiments conducted towards the choice of the proposed model, as well as a comparison with three baseline models for Portuguese; the fourth section presents an error analysis of the proposed model; finally, some final remarks are made.

2 Related Work

There are many initiatives for building models to develop accurate text analysis systems. Some of these initiatives focus on multilingual approaches, as those promoted by the CoNLL shared tasks (Zeman et al., 2018), while others, as the work of Abudouwaili et al. (2023), try to combine models from different languages to produce reasonably accurate models. Although these initiatives are quite useful to low-resource languages, it is acknowledged that the best results are often obtained with approaches focused on a specific language (Vianna et al., 2023).

This is the case of the work of Nehrdich and Hellwig (2022) that aims at the development of an accurate model to parse Latin texts, which is, nonetheless, a language with abundant written resources. In this work the authors integrate three Latin corpora and try several techniques to maximize the accuracy for PoS tagging and dependency parsing. Among the techniques employed, the authors explore the use of Latin specific character and word embeddings, as well as the use of two parsing methods (Biaffine and UDPipe 2). According to the authors, the obtained dependency relation (DEPREL) accuracy is around 93%, which is a significant value for Latin.

Another example of specific language effort is the work of Arnardóttir et al. (2023) that generates a working model for dependency parsing starting from a constituency-based annotated corpus in Icelandic. In this work, the authors develop a conversion pipeline and evaluate the accuracy of dependency relation detection, achieving values around 73% and 81% for dependency relation labels and structure, respectively.

A similar initiative to our work is the work of Silva et al. (2023), which propose a model to predict UD PoS tags for Portuguese texts. Among the techniques employed, the authors adopt language specific word embeddings, as well as UDPipe 2 parsing method. The authors' experiments reach an impressive accuracy of over 99% for PoS tags, but no morphological features, lemmas, or dependency relations are predicted in this work.

It is also interesting to mention the work of Mæhlum et al. (2022), which focuses on the proposition of a model to annotate only PoS tags to Norwegian language varieties employed in Twitter texts. This, although similar in method to our paper, contributes more as an illustration of the need for specific models to obtain an accurate annotation, since it shows that traditional models for Norwegian have very low accuracy, while their model, specific for the target genre and language, achieved nearly 86% accuracy for PoS tag annotation. This work also relate to ours by experimenting their model with different methods, including UDPipe 2 (Straka, 2018) and Stanza (Qi et al., 2020), and by performing a quantitative error analysis.

For comparative purposes, it is important to cite some of the current dependency parsers that are available for Portuguese, specially those using UD relations, which is the adopted framework in this paper. As commented before, UDPipe 2 is probably the most used one. Based on a graph-based biaffine attention architecture, Straka (2018) reports a LAS of 87.04% for news texts (although more recent models have achieved near 90% of LAS, as commented before). Stanza is another well-known system that includes Portuguese. It uses a feature-enriched Bi-LSTM-based deep biaffine neural method. The authors report accuracy metrics for some languages only, not citing the case of Portuguese in the reference paper (Qi et al., 2020)[3], but, for the cited languages, Stanza achieves LAS values above the ones achieved by UDPipe 2. UDify (Kondratyuk and Straka, 2019) is another relevant system (a semi-supervised multitask self-attention model), but the authors report comparative results for news texts that are worse than those produced by UDPipe 2. Finally, although it produces lower results than the ones obtained more recently, it is worthy to cite the work of Zilio et al. (2018), that compares several previous and more classical Portuguese parsing methods, including the well-known PALAVRAS parser (Bick, 2000). The authors report that the best model achieved LAS of 85.21%, slightly outperforming PALAVRAS in an additional small scale evaluation.

3 The Choice of the Model

In order to chose our proposed model, we conducted a series of experiments over a corpus in Brazilian Portuguese of manually annotated newspaper texts (Duran et al., 2023a). This corpus, named Porttinari-base, is composed by 8,418 sentences (168,080 tokens) manually annotated using UD standards for the morphosyntactic and syntactic levels (PoS, morphological features, lemma, and dependency relation information).

To each experiment, we split the 8,418 sentences in training (train), development (dev), and test (test) sets, respectively with 70% (5,893 sentences), 10% (842 sentences), and 20% (1,683 sentences) of the corpus. In order to give more statistical significance to our experiments, we replicated all experiments using 10 random distributions of sentences into the three sets (generating ten models numbered from 0 to 9). Each one of the distributions has a different choice of sentences, thus leading to slightly distinct number of tokens in each as shown in Table 1.

Sets with all 8,418 sentences						
	train		**dev**		**test**	
model	sents.	tokens	sents.	tokens	sents.	tokens
0	5,893	117,025	842	16,811	1,683	34,244
1	5,893	117,789	842	16,941	1,683	33,350
2	5,893	118,387	842	16,439	1,683	33,254
3	5,893	117,952	842	16,726	1,683	33,402
4	5,893	117,805	842	16,749	1,683	33,526
5	5,893	118,453	842	16,664	1,683	32,963
6	5,893	117,482	842	16,663	1,683	33,935
7	5,893	118,226	842	16,665	1,683	33,189
8	5,893	117,797	842	16,686	1,683	33,597
9	5,893	117,301	842	16,727	1,683	34,052

Table 1: Size of sets for 8,418 sentences for the 10 experimented models.

