UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

CORPORA BUILDING PROCESS ACCORDING TO THE UNIVERSAL DEPENDENCIES MODEL: AN EXPERIMENT FOR PORTUGUESE

LUCELENE LOPES
MAGALI SANCHES DURAN
MARIA DAS GRAÇAS VOLPE NUNES
THIAGO ALEXANDRE SALGUEIRO PARDO

Nº 439

RELATÓRIOS TÉCNICOS

São Carlos – SP
Mar./2022

Natural Language Processing initiative (NLP2) of the Center for Artificial Intelligence (C4AI) of the University of São Paulo, sponsored by IBM and FAPESP

POeTiSA

POrtuguese processing – Towards Syntactic Analysis and parsing

Corpora building process according to the Universal Dependencies model: an experiment for Portuguese

Lucelene Lopes, Magali Sanches Duran,
Maria das Graças Volpe Nunes, and
Thiago Alexandre Salgueiro Pardo

March 2022

NILC Technical Report

1 Introduction

This technical report proposes a process to build a fully revised corpus from a pure textual source. The goal is to obtain a large corpus using Universal Dependencies (UD) [12] [13] format and tagset with morphological and morphosyntactic information. Together with the definition of the proposed process, we instantiate it by building a corpus for Portuguese, thus presenting the achievements of our process in a practical case. As such, our work is in line with several recent efforts to build UD treebanks for low resource languages as Occitan [10], North Aribizi [18], Laz [22], and Occitan, Alsatian, and Piccard [2], but also for better resourced ones as Korean [4], Polish [24], and Persian [15], or even for variations of well resourced languages as Latin [3] and Italian [9].

This work is part of the POeTiSA project[1], whichs stands for "POrtuguese processing - Towards Syntactic Analysis and parsing". POeTiSA is a long term project that aims at growing syntax-based resources and developing related tools and applications for Brazilian Portuguese language, looking to achieve world state-of-the-art results in this area. On the resource side, it focuses on the production of a large and comprehensive multi-genre corpus of UD-based part of speech and syntactically annotated texts (a treebank), including mainly news texts and user-generated content (tweets and online comments) [14]. Regarding the tools, the project aims to investigate recent neural and distributional-based methods for training robust parsing models for Portuguese. The project also envisions the production of applications on opinion mining and sentiment analysis tasks that may benefit from syntactic knowledge, as opinion summarization, helpfulness prediction, aspect identification, deception detection and emotion classification. The project is part of the Natural Language Processing initiative (NLP2) of the Center for Artificial Intelligence[2] (C4AI) of the University of São Paulo, sponsored by IBM and FAPESP (grant 2019/07665-4). The center is part of the FAPESP Engineering Research Centers Program[3] and is committed to state-of-the-art research in Artificial Intelligence, exploring both foundational issues and applied research.

This technical report is organized as follows: next section describes the proposed process to build a verified corpus with correct sentences; Section 3 gives details on how the pure text preparation was conducted for our experiment; Section 4 describes the automatic annotation and corrections for our experiment; Section 5 summarizes the manual annotation to correct the PoS tag of all tokens; Section 6 details the correction of tokens' lemmas and Section 7 does the same for the correction of morphological features; Section 8 summarizes the improvements brought by each step of our experiment in terms of correct information towards the verified annotated corpus produced; the last section draws some final remarks and suggests future work.

2 Proposed Process

Our proposed process is composed of five stratified steps as depicted in Figure 1. We believe that it is necessary a human annotation in order to correct as many errors as possible, but, given the size of the corpus, we need to rely on some form of automatic annotation using available parsers and language models. Obviously, the use of a parser in a blind way does not grant the intended correction of the produced corpus, thus, after the automatic annotation, a verification of the information is done in the last steps.

The first step is the preparation of the pure text from the source corpus, including performing sentence segmentation, tokenization, and discard of unwanted sentences. The result of this step is a textual representation of sentences ready to be processed by a Part of Speech (POS) tagger or syntactic parser.

[1] `https://sites.google.com/icmc.usp.br/poetisa`
[2] `http://c4ai.inova.usp.br/`
[3] `https://www.fapesp.br/cpe/`

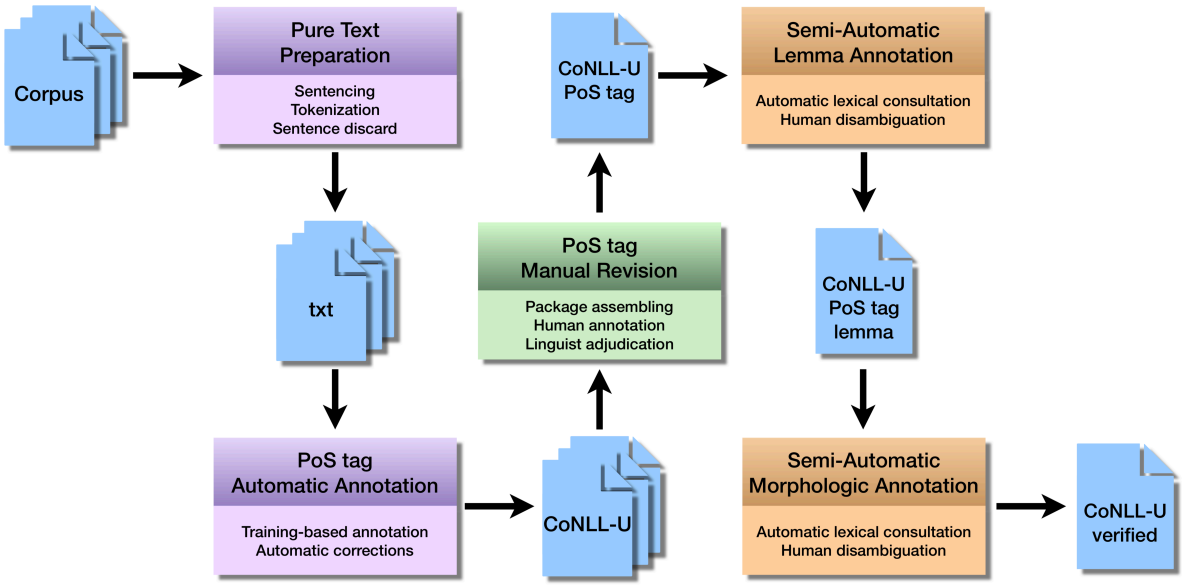

Figure 1: Proposed process to produce a verified annotated corpus in UD.

