Universal Dependencies for Tweets in Brazilian Portuguese: Tokenization and Part of Speech Tagging

Emanuel Huber da Silva[1], **Thiago Alexandre Salgueiro Pardo**[1], **Norton Trevisan Roman**[2], **Ariani Di Felippo**[3]

[1]Núcleo Interinstitucional de Linguística Computacional (NILC),
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (USP)

[2]Escola de Artes, Ciências e Humanidades - Universidade de São Paulo (USP)

[3]Núcleo Interinstitucional de Linguística Computacional (NILC),
Departamento de Letras - Universidade Federal de São Carlos (UFSCar)

emanuel.huber@usp.br, taspardo@icmc.usp.br,
norton@usp.br, ariani@ufscar.br

Abstract. *Automatically dealing with Natural Language User-Generated Content (UGC) is a challenging task of utmost importance, given the amount of information available over the web. We present in this paper an effort on building tokenization and Part of Speech (PoS) tagging systems for tweets in Brazilian Portuguese, following the guidelines of the Universal Dependencies (UD) project. We propose a rule-based tokenizer and the customization of current state-of-the-art UD-based tagging strategies for Portuguese, achieving a 98% f-score for tokenization, and a 95% f-score for PoS tagging. We also introduce DANTEStocks, the corpus of stock market tweets on which we base our work, presenting preliminary evidence of the multi-genre capacity of our PoS tagger.*

1. Introduction

In usual Natural Language Processing (NLP) workflows, text preprocessing is an essential procedure. Amongst the numerous strategies, Part of Speech (PoS) tagging is one of the first processes applied to data, being responsible for assigning each word in a sentence its appropriate grammatical role. Being one of the most elementary text analysis and structuring tasks in this workflow, PoS tagging builds the basis for the development of several NLP tools and applications, such as grammar checking and text simplification.

Although having been investigated for some time in the realm of well written texts, such as news for example, where it achieves state-of-the-art results above 97% accuracy in Portuguese (*e.g.*, [Fonseca et al. 2015, de Sousa and Lopes 2019]), the situation is very different when it comes to user-generated content (UGC), such as texts written by users in social networks, which do not strictly follow the rules of standard writing. These texts are sometimes marked by orality and informality, also making use of slangs, abbreviations and media-specific content (*e.g.*, hashtags and at-mentions in Twitter), which pose considerable challenges to their automatic processing.

As a related task necessary to PoS tagging, tokenization provides the elementary units to be tagged. Even though, at first sight, it might appear to be a straightforward task, UGC makes things considerably more difficult, given the above mentioned phenomena. The consequences, however, may endure all along the NLP pipeline, since badly tokenized text will most certainly have a negative impact on the results of any PoS tagger applied to it and, consequently, on all NLP tasks that depend on this tagger’s results.

Consider, for example, the tweet presented in Figure 1, taken from DANTEStocks, along with its tokenization and PoS tagging, as produced by our system. As it can be seen, the text does not comply with the standard rules for writing (specially regarding capitalization, word splitting, punctuation, the presence of slangs and abbreviations etc.), also presenting elements that are characteristic to the platform where they were written (*e.g.* the presence of hashtags and URLs).

Original:

#VALE5 é #VENDA? rsss #DEAL! #DEAL! #DEAL! ’16 de março às 12:12’ após vencto das opções podem puxar na... http://t.co/4mOMj1Om7d

Tokenized and PoS tagged:

#VALE5/PROPN é/AUX #VENDA/NOUN ?/PUNCT rsss/X #DEAL/NOUN !/PUNCT #DEAL/NOUN !/PUNCT #DEAL/NOUN !/PUNCT ’/PUNCT 16/NUM de/ADP março/NOUN a/ADP as/DET 12:12/NUM ’/PUNCT após/ADP vencto/NOUN de/ADP as/DET opções/NOUN podem/AUX puxar/VERB em/ADP a/DET .../PUNCT http://t.co/4mOMj1Om7d/SYM

Figure 1. Example of tweet from DANTEStocks, tokenized and PoS tagged.

Recently, initiatives have arisen to build morphosyntactically and syntactically annotated corpora of UGC such as, for instance, the treebank of tweets in English created by [Liu et al. 2018]. Although this is probably the most representative work in the area, many others have emerged, motivating authors like [Sanguinetti et al. 2020] to propose unified strategies to annotate UGC.