3.1 Choosing the Parsing Method

The initial experiments aimed to choose the parsing method among those that are widely used in the area, namely, UDPipe 1.3 (Straka and Straková, 2017), Stanza (Qi et al., 2020), and UDPipe 2 (Straka, 2018). In order to reproduce a behavior of most users, we applied our train, dev, and test sets (10 models) with gold tokenization to the default versions of the three methods. Table 2 shows the accuracy of each model, as well as overall average and standard deviation for the annotation. Specifically, we compute the accuracy for the fields PoS (UPOS), morphological features (UFeats) and lemma (Lemmas), and the usual measures Unlabeled Attachment Score[4] (UAS) and LAS to characterize the dependency relations.

Observing the results in Table 2, we notice a better performance of UDPipe 2, which is superior to both UDPipe 1.3 and the Stanza application. To all 5 measures (UPOS, UFEATS, LEmmas, UAS, and LAS), we performed an ANOVA test that indicates the statistical significance of the difference among methods (p-value < 0.0001). Consequently, we will adopt UDPipe 2 in the subsequent experiments, trying to improve even more the accuracy results given our training corpus.

3.2 Choosing the Number of Epochs

The second batch of experiments consisted in applying the default parameters and variate the number of epochs from the default 40-20 to 20-20, 60-20, and 80-20 (always with learning rates of 10^{-3}

[3]However, results for UD version 2.12 are available online at https://stanfordnlp.github.io/stanza/performance.html, achieving 87.75% of LAS for news texts in Portuguese (accessed on January 2024).

[4]Differently from LAS, UAS indicates the accuracy of the HEAD field ignoring the relation name field (DEPREL). It is worthy mentioning that LAS considers only the DEPREL relation, ignoring subrelations.

UDPipe 1.3					
model	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**
0	97.81%	97.59%	97.83%	89.32%	86.59%
1	97.59%	97.46%	97.79%	89.16%	86.31%
2	97.64%	97.64%	97.72%	89.68%	86.92%
3	97.50%	97.32%	97.68%	89.51%	86.71%
4	97.67%	97.60%	97.91%	89.84%	87.20%
5	97.65%	97.51%	97.82%	89.43%	86.61%
6	97.65%	97.51%	97.88%	89.70%	87.03%
7	97.66%	97.55%	97.81%	89.31%	86.62%
8	97.56%	97.41%	97.73%	89.46%	86.69%
9	97.54%	97.42%	97.86%	89.15%	86.51%
average	97.63%	97.50%	97.80%	89.46%	86.72%
st. dev.	0.0820	0.0943	0.0701	0.2195	0.2487
Stanza					
0	96.18%	95.89%	98.75%	89.12%	87.48%
1	96.55%	94.83%	98.36%	89.43%	86.90%
2	96.34%	95.26%	99.01%	89.08%	86.13%
3	96.72%	94.90%	98.84%	88.59%	86.62%
4	95.98%	94.92%	98.47%	89.60%	87.22%
5	95.79%	95.01%	98.83%	88.93%	87.10%
6	96.17%	95.68%	98.57%	88.90%	86.37%
7	95.32%	95.54%	98.73%	89.04%	86.25%
8	96.06%	95.13%	98.23%	88.38%	86.78%
9	96.48%	94.77%	98.07%	88.49%	87.02%
average	96.16%	95.19%	98.59%	88.96%	86.79%
st. dev.	0.3855	0.3683	0.2845	0.3709	0.4189
UDPipe 2					
0	98.50%	98.36%	99.02%	93.59%	91.65%
1	98.33%	98.23%	98.93%	93.31%	91.34%
2	98.41%	98.17%	99.03%	93.77%	91.88%
3	98.37%	98.07%	99.01%	93.87%	91.76%
4	98.51%	98.28%	99.11%	93.76%	91.92%
5	98.31%	98.25%	99.03%	93.53%	91.45%
6	98.41%	98.27%	99.05%	93.67%	91.75%
7	98.40%	98.21%	99.04%	93.57%	91.63%
8	98.28%	98.10%	98.90%	93.44%	91.38%
9	98.29%	98.14%	98.94%	93.62%	91.63%
average	**98.38%**	**98.21%**	**99.01%**	**93.61%**	**91.64%**
st. dev.	**0.0769**	**0.0844**	**0.0605**	**0.1570**	**0.1888**

Table 2: Accuracy of the parsing methods for the 10 models of 8,418 sentences.