The second step is the automatic annotation of the pure text using a parser followed by the correction of the parsers' output errors that can be automatically fixed. The result of this second step is a CoNLL-U representation[4] of the corpus that needs to be revised. For example, the sentence "*Três reportagens da Folha ganham Prêmio Petrobras de Jornalismo.*" is represented in the CoNLL-U format as depicted in Figure 2. Note that, since our goal is to provide a revised corpus with morphological information, in the UD representation only the columns ID, FORM, LEMMA, UPOS, and FEAT will be filled.

ID	FORM	LEMMA	UPOS	XPOS	FEAT	HEAD	DEPREL	DEPS	MISC
1	**Três**	três	NUM	_	Gender=Neut|NumType=Card	2	nummod	_	_
2	**reportagens**	reportagem	NOUN	_	Gender=Fem|Number=Plur	6	nsubj	_	_
3-4	**da**	_	_	_	_	_	_	_	_
3	**de**	de	ADP	_	_	5	case	_	_
4	**a**	o	DET	_	Definite=Def|Gender=Fem|Number=Sing|PronType=Art	5	det	_	_
5	**Folha**	Folha	PROPN	_	Gender=Fem|Number=Sing|Proper=Yes	2	nmod	_	_
6	**ganham**	ganhar	VERB	_	Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin	0	root	_	_
7	**Prêmio**	Prêmio	PROPN	_	Gender=Masc|Number=Sing|Proper=Yes	6	obj	_	_
8	**Petrobras**	Petrobras	PROPN	_	Gender=Fem|Number=Inv|Proper=Yes	7	flat:name	_	_
9	**de**	de	ADP	_	_	10	case	_	_
10	**Jornalismo**	Jornalismo	PROPN	_	Gender=Masc|Number=Sing|Proper=Yes	7	nmod:name	_	_
11	**.**	.	PUNCT	_	_	6	punct	_	SpacesAfter=\r\n

Figure 2: Example of UD annotation of a Portuguese sentence: "*Três reportagens da Folha ganham Prêmio Petrobras de Jornalismo.*"

The third step is the manual revision of all tokens and PoS tags, following specific criteria to assemble revision packages to human annotation and linguist adjudication. The result of this third step

[4]For a full description of UD format and tagset as well as the CoNLL-U format, please visit the formal definition at `https://universaldependencies.org/format.html`.

is a reliable CoNLL-U representation where the columns ID, FORM, and UPOS are correct.

The fourth step is the semi-automatic correction of lemmas, starting from the corrected PoS tags, using a lexical resource and human disambiguation. After the fourth step, the CoNll-U representation is also correct for the information of column LEMMA.

Finally, the fifth step is a similar semi-automatic task to correct the morphological features. This final result is intended to complete the morphological annotation assuring the correctness of column FEAT of the CoNLL-U representation.

We decided for such order of steps, since it allows a better efficacy (fewer errors) and a better efficiency (faster results). Once the token has the correct PoS tag, the lemma options tend to be reduced, and similarly, once the lemma is correct, the morphological features options tend to be reduced too.

As mentioned, together with the process definition, we present a practical experiment in the next sections (Sections 3 to 7). These sections detail the five steps producing a verified corpus (with 8,420 sentences and 168,399 tokens) from a large source, the Folha Kaggle corpus [17]. Additionally, Section 8 summarizes the achievements of the whole process experimented over the Folha Kaggle corpus.

3 Preparing the Pure Text

The first step of the proposed process is to prepare the corpus pure text. The next sections detail the proposed principles for this step (Sec. 3.1) and the instantiation in our practical example (Sec. 3.2).

3.1 Principles to Prepare the Pure Text

An important point to be considered in a treebank preparation is to minimize the chance of errors due to sentencing, tokenization, and unappropriated text. In this section we describe the efforts to deal with these three aspects preparing our corpus.

3.1.1 Sentencing

Initially, we made an automatic analysis of the full text to detect possible mistakes on the formation of sentences. We assumed that every valid sentence has to:

- start with a capital letter, a digit, or symbols (*e.g.*, quotation marks, parenthesis) followed by capital letter or digit;
- end with one of four possible ending punctuations: final period "."; exclamation point "!"; question mark "?"; or suspension points "...".

The challenge to perform sentencing based on such assumptions is how to avoid the end of sentence when the period sign (".") has other uses, as: abbreviations (*e.g.*, *"Sr."*, *"Apto."*), decimal separators (*e.g.*, *"23.518"*, *"2.500,00"*), and electronic addresses (*e.g.*, *"drive.google.com"*, *"www.youtube.com"*). To cope with that we developed a simple code that ignores a period sign if it is not followed by a blank space. This avoid errors with both numbers and electronic addresses. Therefore, only period signs (or equivalents "!", "?" and "...") followed by a blank sign are considered candidates to end a sentence. However, to cope with the use of period sign in abbreviations, we build a list of usual abbreviations in Portuguese, and for such cases (*e.g.*, *"sáb."*, *"apt."*) we ignored period signs that compose a known abbreviation[5]. Finally, a candidate of end of sentence is retained as such if it is either followed by a capital letter, a symbol, a digit, or the end of file. It is important to notice that such sentencing approach is a heuristic as the sentencing problem is likely to be undecidable [6, 16].

[5]For our practical experiment, we considered a list of 397 usual abbreviations in Portuguese. These abbreviations, as well as all other data mentioned in this paper, are available at POeTiSA project webpage.

3.1.2 Tokenization

The tokenization task occurs after the sentencing has already produced in a single string all characters of a sentence. The first step of tokenization is to split the sentence string into chunks (substrings that are separated by blank spaces). Then, to each of those chunks, a recursive process tries to separate the tokens within the chunk. Basically, the following punctuation symbols are set apart from the beginning of the chunk:

simple quotes:	‘	double quotes:	“	hyphens:	-
open parenthesis:	(	open brackets:	[	open curly braces:	{

Similarly, the following punctuation symbols are set apart from the end of the chunk:

simple quotes:	’	double quotes:	”	hyphens:	-
close parenthesis:	)	close brackets:	]	close curly braces:	}
colon:	:	semicolon:	;	final period:	.
exclamation point:	!	question mark:	?	suspension points:	...

Therefore, the following chunk: *(“cliente”)...* will be tokenized into six tokens:

(“ cliente ”) ...

The second step deals with clitic verbs in order to split enclisis (*e.g.*, *“manter-se”*, *“tomá-la”*) and mesoclisis (*e.g.*, *“manter-se-a”*, *“tomá-la-ia”*) into separated tokens. This is done considering all words with an hyphen followed by a reflexive pronoun according to the following list:

“me” “te” “se” “lhe” “nos” “vos” “lhes”

“o” “a” “os” “as” “lo” “la” “los” “las”

All other words containing hyphens are kept as a single token, for example: *“guarda-chuva”*, *“segundas-feiras”*.