Guiding the recent work in morphosyntax and syntax in the area (including the ones cited above) is the Universal Dependencies (UD) initiative[1] [Nivre et al. 2016, Nivre et al. 2020]. UD aims at establishing *universal* tags and syntactical relations for corpus annotation, allowing cross-lingual studies and the reuse of methodologies. Most of the recent work in PoS tagging and syntactical parsing in NLP aligns with such initiative. Currently, the project counts with nearly 200 treebanks in over 100 languages. Amongst these, there are some initiatives for Portuguese (*e.g.* [Rademaker et al. 2017]), but, to the best of our knowledge, none for UGC.

Trying to fulfil this gap, we present in this paper an effort on building tokenization and UD PoS tagging systems for tweets in Brazilian Portuguese. We also introduce DANTEStocks, the corpus of stock market tweets on which we base our work, and which integrates the DANTE (Dependency-ANalised corpora of TwEets) project. To automatically add UD tags to this corpus, we propose a rule-based tokenizer and the customization of current state-of-the-art UD-based tagging strategies for Portuguese. We show that we achieve satisfactory results (98% f-score for tokenization and 95% f-score for PoS tagging), also presenting preliminary evidence of the multi-genre capacity of our PoS tagging system, thereby allowing for the construction of more robust NLP products.

The rest of this article is organized as follows. The next section focuses on briefly

[1] https://universaldependencies.org/

introducing the main related work. Section 3 then presents the details of the DANTE-Stocks corpus and the tokenization and PoS tagging methods that we explore, as well as the achieved results. Finally, our conclusions and final remarks are presented in Section 4.

2. Related work

Recently, PoS tagging and parsing have made their way back into the hot topics in NLP, specially with the advent of the UD project. Several initiatives to build new treebanks or to adapt existent treebanks to the UD formalism have arisen. Formally, a treebank is a corpus that contains sentences paired with their syntactic analyses, usually manually validated. One of the first treebanks in Brazilian Portuguese annotated according to the UD model is Bosque [Rademaker et al. 2017], which comprises well-written sentences extracted from journalistic texts, totaling 9,364 sentences.

More recently, there were also initiatives to annotate UGC texts, such as that of [Liu et al. 2018], which annotated a treebank for tweets in Engligh with 3,550 tweets in total, and that of [Sanguinetti et al. 2018], which built a treebank of 6,738 tweets written in Italian. So far, to the best of our knowledge, there is no such treebank for Portuguese. Given the available treebanks for several languages, PoS tagging and parsing systems have been developed, of which UDPipe [Straka et al. 2016, Straka 2018] is perhaps the most prominent initiative, currently in its second version, with both versions open-sourced.

The first version of UDPipe [Straka et al. 2016] relies on a perceptron network, with pre-computed features from the input text for PoS tagging, along with a bidirectional LSTM (Long Short-Term Memory) network for tokenization. The second version of UDPipe [Straka 2018], in turn, builds on a multi-layer bidirectional LSTM, using contextualized embeddings with a softmax classifier. Its input is a combination of three embedding codifications: (1) embeddings pre-trained on Wikipedia [2], (2) randomly initialized trained embeddings, and (3) character-level word embeddings using bidirectional GRUs.

Besides UDPipe, another popular tool is Udify [Kondratyuk and Straka 2019], which is a multi-task model based on BERT [Devlin et al. 2019], with an additional attention layer that captures the relations between all attention layers presented in the model, and which helps to capture low-level hierarchical information such as syntactic relations for final tasks such as PoS tagging. For tokenization, Udify uses the same word-piece tokenization from BERT [Devlin et al. 2019], where out-of-vocabulary words are split into syllables and their respective embeddings are used.

In what follows, we describe our efforts to build a UD-annotated corpus of UGC and our initiatives for developing appropriate tokenization and PoS tagging systems.

3. Tweet tokenization and PoS tagging

In this work, we build upon the corpus of tweets, written in Brazilian Portuguese for the stock market domain, described in [Vieira da Silva et al. 2020], which is publicly available for download[3]. We refer to the annotated version of this dataset as DANTEStocks, the first corpus to integrate the DANTE (Dependency-ANalised corpora of

[2] https://huggingface.co/bert-base-multilingual-uncased

[3] https://www.kaggle.com/fernandojvdasilva/stock-tweets-ptbr-emotions

TwEets) project. In its current state, DANTEStocks comprehends a total of 2,737 annotated (tokenized and PoS tagged) tweets. By the end of this project, we expect to have annotated all 4,517 tweets presented in the original corpus by [Vieira da Silva et al. 2020].