model	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**
40-20 epochs					
0	98.50%	98.36%	99.02%	93.59%	91.65%
1	98.33%	98.23%	98.93%	93.31%	91.34%
2	98.41%	98.17%	99.03%	93.77%	91.88%
3	98.37%	98.07%	99.01%	93.87%	91.76%
4	98.51%	98.28%	99.11%	93.76%	91.92%
5	98.31%	98.25%	99.03%	93.53%	91.45%
6	98.41%	98.27%	99.05%	93.67%	91.75%
7	98.40%	98.21%	99.04%	93.57%	91.63%
8	98.28%	98.10%	98.90%	93.44%	91.38%
9	98.29%	98.14%	98.94%	93.62%	91.63%
average	98.38%	98.21%	99.01%	93.61%	91.64%
st. dev.	**0.0769**	0.0844	0.0605	**0.1570**	0.1888
60-20 epochs					
0	**98.59%**	**98.41%**	99.02%	93.63%	91.72%
1	98.41%	98.26%	98.96%	93.38%	91.49%
2	98.46%	98.27%	99.06%	93.81%	91.95%
3	98.36%	98.11%	98.98%	**94.06%**	91.97%
4	98.54%	98.32%	99.12%	93.80%	91.95%
5	98.34%	98.27%	99.10%	93.51%	91.49%
6	98.39%	98.28%	99.04%	93.76%	91.86%
7	98.42%	98.26%	99.10%	93.71%	91.84%
8	98.29%	98.15%	98.95%	93.63%	91.51%
9	98.31%	98.19%	99.02%	93.70%	91.73%
average	98.41%	98.25%	99.03%	**93.70%**	**91.75%**
st. dev.	0.0917	0.0810	0.0571	0.1743	0.1851
80-20 epochs					
0	98.55%	98.40%	99.06%	93.57%	91.67%
1	98.35%	98.26%	98.96%	93.39%	91.46%
2	98.42%	98.28%	99.06%	93.78%	91.94%
3	98.38%	98.13%	99.02%	94.04%	**91.98%**
4	98.56%	98.36%	**99.14%**	93.77%	91.96%
5	98.35%	98.30%	99.11%	93.44%	91.43%
6	98.46%	98.34%	99.07%	93.68%	91.83%
7	98.42%	98.27%	99.11%	93.69%	91.80%
8	98.34%	98.16%	98.99%	93.70%	91.67%
9	98.36%	98.24%	99.02%	93.62%	91.65%
average	**98.42%**	**98.27%**	**99.05%**	93.67%	91.74%
st. dev.	0.0771	**0.0796**	**0.0541**	0.1744	0.1871

Table 3: Accuracy variation according to number of epochs for the 10 models of 8,418 sentences.

for the initial epochs and 10^{-4} for the final ones). The other hyper-parameters employed are: batch size 32; character-level embedding dimension 256; maximum sentence length 120; RNN cell type and dimension LSTM 512; word embedding dimension 512; and bert-base multilingual uncased as word embedding model, as mentioned as default by UDPipe 2 initial publication (Straka, 2018).

Table 3 presents the outcome of UDPipe 2 training for the ten models tested, as well as their average and standard deviation.

The results show little variation for the ten experimented models. It is noticeable that, while the number of epochs increases, there is some improvement in terms of the accuracy. In Table 3, the highest values of accuracy and the smallest standard deviation values are marked in bold.

For the numbers in Table 3, we applied the ANOVA test for each measure and could not establish a clear superiority of any result over the other (p-values equal to 0.61204, 0.199917, 0.24114, 0.55917, and 0.39045, respectively for UPOS, UFeats, Lemmas, UAS e LAS). Even the smallest number of epochs experimented (40-20) delivers accuracy values with less than 1% of difference from the best results (60-20 and 80-20 epochs).

In fact, these results indicate that, in a general approach, it is probably not worthy, due to training time, to consider a large number of epochs. It is worthy mentioning that the execution of the training of our 10 models with 80-20 epochs took more than 200 hours of processing (20 hours per model) in a Google Colab Pro+ with 51 Gb System RAM, 225 Gb Disk, TPU accelerator. However, given our specific goal to search for the best possible model, we will consider Model 3 with 80-20 epochs as the best one, as it has the highest value of LAS, since dependency relation (HEAD and DEPREL) is the hardest information to be accurately annotated.

3.3 Considerations on the Model Size

The third set of experiments explores the effect of the train and dev sets' size. In order to do so, we chose reduced sets of 6,314, 4,209, and 2,104 sentences randomly picked from the original 8,418 sentence pool. To each of those reduced sets, we also generated 10 models with randomly picked sentences, being the train set with 70% of the sen-

tences, the dev set with 10% of the sentences, and the test set with 20% of the sentences. Table 4 shows the number of sentences and tokens for each model of each of the sets.