Contracted words, mostly prepositions and determiners, that need to be dealt as two tokens are not split by our tokenization process. These contracted words may be unequivocal as *“do”* that always will be *“de”* + *“o”* (ADP+DET), but also words as *“nos”* that can be either a single word (PRON) or the contracted words *“em”* + *“os”* (ADP+DET). For that reason, we decided not to split the contracted words during the pure text preparation. During the automatic annotation (see Section 4), the split of contracted words is performed taking into account contextual information.

3.1.3 Discarding Unappropriated Sentences

An unappropriated sentence for our goals is a sentence that is not grammatically correct in a strict sense as conveying a message structured around a verb and that is long enough to have the correctness established. Therefore, we discard sentences where:

- There are less than 2 tokens;
- There are unmatched quotations, parentheses, or brackets;
- There is no final punctuation;
- The first token (except quotations, parenthesis, and brackets) does not start with a capital letter or digit;
- There is only functional information, as the definition of telephone numbers, electronic or physical addresses, date and time, prices, *etc.* - those cases are identified by sentences where the second token is a colon (“:”).

One may be worried about discarding sentences that are candidate cases for annotation, but, since in our case the corpus is very large, the loss is not relevant (0.015%, as it will be seen in Section 3.2). Moreover, given the benefits of producing a more reliable annotation of the remaining sentences justifies the sentence discard.

3.2 Instantiating the Pure Text Preparation

To illustrate our proposed process, we conduct the previously described step to the Folha Kaggle corpus [17] and we indicate the amount of produced sentences and tokens. Applying the sentencing and tokenization tasks to all 167,048 news article texts in Folha Kaggle corpus, the process produced 3,614,115 sentences (84,795,823 tokens).

Applying the sentence discarding criteria to the produced 3,614,115 sentences, 52,196 sentences (0.015%) were considered unappropriated, and, therefore, discarded. The resulting corpus was composed of 3,561,919 sentences (83,990,533 tokens). However, for the next experiments of this paper, we limit ourselves to only the first 5 thousand news article texts that correspond to 124,784 sentences, which holds 2,541,722 tokens (20.4 tokens per sentence).

Between the definition and execution of this task, two people were involved, one linguist and one computer scientist, and this step took one month to be completed.

4 Automatic Annotation

The second step of the proposed process is to perform the automatic annotation of the pure text. The next sections detail the proposed principles for this step (Sec. 4.1) and the instantiation in our practical example (Sec. 4.2).

4.1 Principles to Automatic Annotation

Once the sentences are duly processed (sentencing, tokenization, and discard), we propose performing a preliminary annotation using a performant parser, for example, UDPipe parser [20, 19], using the best available language model.

An additional concern is to assure an easy access to the provenance of each sentence, and for that reason we suggest the adoption of a basic format to identify each sentence. Specifically, we intend that each sentence could be identified by the name of the corpus it belongs, plus the number of each document within the corpus, and finally a sentence number within the document.

Once the pure text is annotated, we propose the a series of automatic corrections to adjust the parser output and reduce simple mistakes that may occur. The use of UDPipe, as other approaches based on neural networks, is very effective to accurately estimate the required output, but it does not guarantee the exact prediction of some simple situations, for example, the assignment of PoS tags for functional words [21, 8].

Specifically, we propose three forms of automatic corrections:

- automatic correction of PoS tags for tokens of closed classes that unambiguously have to be in a specific PoS tag;
- automatic correction of usual PoS tag mistakes for specific tokens;
- automatic correction of PoS tags for usual co-occurring token sequences that are non-ambiguous.

4.2 Instantiating the Automatic Annotation

We detail the instantiation in terms of automatic annotation using UDPipe parser [20, 19] (Sec. 4.2.1), the automatic correction of PoS tags for tokens of closed classes (Sec. 4.2.2), usual parsing mistakes (Sec. 4.2.3), and non-ambiguous co-occurring token sequences (Sec. 4.2.4). Finally, we provide some numbers of the results of these tasks (Sec. 4.2.5).

4.2.1 PoS tag Automatic Annotation using UDPipe

We proceed the automatic annotation of the sentences using the UDPipe 2 parser. As language model, we chose the BOSQUE-UD [23], one of three available UD corpora in Portuguese, composed by 9,364 sentences (210,957 tokens).

For this task, performing the automatic annotation by UDPipe with BOSQUE-UD proved to be effective to correctly tokenize the contracted words that were not tokenized at the pure text preparation, with only two problematic cases that were dealt specifically as described in the next section (see Section 5.1.1).

The sentence identification was defined as mentioned, stating the origin of each sentence, giving the corpus, the document, and sentence numbering. For example, the sixth sentence of the second document of the Folha Kaggle corpus is identified by *FOLHA_DOC000002_SENT006*.

The choice of how many digits are allocated to number the documents and sentences may vary, as each corpus has quite different definitions and sizes to determine what is a document in its context. For our example, we assumed that, for the Folha Kaggle, each document is the text of a news article published online. Since the data bank holds 167,048 news and none of these news has more than 999 sentences, we choose 6 digits to identify the document and 3 digits to identify the sentence within each document.

4.2.2 Automatic Correction of PoS tags for Tokens Belonging to Closed Classes

The first automatic correction focused on the PoS tags for tokens of closed classes that unambiguously have to be in a specific PoS tag. These corrections were executed by identifying the target tokens and overruling the UDPipe assigned PoS tag by the unambiguous PoS tag as originally proposed by [8].

Closed class words, due to their classes finiteness, may be tackled differently from open class words. Words from closed classes may belong to one or more than one class. When a closed class word belongs to a unique class, we can unambiguously assign a unique PoS tag to it. With some effort and corpus analyses, we have identified several unambiguous words of closed classes, which we consistently annotated with a unique PoS tag in our corpus.

UD defines 7 PoS tags for closed classes of words: ADP, CCONJ, DET, NUM, PRON, PART and SCONJ. Closed classes are those that have a finite set of possible words. All functional words belong to closed classes as well as two classes of content words: cardinal numbers and pronouns. In Portuguese, primitive adverbs (ADV in UD), those not terminated in *-mente* (-ly adverbs), are also a closed class. This is also the case of ordinal numbers that can be seen as a closed subset of the adjectives class, and then can be always assigned in UD as ADJ. Similarly, some auxiliary verbs (AUX in UD) are a closed class for the flexions of verbs "*ser*" and "*estar*" ("to be" in English).

The automatic correction of PoS tags for non-ambiguous tokens of closed class words is performed for 285 tokens, and Table 1 illustrates the tokens considered to each of our target closed classes[6]. In this table, we indicate the total number of tokens considered in each class (#total), the number of tokens that are non-ambiguous (#n-amb.), and a couple of example of non-ambiguous tokens.