Based on ideas of in [Hovy and Lavid 2010], tweet annotation started with the grouping of tweets from the original data set into packages, following the order they are in that set. Each package was then automatically tokenized and codified according to the CoNNL-U format[4]. They were then automatically annotated with PoS tags, so as to build the starting point for review by human annotators. Each annotator received a copy of this set, with each copy containing the same tweets, but in a different (random) order.

Using a customized editing tool, each set was annotated by its corresponding annotator. Results from all annotators were then adjudicated[5] by a linguist with experience in NLP. Finally, a final version of the annotated (and adjudicated) package was built and incorporated into the final corpus. The automatic processes (tokenization and PoS tagging) were originally performed by UDPipe, being latter replaced by their customized versions that we describe in this paper.

In our experiments (described in the following subsections), 20% of the tweets from each annotated package were randomly sampled, so as to form our test set. The remaining 80% of the data in each package was then used to composed the training set, totaling 2,189 tweets for training and 548 for testing. So far, we have worked with 8 annotated packages (from a predicted total of 12), whose separation in training and testing sets may be visualized in Table 1.

Package	Total	Training data	Testing data
Subset 0	147	117	30
Subset 1	370	296	74
Subset 2	370	296	74
Subset 3	370	296	74
Subset 4	370	296	74
Subset 5	370	296	74
Subset 6	370	296	74
Subset 7	370	296	74
Total	2,737	2,189	548

Table 1. Number of tweets in the training and testing sets for each package.

3.1. Tokenization

Based on the annotated packages and on the orientations of an expert in Linguistics, we developed a rule-based tokenizer – "DANTE tokenizer" – which uses a set of regular expressions (that encode the rules) to split sentences into tokens. Built on top of the NLTK TweetTokenizer[6], the tokenizer was augmented with specific rules to deal with idiosyncrasies of the Portuguese language and the tweets from the stock market domain.

[4] CoNLL-U format is an already traditional column-based style of encoding UD annotation.

[5] *I.e.,* the cases in which annotators did not agree or did not annotate were marked and a decision was made on the final tag.

[6] https://www.nltk.org/api/nltk.tokenize.html

Within DANTEStocks, our tokenizer was responsible for:

- Removing HTML tags and formatting input texts according to the Unicode NFC standard;
- Decomposing some formal contractions found in Portuguese, such as the mixing of prepositions and articles (*e.g.* “no” → “em” + “o”); prepositions and pronouns (*e.g.* “dele” → “de” + “ele” and “deste” → “de” + “este”); and prepositions and adverbs (*e.g.* “daqui” → “de” + “aqui”);
- Splitting up monetary values (*e.g.* “R$300” → “R$” + “300”);
- Splitting up clitics (*e.g.* “localiza-se” → “localiza” + “-” + “se”), since individual words form, according to UD, the basic units of annotation; and
- Applying regular expressions to identify usual UGC phenomena in tweets and the DANTEStocks’ domain (*e.g.*, hashtags, at-mentions, URLs, emoticons, and stock market codes) and turn them into tokens, given their syntactic role in these tweets.

To illustrate the need for this last task, *i.e.* the setting up of rules to deal with specific phenomena, consider the analysis of “PETR4”, the market ticker for Petrobras’ preferred stocks. In this case, the original NLTK TweetTokenizer breaks this code into “PETR” and “4”, which makes no sense in the stock market domain, since “PETR4” refers to a well specified entity within it. To deal with this problems, a specific rule was added, so as to keep such codes as one single token.

The assessment of the tokenizer’s performance was made through traditional Precision, Recall, and Micro F-score[7] measures. However, since the tokenized sentences may be longer than their original counterparts, due to contraction expansion, metric calculations must account for token classes and positions simultaneously. Take, for example, the text spam “da PETR4”[8]. In this case, if the tokenizer outputs “da” + “PETR4”, instead of “de” + “a” + “PETR4” (*i.e.* it failed in expanding the contraction “da”), one true positive (“PETR4”), two false positives (“de” and “a”), and one false negative (“da”) will be added to the contingency tables. This procedure is illustrated in Figure 2.