Sets with only 6,314 sentences						
	train		**dev**		**test**	
model	sents.	tokens	sents.	tokens	sents.	tokens
0	4,420	86,997	631	12,661	1,263	25,229
1	4,420	87,324	631	12,520	1,263	25,043
2	4,420	87,125	631	12,961	1,263	24,801
3	4,420	88,054	631	12,455	1,263	24,378
4	4,420	87,212	631	12,549	1,263	25,126
5	4,420	87,533	631	12,344	1,263	25,010
6	4,420	87,095	631	12,645	1,263	25,147
7	4,420	87,667	631	12,236	1,263	24,984
8	4,420	87,465	631	12,342	1,263	25,080
9	4,420	86,911	631	12,516	1,263	25,460
Sets with only 4,209 sentences						
	train		**dev**		**test**	
model	sents.	tokens	sents.	tokens	sents.	tokens
0	2,946	50,415	421	7,120	842	14,505
1	2,946	50,307	421	7,261	842	14,472
2	2,946	50,290	421	7,348	842	14,402
3	2,946	50,257	421	7,244	842	14,539
4	2,946	50,465	421	7,155	842	14,420
5	2,946	50,751	421	6,959	842	14,330
6	2,946	50,518	421	7,239	842	14,283
7	2,946	50,402	421	7,322	842	14,316
8	2,946	50,343	421	7,092	842	14,605
9	2,946	50,422	421	7,082	842	14,536
Sets with only 2,104 sentences						
	train		**dev**		**test**	
model	sents.	tokens	sents.	tokens	sents.	tokens
0	1,473	24,494	210	3,656	421	6,981
1	1,473	24,873	210	3,477	421	6,781
2	1,473	24,399	210	3,497	421	7,235
3	1,473	24,405	210	3,723	421	7,003
4	1,473	24,721	210	3,358	421	7,052
5	1,473	24,822	210	3,423	421	6,886
6	1,473	24,878	210	3,523	421	6,730
7	1,473	24,791	210	3,422	421	6,918
8	1,473	24,506	210	3,518	421	7,107
9	1,473	24,597	210	3,519	421	7,015

Table 4: Size of each model for the reduced sets.

Performing the analysis of the cases described in Table 4 with 80-20 epochs, we obtain the accuracy values presented in Table 5, that also presents the average and standard deviation per model size. This table shows each group of models indicating the size of sets employed to train (number of sentences for the train and dev sets).

It is noticeable, by the obtained results, that the train and dev sets' size has a clear impact on the accuracy of the generated model. To all 5 measures we performed an ANOVA test that indicates the statistical significance of the difference among methods (p-value < 0.0001). In fact, the results of a model created from larger sets has always a better accuracy than a model generated from smaller ones. For example, the results for the models created from 2,946+421 sentences are always inferior to the results for models created from 4,420+631 sentences, and always superior to those for models created from 1,473+210 sentences.

model	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**
train 5,893 sentences - dev 842 sentences					
0	98.55%	**98.40%**	99.06%	93.57%	91.67%
1	98.35%	98.26%	98.96%	93.39%	91.46%
2	98.42%	98.28%	99.06%	93.78%	91.94%
3	98.38%	98.13%	99.02%	**94.04%**	**91.98%**
4	**98.56%**	98.36%	**99.14%**	93.77%	91.96%
5	98.35%	98.30%	99.11%	93.44%	91.43%
6	98.46%	98.34%	99.07%	93.68%	91.83%
7	98.42%	98.27%	99.11%	93.69%	91.80%
8	98.34%	98.16%	98.99%	93.70%	91.67%
9	98.36%	98.24%	99.02%	93.62%	91.65%
average	**98.42%**	**98.27%**	**99.05%**	**93.67%**	**91.74%**
st. dev.	**0.0771**	**0.0796**	**0.0541**	0.1744	0.1871
train 4,420 sentences - dev 631 sentences					
0	98.14%	98.05%	98.89%	92.82%	90.70%
1	98.02%	97.83%	98.77%	92.96%	90.85%
2	98.00%	97.86%	98.80%	92.93%	90.50%
3	97.97%	97.76%	98.82%	93.26%	91.02%
4	98.08%	97.78%	98.77%	92.98%	90.87%
5	98.19%	98.04%	98.73%	93.13%	90.75%
6	98.21%	98.15%	98.87%	92.72%	90.60%
7	98.01%	97.97%	98.82%	93.00%	90.67%
8	98.11%	97.90%	98.87%	92.93%	90.87%
9	98.26%	98.09%	98.90%	93.09%	91.12%
average	98.10%	97.94%	98.82%	92.98%	90.80%
st. dev.	0.0945	0.1295	0.0544	**0.1456**	**0.1795**
train 2,946 sentences - dev 421 sentences					
0	97.84%	97.42%	98.61%	92.02%	89.73%
1	97.64%	97.28%	98.22%	92.15%	89.57%
2	97.74%	97.54%	98.67%	92.30%	90.20%
3	97.87%	97.41%	98.68%	92.35%	89.76%
4	97.55%	97.32%	98.40%	92.06%	89.36%
5	97.75%	97.48%	98.51%	91.56%	89.19%
6	97.40%	97.21%	98.45%	92.35%	89.73%
7	97.61%	97.47%	98.37%	92.71%	90.12%
8	97.41%	97.22%	98.22%	92.17%	89.60%
9	97.65%	97.35%	98.58%	91.90%	89.41%
average	97.65%	97.37%	98.47%	92.16%	89.67%
st. dev.	0.1530	0.1069	0.1605	0.2918	0.3013
train 1,473 sentences - dev 210 sentences					
0	97.32%	96.73%	98.02%	91.03%	88.65%
1	97.17%	96.84%	97.92%	92.38%	89.26%
2	97.35%	96.78%	98.11%	91.53%	88.80%
3	97.12%	96.74%	98.00%	91.90%	88.78%
4	96.81%	96.84%	97.87%	91.15%	88.57%
5	97.23%	96.66%	97.98%	91.58%	88.85%
6	97.36%	96.73%	97.93%	92.10%	89.72%
7	97.35%	96.55%	97.96%	91.69%	88.65%
8	97.30%	97.12%	98.00%	91.50%	88.84%
9	96.68%	96.45%	97.55%	90.75%	87.68%
average	97.17%	96.74%	97.93%	91.56%	88.78%
st. dev.	0.2272	0.1711	0.1420	0.4697	0.4910

Table 5: Accuracy variation according to number of sentences.