[6]The full lists of considered closed classes tokens, with ambiguity indication, as well as all other data mentioned in this report, are available at POeTiSA project webpage.

Table 1: Closed PoS tags considered in our work.

PoS tag	Definition	#total	#n-amb.	Examples
ADJ	Adjectives (ordinals)	115	86	"*nono*" ("nineth" in English) "*décima*" ("tenth" in English)
ADP	Adpositions	36	12	"*contra*" ("against" in English) "*de*" ("of" in English)
ADV	Natural Adverbs	161	54	"*acima*" ("above" in English) "*não*" ("no" in English)
AUX	Auxiliary verbs "*ser* and *estar*"	294	77	"*é*" ("is" in English) "*sou*" ("am" in English)
CCONJ	Coordenative conjunctions	22	8	"*e*" ("and" in English) "*mas*" ("but" in English)
DET	Determiners	127	4	"*cujo*" ("which" in English) "*diversos*" ("several" in English)
NUM	Numerals (cardinals)	49	11	"*sete*" ("seven" in English) "*onze*" ("eleven" in English)
PRON	Pronouns	160	32	"*ela*" ("she" in English) "*isso*" ("this" in English)
SCONJ	Subordinate conjunctions	14	1	"*conquanto*" ("although" in English)
Total Non-Ambiguous Tokens			285	

4.2.3 Automatic Correction of Usual PoS tag Mistakes for Specific Tokens

We made five kinds of automatic changes in specific tokens consistently assigned by the parser. The first one is due to not following the current UD Guidelines. The other four kinds are due to project decisions we made differently from the corpus used to train the parser. All these five kinds of changes have not been reported in [8], as they are not useful for every corpus, only for those using the same parser/training set as ours.

A recurrent problem we observed is the assignment of the PoS tag SCONJ to prepositions followed by non-finite verb clauses. Only finite verb clauses take SCONJ and, therefore, the prepositions "*a*", "*por*", and "*para*" tagged as SCONJ followed by non-finite verb clauses was automatically changed to the PoS tag ADP (the PoS tag in UD tagset for adpositions, that is, prepositions and postpositions).

In our project, we define the modifiers "*mesmo*", "*mesma*", "*mesmos*" and "*mesmas*" (in English "same") as determiners and not adjectives. Therefore, their PoS tags were automatically changed from ADJ to DET. Note that the adverb "*mesmo*", invariable, maintain its PoS tag ADV.

A project decision, as well, is to annotate as adjective the words "*melhor*", "*melhores*", "*pior*" and "*piores*" (in English "better", "best", "worse", "worst"), regardless of whether they were used as nominals, due to the ellipsis of the modified noun. Therefore, their PoS tags were automatically changed from NOUN and PRON to ADJ.

Following UD Guidelines, the ordinal numerals are annotated as adjectives (ADJ). In Portuguese, "*primeiro*" ("first" or "firstly") may be used as an adverb as well and is assigned the PoS tag ADV when it conveys the meaning of "firstly". However, other ordinal numerals, when used to enumerate things in a sentence, should keep their ADJ PoS tag. Therefore, the words "*segundo*", "*terceiro*", "*quarto*", *etc.*, if annotated as ADV, become ADJ.

We decided to annotate the words "*gente*", "*cima*" and "*vez*" with the PoS tag of their original category - NOUN, even if "*gente*" is used in an expression alternative to the nominal pronoun "*nós*" and even if "*cima*" and "*vez*" are used in adverbial expressions. Therefore, if annotated as ADP, ADJ, ADV or PRON, these words become NOUN.

4.2.4 Automatic Correction of PoS tags for Usual Co-occurring Token Sequences

We observe that some ambiguous tokens of closed classes became non-ambiguous when found inside n-gram sequences of tokens that sometimes constitute multiword expressions, and other times are merely co-occurring tokens. As these n-gram sequences are highly frequent and not always correctly annotated by the parser, the pre-definition of their PoS tags followed by automatic correction favours consistency of annotation.

The automatic correction of PoS tags for usual co-occurring tokens that are non-ambiguous was applied to 141 sequences[7] that are illustrated in Table 2. In this table, we indicate the number of sequences according to the number of tokens within, and show a couple of examples with the associated PoS tags.

Table 2: Non-ambiguous co-occurring token sequences considered in our work.

#tok.	#seq.	Examples	in English
2	65	"*cada um*" DET PRON	*each one*
		"*em cima*" ADP NOUN	*on top*
3	59	"*a partir de*" ADP VERB ADP	*starting from*
		"*mais ou menos*" ADV CCONJ ADV	*more or less*
4	15	"*nem a o menos*" ADV ADP DET NOUN	*not even*
		"*a não ser que*" ADP ADV AUX SCONJ	*more or less*
5	2	"*em a medida em que*" ADP DET NOUN ADP PRON	*in the means that*
		"*mais de o que nunca*" ADV ADP PRON PRON ADV	*more than ever*
Total	141		

4.2.5 Numerical Results for the Automatic Annotation

We started identifying all 124,784 sentences (2,541,722 tokens) and processing them all through UDPipe using BOSQUE-UD as language model. The result was a repository of 124,784 CoNLL-U annotated sentences with 3,152,055 tokens (an average of 25.3 tokens per sentence). This increase of 610,333 tokens from the pure text version happened because UDPipe tokenized:

- all contracted words

  e.g., the word "*do*" was split into "*de*" (ADP) and "*o*" (DET);

- compound words

  e.g., the word "*fim-de-semana*" was split into "*fim*", (NOUN), "*de*" (ADP), and "*semana*" (NOUN);

- other mixed compounds

  e.g., the date "*07/09*" was split into "*07*" (NUM), "*/*" (PUNCT), and "*09*" (NUM).

[7]The full lists of considered co-occurring tokens sequences, as well as all other data mentioned in this paper, is available at POeTiSA project webpage.

While the tokenization of contracted words rarely produced errors, almost all cases of compounds should not be tokenized following UD annotation principles. For the mentioned examples, only the tokenization of the word "*do*" into "*de*" (ADP) and "*o*" (DET) was correct. Both examples of the compound word "*fim-de-semana*" and the date "*07/09*" had to be fixed during the manual revision of structural problems (see Section 5).

Applying the PoS tag automatic adjustments to the UDPipe annotated sentences, we have obtained the statistics in Table 3 that refers to the first five thousand news articles of Folha Kaggle composed by 124,784 sentences (3,152,055 tokens). In this table, we present the number of tokens detected and, among those, the ones corrected according to:

- **Closed Classes**: words of closed classes with non-ambiguous PoS tag (Section 4.2.2);
- **Usual Mistakes**: usual PoS tag mistakes for specific tokens (Section 4.2.3);
- **Usual Sequences**: usual co-occurring token sequences with non-ambiguous PoS tags (Section 4.2.4).