Original sentence	O aumento da PETR4 não para! #continuaassim
Predicted sentence	O aumento da PETR4 não para ! #continua assim
True positives	[O, aumento, PETR4, não, para, !]
False positives	[da, #, continua, assim]
False negatives	[de, a, #continuaassim]
Precision	$\frac{tp}{tp+fp} = \frac{6}{6+4} = 0.6$
Recall	$\frac{tp}{tp+fn} = \frac{6}{6+3} \simeq 0.667$
Micro F-score	$2\frac{prec*rec}{prec+rec} = 2\frac{0.6*0.667}{0.6+0.667} \simeq 0.632$

Figure 2. Assessment of the tokenizer’s performance.

[7] Counts true/false positives and negatives globally

[8] Of PETR4, or PETR4’s.

As a benchmark for comparison, we also tested the rule-based methods NLTK Word Tokenizer [Loper and Bird 2002], spaCy [Honnibal et al. 2020], NLTK TweetTokenizer [Loper and Bird 2002] and Twikenizer[9]. The main difference between them is that while NLTK TweetTokenizer and Twikenizer were tailored to the tokenization of tweets, by adding rules specific to its writing style, NLTK Word Tokenizer and spaCy had their rules derived from other genres, with NLTK Word Tokenizer being based on the Penn Treebank [Marcus et al. 1993] and spaCy being designed for the Bosque [Rademaker et al. 2017] dataset. With the exception of SpaCy, which deals with Portuguese, all tokenizers were designed for the English language.

Results for the tokenizers' evaluation on DANTE's test set can be seen in Table 2. In this table, we show the overall Precision, Recall and Micro F-score, averaged over their individual values at each tweet in the corpus. As it turns out, DANTE Tokenizer outscores its counterparts by at least 17.5% (at recall, against NLTK TweetTokenizer), ranging up to 52.2% (at recall, against spaCy). Regarding precision, gains ranged from 20.2% (against NLTK TweetTokenizer) to 43.3% (against Twikenizer), wheras for micro f-score they ranged from 19.2% (against NLTK TweetTokenizer) to 39.9% (against spaCy).

Tokenizer	Precision	Recall	Micro F-score
NLTK Word Tokenizer	0.7333 ± 0.1482	0.7784 ± 0.1304	0.7516 ± 0.1338
NLTK Twitter Tokenizer	0.8213 ± 0.1531	0.8385 ± 0.1286	0.8275 ± 0.1379
Twikenizer	0.6890 ± 0.2122	0.8286 ± 0.1174	0.7410 ± 0.1668
spaCy	0.7822 ± 0.1390	0.6476 ± 0.1886	0.7051 ± 0.1689
DANTE Tokenizer	0.9873 ± 0.0372	0.9854 ± 0.0447	0.9861 ± 0.0400

Table 2. Tokenization's evaluation results on DANTE's test set.

While looking at these results, however, one must bear in mind that DANTE Tokenizer was tailored to the same genre and domain as the test corpus, whereas others came either from the same genre (*i.e.* tweets), but different domains, as is the case with NLTK TweetTokenizer and Twikenizer; or from different genres and domains, as is the case with NLTK Word Tokenizer and spaCy. This only provides evidence on the low scalability of these tokenizers to other domains.

3.2. PoS tagging

In this work, PoS tagging aims at assigning each token the most probable tag from a subset of 17 possible tags defined by the UD project. At this stage, however, we are not focused on determining syntactic relations between them, a task to be approached in the forthcoming months of the project. Furthermore, since training a PoS tagger usually requires large data sets, which are not available at this stage of our project, we took Bosque as our initial training set, incrementally incorporating tweet packages, as they are being produced by annotators, into it and measuring tagger accuracy in this mixed set.