3.4 Changing the Word Embeddings (WE)

The last set of experiments changes the choice of word embeddings (WE) from the bert-based-multilingual-uncased used by default in UDPipe 2 to the bert-large-portuguese-cased, also know as BERTimbau (Souza et al., 2020). This choice aims to pass from the multilingual encoding to a encoding designed for Brazilian Portuguese, thus, more likely to improve the accuracy of the proposed model (Vianna et al., 2023).

The process to change WE in UDPipe 2 requires some additional processing to previously compute the embedding of each token of the train, dev, and test sets according to the chosen WE model. This process creates .npz files that must accompany the .conllu files of the annotated sets. Analogously, to

use the models to annotate, it is required to generate the WE for the text to annotate (the .npz file).

To perform the last set of experiments, we used the UDPipe 2 default hyperparameters, except for the usage of Brazilian Portuguese WE. Therefore, in Table 6, we are comparing the model results obtained by the best multilingual (80-20 epochs) with the usage of BERTimbau and 40-20 epochs (UDPipe 2 default)[5].

model	UPOS	UFeats	Lemmas	UAS	LAS
BERT multilingual WE					
0	98.55%	98.40%	99.06%	93.57%	91.67%
1	98.35%	98.26%	98.96%	93.39%	91.46%
2	98.42%	98.28%	99.06%	93.78%	91.94%
3	98.38%	98.13%	99.02%	94.04%	91.98%
4	98.56%	98.36%	99.14%	93.77%	91.96%
5	98.35%	98.30%	99.11%	93.44%	91.43%
6	98.46%	98.34%	99.07%	93.68%	91.83%
7	98.42%	98.27%	99.11%	93.69%	91.80%
8	98.34%	98.16%	98.99%	93.70%	91.67%
9	98.36%	98.24%	99.02%	93.62%	91.65%
average	98.42%	98.27%	99.05%	93.67%	91.74%
st. dev.	0.0771	0.0796	0.0541	0.1744	0.1871
BERTimbau Brazilian Portuguese WE					
0	99.17%	98.92%	99.34%	95.70%	94.32%
1	99.01%	98.83%	99.30%	95.43%	94.04%
2	99.12%	**98.92%**	99.37%	95.73%	94.45%
3	99.10%	98.74%	99.40%	**96.08%**	**94.70%**
4	**99.19%**	98.82%	**99.42%**	95.81%	94.60%
5	99.11%	98.87%	99.42%	95.89%	94.51%
6	99.04%	98.86%	99.33%	95.77%	94.39%
7	99.01%	98.73%	99.32%	95.63%	94.24%
8	99.06%	98.74%	99.28%	95.89%	94.50%
9	99.08%	98.83%	99.35%	95.62%	94.30%
average	**99.09%**	**98.83%**	**99.35%**	**95.76%**	**94.41%**
st. dev.	**0.0584**	**0.0670**	**0.0463**	**0.1698**	**0.1806**

Table 6: Accuracy variation according to WE.

Observing the results of Table 6, we see a clear accuracy improvement with the BERTimbau WE. To all 5 measures we performed an ANOVA test that indicates the statistical significance of the difference among methods (p-value < 0.0001).

While the UPOS, UFeats, and Lemmas are annotated with a nearly perfect accuracy (99%), the dependency relation measurements UAS and LAS increased between 2% and 3%, depending on the model. Focusing on the obtained accuracy values of each model, it is possible to observe Model 3 with the best results for LAS (with impressive 94.70%). This model will therefore be adopted as our proposed learned model, that shall compose the first version of Portparser.

[5]For this experiment, we employed the default number of epochs (40-20) instead of the larger experimented 60-20 and 80-20 settings, since the accuracy results with a larger number of epochs were not really affected, showing that the training process has converged already for the 40-20 epochs case.

3.5 Comparison of the Proposed Model with Baselines

To illustrate the benefits brought by the proposed model, we draw a comparison with three baseline models currently available for Portuguese, which correspond to UDPipe 2 method trained on the following UD datasets[6] version 2.12:

- CINTIL-UDep (Branco et al., 2022) is a dependency bank that is composed mostly by newspaper texts;
- Bosque-UD (Rademaker et al., 2017) is a treebank based on the Constraint Grammar converted version of the Bosque corpus;
- PetroGold (Souza et al., 2021) is a fully revised treebank that consists of academic texts from the oil and gas domain.

We employed the three baselines to annotate the same test data of our proposed model. Table 7 shows comparatively the accuracy of the baselines, as well as the accuracy of our proposed model presented at Section 3.4.

model	UPOS	UFeats	Lemmas	UAS	LAS
CINTIL	95.11%	90.33%	82.54%	84.37%	68.21%
Bosque	96.21%	82.53%	97.91%	91.34%	86.87%
PetroGold	97.40%	83.41%	98.21%	90.93%	87.48%
Our Model	**99.10%**	**98.74%**	**99.40%**	**96.08%**	**94.70%**

Table 7: Comparison of our proposed model accuracy to the accuracy of three baselines.