Table 3: Statistics of the automatic adjustment applied to the sentences of the first five thousand news articles of Folha Kaggle.

correction	total tokens	detected tokens	% of total	corrected tokens	% of total	% of detected
Closed Classes	3,152,055	1,005,096	31.89%	22,302	0.71%	2.22%
Usual Mistakes		24,848	0.79%	24,848	0.79%	100.00%
Usual Sequences[1]		7,732	0.25%	3,353	0.11%	43.37%
totals		1,037,676	32.92%	50,503	1.60%	4.87%

[1]The numbers in this table are the numbers of tokens. The total number of detected usual co-occurring token sequences with non-ambiguous PoS tags was 15,600. The number of those sequences where at least one of their PoS tags was altered from UDPipe original annotation was 3,008.

The results in Table 3 show a large number of automatic corrections that surely improves the original outcome of UDPipe. It is particularly noticeable the large number of words belonging to closed classes and the usual mistakes that are responsible each to correct more than 20 thousand tokens each.

Between the definition and execution of this task, two people were involved, one linguist and one computer scientist, and this task took a little more than two months to be completed.

5 PoS tag Manual Annotation

The third step of the proposed process is to perform the manual revision of PoS tag automatic annotation produced in the second step. The next sections detail the proposed principles for this step (Sec. 5.1) and the instantiation in our practical example (Sec. 5.2).

5.1 Principles to Perform the PoS tag Manual Annotation

Having the corpora automatically annotated using UDPipe and corrected by our automatic adjustments, we performed the manual revision of the PoS tag annotation. To perform this task we gather a set of linguists to manually review sets of annotated sentences. Basically, we assembled packages of sentences according to specific criteria, as the sentence size (number of tokens) and specific features (tokens of interest) to be analyzed by several linguists in parallel. Each of those packages was replicated ten times

and each linguist received one of these replicas to be reviewed using the Arborator-Grew annotation tool [7]. Each replica had the sentence order shuffled to avoid bias from linguists.

Each linguist had to decide for each token to keep the automatically assigned PoS tag, to change it to one of the other 16 UD PoS tags[8], or to identify a sentence structural problem that could be: a sentencing error, a tokenization error, or a typing error. Once the linguists reviewed all sentences of their package, all reviewed replicas are analyzed and an agreement measure is computed according to the formula [1]:

$$\text{agr}_i = \frac{1}{\binom{c}{2}} \sum_{k \in K} \binom{n_{ik}}{2} = \frac{1}{c(c-1)} \sum_{k \in K} n_{ik}(n_{ik} - 1)$$

where n_{ik} represents the number of annotators, taken pairwise, assigning the label k, from a set of K possible labels, to the same token i. Agreement values range from 0%, representing total disagreement (*i.e.*, each annotator assigns a different label), to 100% (full agreement - all annotators assigned the same label to the token).

The resulting suggestion given by each of the annotators and their agreement are considered by a chief linguist to adjudicate the issues, in order to assign a definitive PoS tag to each token.

5.1.1 Manual Identification of Structural Problems

During the manual revision of PoS tags, three kinds of structural problems were identified:

- **Corpus typos**: when sentences of the corpus have typing errors, this problem kind motivates the discard of the whole sentences. An example of such problem happens in the sentence *"A ora do encontro não foi divulgada."* - the token *"ora"* is the misspelled version of the token *"hora"*.
- **Sentencing errors**: when the automatic split of sentences is wrongfully done, this problem kind motivates the edition of the sentence, eventually discarding portions of the sentence, keeping the correct portion of the sentence. An example of such problem happens in the sentence *"PILOTO APTO. O piloto Arno Ratzemberger foi aprovado após o teste de aptidão."* - this sentence was edited to consider only *"O piloto Arno Ratzemberger foi aprovado após o teste de aptidão."*.
- **Tokenization errors**: when the automatic tokenization introduces errors, this problem kind motivates the edition of the sentence, refactoring the tokenization. An example of such problem happens in the sentence "*Eu consigo chegar em 40 minutos com a lotação.*", where the word *"consigo"* was wrongfully tokenized into tehe tokens *"com"* (ADP) and *"si"* (PRON) - this sentence was refactored, reassembling the two tokens back to *"consigo"* (VERB). This kind of problem was also encountered for the word *"nos"* that sometimes was wrongfully tokenized into *"em"* (ADP) and *"os"* (DET), instead of *"nos"* (PRON), besides some tokens of compound words that also had to be reunited, for example, the token *"ex-presidente"* (NOUN) that was mapped to two tokens *"ex"* (NOUN) and *"presidente"* (NOUN), which is not our standard to deal with compound words.

5.1.2 Manual Identification of Definitive PoS tag

If no structural error is identified, each annotator chooses to keep the automatically assigned PoS tag or to change it, observing the whole sentence. We computed the agreement measure among the annotators to each token individually and to the whole sentence. Using the agreement measure to prioritize the decisions of the chief linguist (adjudication), we conducted the revision of all cases in the corpus. More details about this process can be found in a previous publication [5].

[8]The full list of the 17 UD PoS tags is available at: `https://universaldependencies.org/u/pos/`.

5.2 Instantiation of the PoS tag Manual Annotation

The manual annotation is naturally limited by the amount of text that can be revised. We spent four months working with 10 trained annotators and a linguist to manually revise the automatic annotation. During this time, a total of 47 annotation packages were produced following criteria as number of sentences, number of tokens, specific words, *etc.* These packages submit to manual annotation a total of 8,800 sentences and, as described in Section 5.1.1, structural problems were identified.

Tables 4 and 5 presents each of the 47 packages size in terms of number of sentences and tokens, indicating how many sentences (and tokens) were kept after discarding sentences that were considered with typos and pre-processing errors. This table also indicate the number of trimmed sentences in each package and the number of tokenization adjustments, as those adjustments also affect the number of tokens in each package. The last two columns indicate the number of tokens that had the PoS tag changed according to the manual annotation.