With this approach, we expect to (i) overcome the problem of data limitation, taking into consideration the acquired "learned knowledge" for standard general language, as brough by Bosque, and progressively incorporating knowledge from UGC; and (ii) evaluate the multi-genre capacity of the trained PoS taggers. Hopefully, as we incorporate

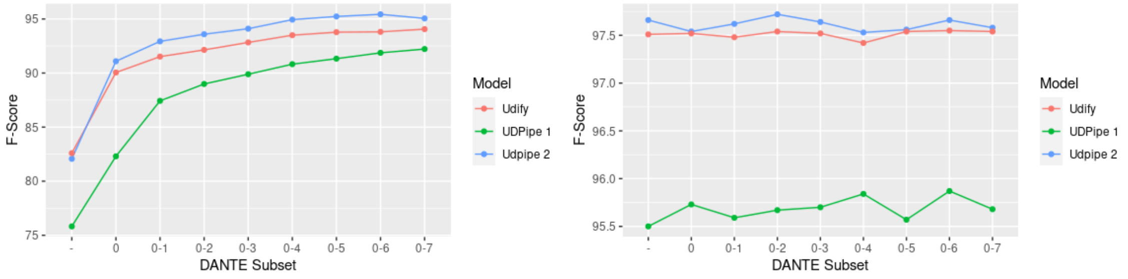
[9] https://pypi.org/project/twikenizer/

tweet information into the training set, taggers will maintain their performance in news texts, while at the same time improving their results in the tweets from DANTEStocks.

Both DANTEStocks' training set and Bosque's training and validation subsets add up to a total of 11,087 training samples, whereas their testing counterpart comprises 1,024 samples. In this work, we tested both version of UDPipe, along with Udify (*cf.* Section 2). To do so, we measured their F-score in the test set as new DANTEStocks' packs were added to the training set. At first, they were trained in Bosque's training set only. Then, after the annotation of DANTEStocks' first pack, the training was repeated with Bosque and this pack. This cycle goes on incrementally, pack by pack, until the last pack available so far. For all these runs, the testing set was kept the same, so as to determine whether there was any improvement along the way.

Results at each run are presented in Table 3. Values under the "DANTE Subset" column refer to the identification number (starting with zero) of the DANTEStocks packs added to the training set, with '-' indicating that no pack was used (*i.e.,* the system was trained only with Bosque). Next, we present the F-score values in the training set, so that differences between training and testing scores are shown. The last two columns report the systems' results when tested separately with DANTEStocks and Bosque test sets.

As expected, the more tweets are incorporated into the training set, the better the results in DANTEStocks' test set, for all tested taggers (Figure 3a). Interestingly, even though we are adding data from a different domain and genre as that of Bosque (which, in turn, might be considered as noise), tagger performance in Bosque's test set does not seem to be affected (Figure 3b). This could be an indication of the multi-genre capability of the tested taggers, even though more in-depth tests are needed to come to such a conclusion.



(a) DANTEStocks test set

(b) Bosque test set

Figure 3. Taggers performance (F-Score) across the training subsets

As it turns out, the best results were obtained by UDPipe 2 in both test sets, reaching some impressive 95% F-Score in DANTEStocks, specially considering the difficulties of automatically analysing the twitter writing style, as pointed out in Section 1. Although all differences in Bosque were found to be significant[10], only the observed differences between UDPipe 1 and 2 in DANTEStocks were found to be relevant[11], with differences between Udify and both UDPipe 1 and 2 not being of statistical significance[12].

[10] Overall Kruskal-Wallis $(df = 2) = 21.514, p << 0.001$, pairwise Dunn (with Benjamini-Hochberg p-value adjust for multiple testing) $Z = -4.626, p << 0.001$ (UDPipe 1 vs. UDPipe 2), $Z = -2.023, p = 0.043$ (Udify vs. UDPipe 2); and $Z = 2.603, p = 0.014$ (Udify vs. UDPipe 1), at the 95% confidence level.

[11] $Z = -3.059, p = 0.007$

[12] $Z = 1.930, p = 0.080$ (Udify vs. UDPipe 1); and $Z = -1.128, p = 0.259$ (Udify vs. UDPipe 2).