Using different training datasets has certainly an impact on the results and on the conclusions that one may draw, but helps to put things in (relative) perspective. Having this warning been made, it is clear the superiority of our proposed model for all accuracy values. It is noticeable the improvements in terms of PoS tags and lemmas that were already well annotated by the baselines. For morphological features, we notice a very significant improvement, bringing the accuracy to the same level of PoS and lemma. Another impressive result is in terms of an improvement of UAS, which reflects a better annotation of the dependency structure. The UAS accuracy became nearly 5% better than the best baseline. The more relevant achievement, thought, is the accuracy of 94.70% in LAS, that raises more than the 7% in comparison with the best baseline.

[6]https://universaldependencies.org

4 Proposed Model Error Analysis

We also performed an analysis of our proposed model observing the wrong predictions for UPOS and DEPREL tags (affecting LAS). The subject of this analysis was the test dataset that is composed of 1,683 sentences.

4.1 PoS tag errors

Table 8 presents the number of tokens wrongfully predicted for each PoS tag, indicating the percentage of error (% error) and absolute number of errors (# tokens), plus the total number of tokens that should have been annotated with the corresponding PoS tag (# total tokens). It is important to recall that the test dataset has 33,402 tokens, and our proposed model committed errors for 300 of those tokens, i.e., an accuracy of 99.1%.

UPOS	% error	# tokens	# total tokens
X	60%	37	62
INTJ	50%	3	6
ADJ	3%	54	1,756
SCONJ	2%	10	464
PRON	2%	23	1,281
NUM	2%	12	676
ADV	2%	21	1,319
CCONJ	1%	11	819
VERB	1%	39	3,422
SYM	1%	1	120
NOUN	1%	43	6,254
PROPN	1%	14	2,041
AUX	1%	5	949
DET	≈0%	18	4,761
ADP	≈0%	8	4,924
PUNCT	≈0%	1	4,548

Table 8: Error for each PoS tag using our proposed model.

Observing the confusion matrix of PoS tags (Table 9), we noticed three clusters:

- a large cluster involving most mistakes (54 *ADJ*, 43 *NOUN*, 39 *VERB*) with tokens that should be *ADJ* and were annotated as *NOUN* (25 tokens) and *VERB* (25 tokens), tokens that should be *NOUN* and were annotated as *ADJ* (20 tokens) and *VERB* (4 tokens), and tokens that should be *VERB* and were annotated as *ADJ* (21 tokens) and *NOUN* (7 tokens);
- a cluster with errors between *DET* and *PRON*, where 15 tokens that should be *PRON* were annotated as *DET*, and 6 tokens that should be *DET* were annotated as *PRON*;
- a cluster with errors between *VERB* and *AUX*, where 11 tokens that should be *VERB* were annotated as *AUX*, and 5 tokens that should be *AUX* were annotated as *VERB*.

It was also noticed a difficulty to predict tokens that should be *X*, which were frequently annotated as *NOUN* (26 tokens), plus another 11 errors being annotated as *ADJ* (5 tokens), *ADP* (3 tokens), *PROPN* (2 tokens), and even *ADV* (1 token). Similarly, we also noticed a difficulty of the method to recognize 17 tokens that should be *NOUN* but were annotated as *PROPN*.

4.2 DEPREL tags errors

Performing the same analysis for the errors in the DEPREL field (which has an direct impact on the LAS accuracy), we have the results presented in Table 10. A total of 1,028 errors of DEPREL tag were found for the 33,204 tokens, which represents an accuracy of 96.9%. Note that LAS accuracy is slightly lower (94.7%), since LAS indicates HEAD and DEPREL fields correctly predicted.

Table 10 shows that some DEPREL tags were frequently predicted wrongfully due to under representation in the training set, as *dislocated*, *vocative*, *orphan*, and *iobj* relations. However, other DEPREL tags, as *obl*, *nmod*, *nsubj*, and *obj*, had a large number of errors despite an abundance of occurrences.

Pushing the analysis, we have focused on the 16 DEPREL tags with the highest absolute number of prediction errors. These tags are responsible for 885 out of the 1,028 errors in total for this test. Table 11 presents these numbers, indicating the prediction errors (confusion matrix). In this table, the last row indicates the number of annotation errors of a token with a tag not belonging to the chosen 16 DEPREL tags we focused on.

Observing the errors in the DEPREL tags from Table 11, it is possible to observe some common mistakes between pairs of DEPREL tags. For example, the more common mistakes were between the tags *obl* and *nmod*[7], since 114 tokens that should be annotated as *obl* were predicted as *nmod*. Analogously, 75 tokens that should be annotated as *nmod* were predicted as *obl*. The pair *case* and *mark* also shows a relevant confusion, with 19 tokens that should be annotated as *mark* being predicted as *case*, and 6 tokens that should be annotated as *case* being predicted as *mark*.