Table 4: Statistics for the manual annotation for the subset of the Folha Kaggle (first 24 packs).

pack.	sentences			tokens			trim.	rep.	corrected PoS tag	
name	orig.	kept	%	orig.	kept	%	sent.	tok.	tokens	%
p01	500	481	96.2%	7,144	6,857	96.0%	0	5	204	3.0%
p02	500	492	98.4%	7,029	6,919	98.4%	0	8	118	1.7%
p03	500	482	96.4%	7,056	6,787	96.2%	1	10	209	3.1%
p04	500	480	96.0%	6,865	6,553	95.5%	0	6	260	4.0%
p05	250	242	96.8%	7,348	7,104	96.7%	2	2	235	3.3%
p06	335	332	99.1%	4,889	4,856	99.7%	1	5	453	9.3%
p07	285	282	98.9%	3,395	3,347	99.6%	1	16	413	12.3%
p08	140	138	98.6%	3,471	3,415	98.4%	0	0	106	3.1%
p09	140	134	95.7%	3,474	3,329	95.8%	0	0	113	3.4%
p10	125	119	95.2%	3,531	3,344	94.7%	0	0	78	2.3%
p11	125	122	97.6%	3,432	3,349	97.6%	0	3	108	3.2%
p12	125	112	89.6%	3,505	3,163	90.3%	0	4	126	4.0%
p13	125	120	96.0%	3,451	3,313	96.0%	0	1	105	3.2%
p14	125	122	97.6%	3,580	3,472	97.0%	0	0	102	2.9%
p15	125	119	95.2%	3,645	3,471	95.3%	0	7	85	2.4%
p16	125	116	92.8%	3,523	3,230	91.7%	0	2	113	3.5%
p17	125	117	93.6%	3,565	3,301	92.6%	0	1	109	3.3%
p18	125	118	94.4%	3,458	3,265	94.4%	0	3	102	3.1%
p19	125	113	90.4%	3,579	3,255	90.9%	0	0	106	3.3%
p20	125	118	94.4%	3,489	3,288	94.2%	0	1	108	3.3%
p21	125	118	94.4%	3,544	3,317	93.7%	2	3	111	3.3%
p22	125	120	96.0%	3,468	3,315	95.5%	0	2	96	2.9%
p23	200	183	91.5%	5,771	5,225	90.5%	0	2	227	4.3%
p24	200	186	93.0%	5,493	5,077	92.4%	0	1	170	3.3%

Observing the results from Tables 4 and 5, the first observation is that it is an expensive process, as it took 4 months to a large team to deal with 8,800 sentences and to produce 8,420 corrected sentences (168,397 tokens). At the same time, the corrections sum up to a total of 380 discarded sentences, 9 trimmed sentences, 156 refactored tokenizations, and 6,815 token's PoS tag corrections. However, that

Table 5: Statistics for the manual annotation for the subset of the Folha Kaggle (last 23 packs).

pack.	sentences			tokens			trim.	rep.	corrected PoS tag	
name	orig.	kept	%	orig.	kept	%	sent.	tok.	tokens	%
p25	125	110	88.0%	3,477	3,021	86.9%	0	0	134	4.4%
p26	125	120	96.0%	3,387	3,251	96.1%	0	3	137	4.2%
p27	125	106	84.8%	3,589	3,017	84.1%	0	1	181	6.0%
p28	175	173	98.9%	2,985	2,950	98.8%	0	0	107	3.6%
p29	175	169	96.6%	2,943	2,838	96.4%	0	1	121	4.3%
p30	200	190	95.0%	3,427	3,249	94.8%	0	2	146	4.5%
p31	200	193	96.5%	3,398	3,276	96.4%	0	4	161	4.9%
p32	200	183	91.5%	3,404	3,109	91.3%	0	0	139	4.5%
p33	200	192	96.0%	3,399	3,259	95.9%	0	2	143	4.4%
p34	200	196	98.0%	3,432	3,360	97.9%	1	0	137	4.1%
p35	175	170	97.1%	3,512	3,348	95.3%	0	21	186	5.6%
p36	175	160	91.4%	3,550	3,293	92.8%	0	15	190	5.8%
p37	175	164	93.7%	3,530	3,211	91.0%	0	17	154	4.8%
p38	100	99	99.0%	1,532	1,512	98.7%	0	1	49	3.2%
p39	100	98	98.0%	1,559	1,532	98.3%	0	0	52	3.4%
p40	100	98	98.0%	1,481	1,456	98.3%	0	0	53	3.6%
p41	225	217	96.4%	4,422	4,265	96.4%	0	2	185	4.3%
p42	225	221	98.2%	4,505	4,414	98.0%	1	1	196	4.4%
p43	225	215	95.6%	4,508	4,317	95.8%	0	0	167	3.9%
p44	100	98	98.0%	1,719	1,687	98.1%	0	1	79	4.7%
p45	100	98	98.0%	1,734	1,698	97.9%	0	1	78	4.6%
p46	100	95	95.0%	1,720	1,634	95.0%	0	0	84	5.1%
p47	100	89	89.0%	2,461	2,150	87.3%	0	2	83	3.8%
total	8,800	8,420	95.7%	177,377	168,399	94.9%	9	156	6,817	4.0%

effort was necessary to achieve the correction of 4% of the tokens. It is important to notice that the automatic adjustments brought in average 1.6% of corrected tokens over an already good output of UDPipe. Therefore, to manually pinpoint the necessary corrections was a very difficult, but essential, task to be performed.

This task was the more demanding one, both in terms of time and people involvement. It started with ground definitions and the production of an annotation manual, then the involvement of annotators to perform the revision of PoS tags. This task took about three months and a half. During all this time, it involved one computer scientist and one linguist, plus ten annotators in the first month, then six in the second, and finally four in the last month.

6 Semi-Automatic Lemma Annotation

The fourth step of the proposed process is to perform the semi-automatic revision of lemma automatic annotation produced in the second step and taking into account the PoS tag adjustments made at the third step. The next sections detail the proposed principles for this step (Sec. 6.1) and the instantiation in our practical example (Sec. 6.2).

6.1 Principles of the Semi-Automatic Lemma Annotation

Having the sentences free of structural problems and each token with the correct PoS tag assigned, we proceed to the determination of the lemma of each token. We performed an automatic verification of each token lemma according to a previously existing lexical database (UNITEX-PB [11])[9]. The exception to these are tokens annotated with PoS tags PROPN, PUNCT, SYM, and X, as for those the token is automatically repeated as lemma. This automatic verification results in three possible outcomes:

- The token, and the PoS tag, may result in a single non-ambiguous option of lemma, and in that case the lemma is incorporated in the token annotation, regardless of the assignment made by UDPipe annotation;
- The token, and the PoS tag, may result in more than one option for lemma, and in that case the lemma options are submitted to a manual disambiguation to be made by linguists;
- The token, and the PoS tag, may be absent of the lexical database, and in that case the UDPipe assigned lemma is submitted to a manual analysis and decision to be made by linguists.