Model	DANTE Subset	train f-score	DANTE test f-score	Bosque test f-score
UDPipe 1	-	98.81%	75.82%	95.50%
	0	99.27%	82.30%	95.73%
	0-1	99.75%	87.43%	95.59%
	0-2	99.73%	88.99%	95.67%
	0-3	99.67%	89.89%	95.70%
	0-4	99.65%	90.82%	95.84%
	0-5	99.67%	91.33%	95.57%
	0-6	99.60%	91.87%	95.87%
	0-7	99.58%	92.22%	95.68%
UDPipe 2	-	99.60%	82.07%	97.66%
	0	99.57%	91.09%	97.54%
	0-1	99.51%	92.93%	97.62%
	0-2	99.45%	93.59%	97.72%
	0-3	99.43%	94.10%	97.64%
	0-4	99.44%	94.94%	97.53%
	0-5	99.41%	95.23%	97.56%
	0-6	99.42%	95.43%	97.66%
	0-7	98.86%	95.05%	97.58%
Udify	-	98.13%	82.59%	97.51%
	0	98.11%	90.05%	97.52%
	0-1	98.01%	91.52%	97.48%
	0-2	97.90%	92.14%	97.54%
	0-3	97.87%	92.83%	97.52%
	0-4	97.81%	93.50%	97.42%
	0-5	97.76%	93.78%	97.54%
	0-6	97.65%	93.81%	97.55%
	0-7	97.62%	94.06%	97.54%

Table 3. PoS tagging results

F-Score results for each tag[13], as produced at the DANTEStocks test set, are presented in Table 4. As shown in the table, results range from 34.29% (with INTJ and Udify) to 99.33% (with CCONJ and UDPipe 2), with worst cases happening with the INTJ and X tags for all taggers. A possible reason for such anomalous values in these two tags, as illustrated in Figure 4, might be the fact that INTJ has fewer occurrences in the corpus, whereas X was used as a "left over" tag, being applied in cases of typos, pre-processing errors, or when annotators could not find a proper tag for the token. Interestingly, token-wise differences between taggers were not significant[14]. The confusion matrix for UDPipe 2 best model can be seen in Figure 5, which elucidates our analysis on INTJ and X classes. The confusion matrices for UDPipe 1 and Udify are available at our Github repository [15].

[13] The PART tag was omitted because there were no occurrences of this tag in DANTEStocks.

[14] Kruskal-Wallis $(df = 2) = 1.351, p = 0.509$

[15] https://github.com/huberemanuel/dante-tagging-eniac2021

Tag	UDPipe	UDPipe 2	Udify
ADJ	87.01	90.46	91.36
ADP	95.04	97.45	97.21
ADV	93.87	94.69	91.10
AUX	88.20	92.39	90.86
CCONJ	99.10	99.33	99.10
DET	97.40	97.70	97.30
INTJ	75.56	44.44	34.29
NOUN	91.63	94.20	92.98
NUM	96.51	97.99	97.27
PRON	88.89	92.10	88.64
PROPN	92.32	95.77	94.78
PUNCT	99.11	99.30	98.88
SCONJ	84.07	89.90	88.67
SYM	97.84	99.13	97.80
VERB	93.22	94.88	94.56
X	74.05	79.06	73.52

Table 4. F-score per class after training with all DANTE subsets (0-7)

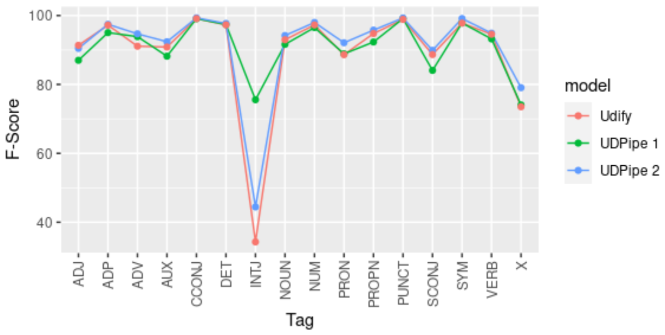


Figure 4. Models' performance at each tag found in the corpus.

4. Conclusion

This paper presented our current effort in building tokenization and PoS tagging services for tweets written in Brazilian Portuguese, inline with the Universal Dependencies international model, and so adding up to the increasing amount of resources devoted to this widely adopted standard for morpho-syntactical annotation. Moreover, we introduced DANTEStocks, a corpus of tweets from the financial market, which served as the basis for our experiments, also showing some preliminary evidence that our PoS tagging strategies have multi-genre capacity, producing good results for tweets while, at the same time, holding their performance in news texts.