[7]Some of these mistakes had already been noticed by other researchers when analyzing UDPipe errors (Duran et al., 2023b).

annotated as	should be													
	ADJ	*ADP*	*ADV*	*AUX*	*DET*	*CCONJ*	*INTJ*	*NOUN*	*NUM*	*PRON*	*PROPN*	*SCONJ*	*VERB*	*X*
ADJ	-	0	3	0	4	0	0	20	0	1	2	0	21	5
ADP	0	-	3	0	2	0	0	0	0	0	0	2	0	3
ADV	2	0	-	0	2	6	1	1	0	1	0	2	0	1
AUX	0	0	0	-	0	0	0	0	0	0	1	0	11	0
DET	1	3	2	0	-	0	0	0	10	15	0	0	0	0
CCONJ	0	0	4	0	0	-	0	0	0	0	0	1	0	0
INTJ	0	0	0	0	0	0	-	0	0	0	0	0	0	0
NOUN	25	1	1	0	0	0	1	-	1	2	8	0	7	26
NUM	0	0	0	0	2	0	0	0	-	2	1	0	0	0
PRON	0	1	1	0	6	0	0	0	1	-	1	5	0	0
PROPN	1	0	0	0	0	0	1	17	0	0	-	0	0	2
SCONJ	0	1	7	0	2	5	0	0	0	2	0	-	0	0
VERB	25	2	0	5	0	0	0	4	0	0	1	0	-	0
X	0	0	0	0	0	0	0	1	0	0	0	0	0	-

Table 9: Confusion matrix between the 14 UPOS tags with higher number of errors (not *SYM* and *PUNCT*).

DEPREL	% error	# tokens	# total tokens
dislocated	100%	6	6
vocative	67%	2	3
orphan	56%	9	16
iobj	37%	7	19
discourse	31%	11	35
parataxis	26%	46	174
csubj	15%	11	74
acl	11%	81	719
advcl	11%	51	474
xcomp	9%	41	456
obl	8%	162	1,910
fixed	7%	18	250
ccomp	7%	26	379
conj	7%	58	877
appos	5%	12	219
nmod	5%	118	2,511
nummod	4%	16	369
obj	4%	53	1,433
aux	4%	13	361
flat	4%	24	678
nsubj	3%	70	2,066
mark	3%	28	850
amod	3%	39	1,332
advmod	3%	32	1,265
root	2%	35	1,683
cc	2%	15	837
expl	1%	1	145
case	≈0%	21	4,432
det	≈0%	19	4,710
cop	≈0%	2	571
punct	≈0%	1	4,548

Table 10: Error for each DEPREL tag using our proposed model.

5 Final remarks

This paper focused on producing a model capable of accurately annotating morphosyntactic and syntactic information in Portuguese news texts according to UD standards. We adopted Porttinari-base as dataset and explored different parsing methods and parameters for training. Our best model achieved PoS tag, morphological features and lemma annotation accuracy of around 99%, and dependency relation accuracy around impressive 96% (UAS) and 95% (LAS) values. Notably, our proposed model brings an improvement of LAS around 7% over some well-known existing baselines. We also presented a quantitative analysis of the errors of our proposed model for UPOS and DEPREL tags, which offer insights for future improvements. Future experiments may be based on some of these findings by indicating candidates for data augmentation initiatives (Pellicer et al., 2023), as the case of under-represented PoS and DEPREL tags.

Future works also include testing new parsing methods and performing qualitative analysis of the errors. Another interesting endeavor consists in extending our experiments to other Portuguese corpora with other text genres and domains. For example, PetroGold (Souza et al., 2021) may be an interesting corpus to tackle, as its parsing model reaches good LAS accuracy when tested on in-domain data (94.42% reported in UDPipe 2 benchmarks for UD version 2.12).

Our proposed model, as well as all datasets and full instructions to reproduce the experiments conducted in this paper, are freely available at https://github.com/LuceleneL/Portparser. More details about this work may also be found at the POeTiSA project webpage at https://sites.google.com/icmc.usp.br/poetisa/.

Acknowledgements

This work was carried out at the Center for Artificial Intelligence of the University of São Paulo (C4AI - http://c4ai.inova.usp.br/), with support by the São Paulo Research Foundation (FAPESP grant #2019/07665-4) and by the IBM Corporation. The project was also supported by the Ministry of Science, Technology and Innovation, with resources of Law N. 8.248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published as Residence in TIC 13, DOU 01245.010222/2022-44.