6.2 Instantiation of the Semi-Automatic Lemma Annotation

The resulting corpus of the experiment kept 8,420 sentences with each of its 168,399 tokens manually revised and the correct PoS tags, and all the sentences were stored in a single CoNLL-U data set. The next task is to verify the lemma assigned by the parser to each token. To that end, we performed a semi-automatic approach, starting with a consult to the previously existent lexical resource (UNITEX-PB) that, given a token and a PoS tag, delivers the possible lemmas.

At this point, three possible outcomes of the lexical consult were considered:

- **Defined**: If the lexical consult delivers a single lemma, this lemma is adopted as the token lemma, regardless of the lemma assigned by UDPipe:

  e.g., the token "*casas*" as NOUN has only[10] the lemma "*casa*";

- **Ambiguous**: If more than one lemma is delivered by the lexical consult, these lemma options were submitted to human revision to disambiguate among the choices available:

  e.g., the token "*foi*" as VERB has either the lemma "*ir*" or "*ser*", and the human revision needs to choose either of the lemmas to each occurrence of the token "*foi*";

- **Unknown**: If the token was absent of the lexical resource for the assigned PoS tag, the token was submitted to human revision to decide which would be the appropriate lemma to assign:

  e.g., the token "*desracializa*" as VERB is not found on the lexical resource and needs to be analyzed by humans to receive the lemma "*desracializar*".

Table 6 presents the overall numbers in each of these outcomes for all tokens in the resulting corpus.

The results in Table 6 demonstrate that most of the tokens could have their lemmas verified (**Defined**), as almost 93% of the tokens had the token lemma unambiguously defined by the lexical resource. Additionally, among the 8,665 tokens not found in the lexical resource (**Unknown**), the manual revision process analyzed only 3,794 terms, since many of those tokens had multiple occurrences. The 3,295 tokens with two or more options of lemma (**Ambiguous**) also had to be analyzed manually,

[9]The UNITEX-PB lexical resource provides a list of 900,624 distinct words and 9,072,339 entries, stating in each entry a word and a single set of PoS tag, lemma, and morphological features. However, UNITEX-PB uses a traditional Portuguese language grammatical paradigm that is slightly different from UD definitions for PoS tag and morphological features.

[10]Note that the token "*casa*" as VERB would delivers a different lemma ("*casar*").

Table 6: Statistics for the semi-automatic lemma annotation for the resulting corpus.

total tokens	**Defined**		**Ambiguous**		**Unknown**	
	tokens	%	tokens	%	tokens	%
168,399	156,439	92.90%	3,295	1.96%	8,665	5.14%

but unlike the ones not found, they had to be analyzed for each occurrence, since the same token may have different lemmas assigned to different occurrences. For example, the token "*foi*" with PoS tag AUX was manually assigned with lemma "*ser*" 546 times and with "*ir*" 35 times.

This task was performed by one computer scientist and two linguist plus three annotators for a little more than one month.

7 Semi-Automatic Morphologic Features Annotation

The fifth step of the proposed process is to perform the semi-automatic revision of morphological features that were automatically produced in the second step, taking into account both the PoS tag adjustments made at the third step and the lemma adjustments made at the fourth step. The next sections detail the proposed principles for this step (Sec. 7.1) and the instantiation in our practical example (Sec. 7.2).

7.1 Principles of the Semi-Automatic Morphologic Features Annotation

After having all tokens with lemmas corrected, we proceed to a similar process to verify the morphological features[11] using the same lexical database (UNITEX-PB). The exceptions for these are the tokens with PoS tags ADP, ADV, CCONJ, INTJ, PUNCT, SCONJ, SYM, and X that have no morphological features. For the tokens of these PoS tags, all morphological features are automatically removed. Other exceptions are the tokens with PoS tag PROPN, which are dealt as a particular case. Given that, this second automatic verification results in three possible outcomes:

- The token, the PoS tag, and the lemma may result in a single non-ambiguous option of morphological features, and in that case the determined features are assigned as the token morphological annotation;
- The token, the PoS tag, and the lemma may be absent of the lexical database, and in that case the morphological features assigned by UDPipe are submitted to a manual analysis and decision to be made by linguists. The only exception to that are the tokens tagged as PROPN that can be found in the lexical database as NOUN or ADJ, in which case the token receives the morphological features associated with the corresponding NOUN or ADJ;
- The token, the PoS tag, and the lemma may result in more than one option for morphological features, and in that case we attempt a series of heuristics to reduce the volume of tokens with the feature options to be submitted to manual disambiguation by linguists. The heuristics applied concern some deteminers (DET), pronouns (PRON), and verbs, being auxiliaries (AUX) or not (VERB). If the heuristics cannot be applied, the token morphological feature options are sent to be disambiguated by linguists.

The heuristics applied to determiners consider that, if the token PoS tag is DET, and the token is one of the 8 possible articles in Portuguese ("*a*", "*as*", "*o*", "*os*", "*um*", "*uma*", "*umas*", "*uns*"), we automatically assign the morphological feature *PronType* with the value *Art*, plus the features *Definite*, *Gender*, and *Number* accordingly to the token.

[11] The full list of the UD morphological features is available at `https://universaldependencies.org/u/feat/`.

The heuristics for pronouns consider that, if the token PoS tag is PRON, we disambiguate using specific rules for the tokens:

- "*a*", "*as*", "*o*" and "*os*" followed by the tokens "*de*" (ADP), "*qual*" (PRON), "*quais*" (PRON), or "*que*" (PRON), that are always considered demonstrative pronouns, receive the morphological features *PronType* with the value *Dem*, *Person* with the value *3*, and the features *Gender* and *Number* accordingly to the token;

- "*a*", "*as*", "*o*" and "*os*" preceeded by the token "*que*" (PRON), that are always considered personal pronouns, receive the morphological features *PronType* with the value *Prs*, *Case* with the value *Acc*, *Person* with the value *3*, and the features *Gender* and *Number* accordingly to the token;

- "*qual*" and "*quais*" preceded by the tokens "*a*" (PRON), "*as*" (PRON), "*o*" (PRON) or "*os*" (PRON) that are always considered relative pronouns, receive the morphological features *PronType* with the value *Rel*, *Gender* with the value *Neut*, and *Number* accordingly to the token.

The heuristics for verbs, being auxiliary or not, consider that, if the token PoS tag is VERB or AUX, we disambiguate using the morphological features assigned by UDPipe, if the UDPipe choice is among the lexical database options; otherwise, the token morphological feature options are sent to be disambiguated by linguists.

After all that, the result is the annotated text with correct sentences, duly tokenized, all tokens with verified correct PoS tag, lemma, and morphological features. In terms of CoNLL-U representation of the annotated text, the result is fully verified, except for the dependency relations, which can futurely be better verified based on reliable token, lemma, PoS tag, and morphological information.