This work comes as a part of a larger project that aims at fostering research on syntax and parsing for Brazilian Portuguese: the POeTiSA project[16]. Our final goal in this broader project is to build a large multi-genre treebank for Portuguese, also developing state-of-the-art PoS tagging and parsing systems for this language. Within this context,

[16] https://sites.google.com/icmc.usp.br/poetisa

DANTEStocks comes up as one of the corpora, being the first to integrate the DANTE initiative – a treebank of corpora for tweets, which is itself part of POeTiSA.

As our next steps, we intend to refine our evaluation of the presented taggers' performance, by using the full set of DANTEStocks' annotated data (which was not yet fully annotated during the writing of this article), and to explore other directions for tokenization (*e.g.*, to use sequence models such as the one presented in [Devlin et al. 2019]). We also envision the syntactic annotation of the tweets according the UD guidelines.

As a final remark, it is worth mentioning that, although in this article we have focused on describing how systems were developed and tested, the adoption of the UD model required an extensive linguistic study and adaptation of its guidelines to the Portuguese language, along with the development of strategies and guidelines for the annotation of different genres. These are results that are still under construction, and which will be left for future publications, more centered on the linguistic aspects of the project.

Acknowledgments

The authors are grateful to the USP/IBM/FAPESP Center for Artificial Intelligence (C4AI - grant 2019/07665-4), the Facens University Center, and the researchers and linguists of the POeTiSA project for their hard work on the DANTE treebank.

References

de Sousa, R. C. C. and Lopes, H. (2019). Portuguese pos tagging using blstm without handcrafted features. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 120–130.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fonseca, E., Rosa, J., and Aluísio, S. (2015). Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *Journal of the Brazilian Computer Society*, 21(2):1–14.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Hovy, E. and Lavid, J. (2010). Towards a 'science' of corpus annotation: A new methodological challenge for corpus linguistics. *International Journal of Translation*, 22(1):13–36.

Kondratyuk, D. and Straka, M. (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 (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., and Smith, N. A. (2018). Parsing tweets into Universal Dependencies. In *Proceedings of the 2018 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 965–975, New Orleans, Louisiana. Association for Computational Linguistics.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., and Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

Rademaker, A., Chalub, F., Real, L., Freitas, C., Bick, E., and de Paiva, V. (2017). Universal dependencies for portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling)*, pages 197–206, Pisa, Italy.

Sanguinetti, M., Bosco, C., Cassidy, L., Çetinoğlu, Ö., Cignarella, A. T., Lynn, T., Rehbein, I., Ruppenhofer, J., Seddah, D., and Zeldes, A. (2020). Treebanking user-generated content: A proposal for a unified representation in Universal Dependencies. In *Proceedings of the 12th Language Resources and Evaluation Conference*.