annotated	should be															
as	*acl*	*advcl*	*advmod*	*amod*	*case*	*ccomp*	*conj*	*flat*	*mark*	*nmod*	*nsubj*	*obj*	*obl*	*parataxis*	*root*	*xcomp*
acl	-	10	0	14	0	6	1	2	0	1	1	0	0	3	2	7
advcl	32	-	0	0	1	4	5	0	1	1	2	0	0	4	0	7
advmod	0	1	-	1	7	1	6	0	2	0	0	2	4	1	0	2
amod	19	3	1	-	0	0	6	4	0	12	2	1	0	0	3	5
case	0	0	5	0	-	0	0	0	19	2	0	0	0	0	0	0
ccomp	3	3	1	0	0	-	2	0	0	0	1	0	1	2	8	3
conj	4	4	4	3	0	0	-	1	0	2	6	1	4	12	1	0
flat	0	0	0	0	0	0	3	-	0	14	0	0	0	0	1	0
mark	1	0	7	0	6	0	0	0	-	0	0	2	0	0	0	0
nmod	5	3	0	4	0	0	5	10	0	-	8	3	114	1	3	0
nsubj	4	5	0	4	0	3	1	2	3	2	-	17	6	1	2	5
obj	0	0	2	4	1	0	2	2	0	1	24	-	21	0	0	11
obl	1	8	0	6	1	0	0	0	0	75	3	3	-	1	1	0
parataxis	0	1	1	0	0	3	18	2	0	1	1	3	1	-	3	0
root	0	2	1	1	0	3	0	1	0	0	8	0	2	10	-	1
xcomp	5	11	1	2	0	4	0	0	0	0	0	3	0	0	2	-
OTHER	7	0	9	0	5	2	9	0	3	7	14	18	9	11	9	0

Table 11: Confusion matrix between the 16 DEPREL tags with higher absolute number of errors.

References

Gulinigeer Abudouwaili, Kahaerjiang Abiderexiti, Nian Yi, and Aishan Wumaier. 2023. Joint learning model for low-resource agglutinative language morphological tagging. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 27–37, Toronto, Canada. Association for Computational Linguistics.

Þórunn Arnardóttir, Hinrik Hafsteinsson, Atli Jasonarson, Anton Ingason, and Steinþór Steingrímsson. 2023. Evaluating a Universal Dependencies conversion pipeline for Icelandic. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 698–704, Tórshavn, Faroe Islands. University of Tartu Library.

Eckhard Bick. 2000. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. University of Arhus, Århus.

António Branco, João Ricardo Silva, Luís Gomes, and João António Rodrigues. 2022. Universal grammatical dependencies for Portuguese with CINTIL data, LX processing and CLARIN support. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5617–5626, Marseille, France. European Language Resources Association.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Magali Duran, Lucelene Lopes, Maria das Graças Nunes, and Thiago Pardo. 2023a. The dawn of the Porttinari multigenre treebank: Introducing its journalistic portion. In *Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 115–124, Porto Alegre, RS, Brasil. SBC.

Magali Duran, Lucelene Lopes, and Thiago Pardo. 2021a. Descrição de numerais segundo modelo universal dependencies e sua anotação no português. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 344–352, Porto Alegre, RS, Brasil. SBC.

Magali Duran, Adriana Pagano, Amanda Rassi, and Thiago Pardo. 2021b. On auxiliary verb in Universal Dependencies: untangling the issue and proposing a systematized annotation strategy. In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 10–21, Sofia, Bulgaria. Association for Computational Linguistics.

Magali S. Duran, Maria das Graças V. Nunes, and Thiago A. S. Pardo. 2023b. Construções sintáticas do português que desafiam a tarefa de parsing: uma análise qualitativa. In *Proceedings of the 2nd Edition of the Universal Dependencies Brazilian Festival*, pages 432–441, Belo Horizonte, Brazil. Association for Computational Linguistics.

Magali S. Duran, Heloisa Oliveira, and Clarissa Scandarolli. 2022. Que simples que nada: a anotação da palavra que em corpus de UD. In *Proceedings of the Universal Dependencies Brazilian Festival*, pages 1–11, Fortaleza, Brazil. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Petter Mæhlum, Andre Kåsen, Samia Touileb, and Jeremy Barnes. 2022. Annotating Norwegian language varieties on Twitter for part-of-speech. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 64–69, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Sebastian Nehrdich and Oliver Hellwig. 2022. Accurate dependency parsing and tagging of Latin. In *Proceedings of the Second Workshop on Language*

Technologies for Historical and Ancient Languages, pages 20–25, Marseille, France. European Language Resources Association.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. 2023. Data augmentation techniques in natural language processing. *Applied Soft Computing*, 132:109803.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria de Paiva. 2017. Universal Dependencies for Portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 197–206, Pisa, Italy. Linköping University Electronic Press.

Emanuel Huber da Silva, Thiago Alexandre Salgueiro Pardo, and Norton Trevisan Roman. 2023. Etiquetagem morfossintática multigênero para o português do brasil segundo o modelo "universal dependencies". In *Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana - STIL*. SBC.

Elvis Souza, Aline Silveira, Tatiana Cavalcanti, Maria Castro, and Cláudia Freitas. 2021. Petrogold – corpus padrão ouro para o domínio do petróleo. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 29–38, Porto Alegre, RS, Brasil. SBC.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. BERTimbau: Pretrained BERT models for brazilian portuguese. In *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Daniela Vianna, Fernando Carneiro, Jonnathan Carvalho, Alexandre Plastino, and Aline Paes. 2023. Sentiment analysis in portuguese tweets: an evaluation of diverse word representation models. *Language Resources and Evaluation*, pages 1–50.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Conference on Computational Natural Language Learning*.

Leonardo Zilio, Rodrigo Wilkens, and Cédrick Fairon. 2018. Passport: A dependency parsing model for portuguese. In *Computational Processing of the Portuguese Language*, pages 479–489, Cham. Springer International Publishing.