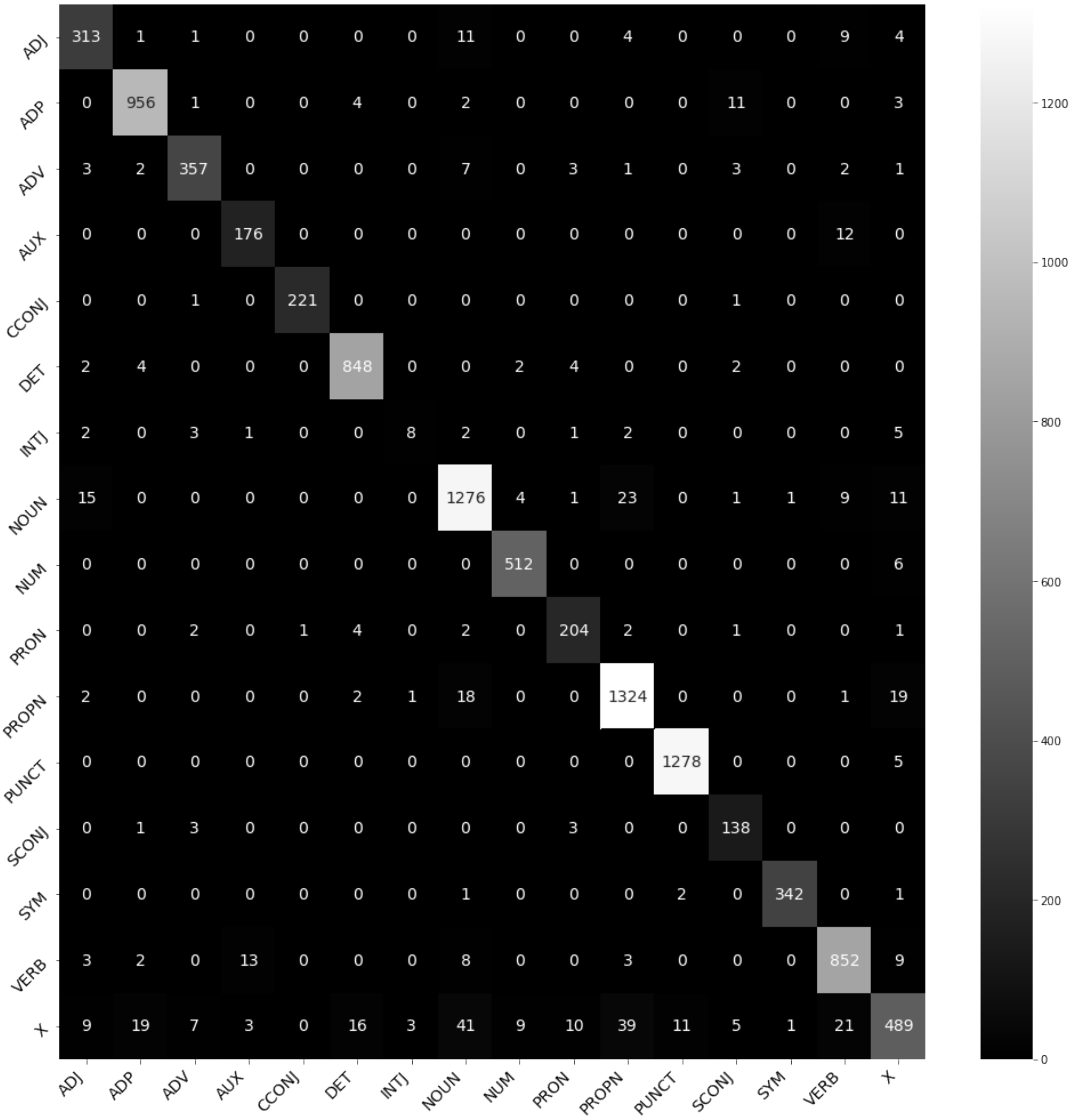
Sanguinetti, M., Bosco, C., Lavelli, A., Mazzei, A., Antonelli, O., and Tamburini, F. (2018). PoSTWITA-UD: an Italian Twitter treebank in Universal Dependencies. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Straka, M. (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, Brussels, Belgium. Association for Computational Linguistics.

Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Vieira da Silva, F. J., Roman, N. T., and Carvalho, A. M. (2020). Stock market tweets annotated with emotions. *Corpora*, 15(3):343–354.

A. Confusion Matrix



	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
ADJ	313	1	1	0	0	0	0	11	0	0	4	0	0	0	9	4
ADP	0	956	1	0	0	4	0	2	0	0	0	0	11	0	0	3
ADV	3	2	357	0	0	0	0	7	0	3	1	0	3	0	2	1
AUX	0	0	0	176	0	0	0	0	0	0	0	0	0	0	12	0
CCONJ	0	0	1	0	221	0	0	0	0	0	0	0	1	0	0	0
DET	2	4	0	0	0	848	0	0	2	4	0	0	2	0	0	0
INTJ	2	0	3	1	0	0	8	2	0	1	2	0	0	0	0	5
NOUN	15	0	0	0	0	0	0	1276	4	1	23	0	1	1	9	11
NUM	0	0	0	0	0	0	0	0	512	0	0	0	0	0	0	6
PRON	0	0	2	0	1	4	0	2	0	204	2	0	1	0	0	1
PROPN	2	0	0	0	0	2	1	18	0	0	1324	0	0	0	1	19
PUNCT	0	0	0	0	0	0	0	0	0	0	0	1278	0	0	0	5
SCONJ	0	1	3	0	0	0	0	0	0	3	0	0	138	0	0	0
SYM	0	0	0	0	0	0	0	1	0	0	0	2	0	342	0	1
VERB	3	2	0	13	0	0	0	8	0	0	3	0	0	0	852	9
X	9	19	7	3	0	16	3	41	9	10	39	11	5	1	21	489

Figure 5. Confusion matrix for UDPipe 2 trained on all DANTE subsets