7.2 Experiment for Semi-Automatic Morphologic Annotation

Similarly to what was done for the lemma annotation, we start from the reliable triplet token, PoS tag, and lemma and search for the morphological information associated to the triplet in the UNITEX-PB lexical resource. Once again three possible outcomes of the lexical consult were considered, but, in the ambiguous case, we tried heuristics to resolve the ambiguity without human disambiguation for some cases. Therefore, the outcomes were split in the following situations:

- **Defined**: If the lexical consult delivers a single morphological features option, this morphological features option is adopted for the token, regardless of the ones assigned by UDPipe:

  - *e.g.*, the token "*casas*" as NOUN, with lemma "*casa*", delivers the single morphological features: "Gender=Fem|Number=Plur";

- **Heuristic**: If more than one morphological feature option is delivered by the lexical consult, but one of our heuristics is capable to automatically disambiguate:

  - *e.g.*, the token "*diga*" as VERB, with lemma "*dizer*", delivers either the morphological features:

  "Mood=Sub|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin"; or

  "Mood=Imp|Number=Sing|Person=3|VerbForm=Fin";

  that is automatically disambiguated to the first set of morphological features for all five occurrences in our experiment, as in all occurrences this was the choice made by UDPipe annotation;

- **Ambiguous**: If more than one morphological feature is delivered by the lexical consult and the heuristics were not enough to fully disambiguate the morphological features, the token is submitted to human revision to disambiguate the remaining ambiguous morphological features;

- *e.g.*, the token "*tais*" as PRON, with lemma "*tal*", delivers either the morphological features:

"Gender=Neut|Number=Plur|PronType=Ind"; or

"Gender=Neut|Number=Plur|PronType=Dem";

that cannot be automatically disambiguated, and has to be sent to human disambiguation;

- **Unknown**: If the token was absent of the lexical resource for the assigned PoS tag, the token is submitted to human revision to decide which would be the appropriated morphological features to assign:

  - *e.g.*, the token "*fintech*" as NOUN, with lemma "*fintech*", that is not in the lexical resource, and has to be sent to human disambiguation and have assigned the morphological features : "Gender=Fem|Number=Sing".

Table 7 presents the overall numbers in each of these outcomes for all tokens in the resulting corpus.

Table 7: Statistics for the semi-automatic morphological features annotation for the resulting corpus.

total	**Defined**		**Heuristic**		**Ambiguous**		**Unknown**	
tokens	tokens	%	tokens	%	tokens	%	tokens	%
168,399	122,183	72.56%	37,212	22.10%	7,923	4.70%	1,076	0.64%

Observing the results in Table 7, we notice that most of the tokens (73%) had just one option of morphological features (**Defined**). Considering the 22% additional tokens defined with heuristics (**Heuristic**), we avoided human revision for a total of 95% of the tokens. Therefore, only the remaining 5% (**Ambiguous** and **Unknown**) tokens needed to be reviewed by linguists.

This task was performed by one computer scientist and two linguist plus three annotators for a little more than one month.

8 Overall Analysis

Summarizing the improvement achieved by all corrections performed, we state in Table 8 the number of tokens corrected at each step of our experiments for the verified corpus.

The numbers in Table 8 concern only the retained 8,420 sentences at the end of the process to produce a verified annotated corpus from the Folha Kagle corpus. Therefore, the table's first row (row 1) indicates that the resulting 8,420 sentences were outputted with only 135,469 tokens already corrected, which is 86.5% of the total 168,397 tokens obtained at the end of all improvements. The second row (row 2a) indicates the passage of the pure text sentences through UDPipe. During this passage, which performed the tokenization of the contracted words (see Section 3.1.2), a large number of additional correct token was produced, increasing the number of correct tokens to 168,241. The remaining 156 incorrect tokens were identified and corrected during the linguist adjudication (see Section 5.1.1 about dealing with sentences' structural problems).

Observing the numbers for corrected PoS tags on Table 8, it is noticeable the improvements of 6,601 PoS tags with the automatic corrections: from 155,004 (row 2a) to 161,605 (row 2d) correct PoS tags (4%). Subsequently the correction of the 6,792 PoS tags in the manual annotation (row 3) corresponds to obtain all tokens with correct PoS tags (additional 4%).

The improvements in terms of correction of lemmas happened when the PoS tags were corrected both automatically and manually, increasing from 156,994 (row 2a) to 161,990 (row 3) correct lemmas (3%), followed by the corrections by lexical consult (row 4a) and human disambiguation (row 4b) to correct the remaining 6,407 lemmas (3.8%).

Table 8: Improvements achieved at each step for the produced verified corpus in number and percentage of correct tokens.

improvements		tokens		PoS tags		lemmas		morp. features	
Pure Text Preparation									
1	Preprocessing	135,469	86.5%	-	-	-	-	-	-
Automatic Annotation									
2a	UDPipe Annotation	168,241	99.9%	155,004	92.0%	156,994	93.2%	139,858	83.0%
	Automatic Corrections								
2b	Functional Words	168,241	99.9%	156,995	93.2%	158,932	94.4%	139,858	83.0%
2c	Usual Mistakes	168,241	99.9%	160,196	95.1%	159,343	94.6%	139,867	83.1%
2d	Usual Sequences	168,241	99.9%	161,605	96.0%	160,229	95.1%	139,905	83.1%
PoS tag Manual Annotation									
3	Manual Revision	168,399	100.0%	168,399	100.0%	161,990	96.2%	140,311	83.3%
Semi-Automatic Lemma Annotation									
4a	Lexical Consultations	-	-	-	-	162,467	96.5%	159,911	95.0%
4b	Human Disambiguation	-	-	-	-	168,399	100.0%	166,594	98.9%
Semi-Automatic Morphologic Features Annotation									
5a	Lexical Consultations	-	-	-	-	-	-	167,051	99.2%
5b	Human Disambiguation	-	-	-	-	-	-	168,399	100.0%

Similarly, the correction of morphological features was impacted in small measure by PoS tag corrections, from 83% (row 2a) to 83.3% (row 3), but greatly improved (15.6%) by lemma corrections, going from 83.3% (row 3) to 98.9% (row 4b). The remaining errors (1.1%) were corrected by the lexical consult (row 5a) and human disambiguation (row 5b).

In terms of the whole project timeline, Figure 3 describes the layout of the tasks (as described in Figure 1) in terms of the necessary time that each task required, and how many people were involved. The more important point to be observed is that each step was practically executed sequentially with very little overlap. This is the more innovative aspect of our approach, and as seen in the results of Table 8, the execution of the tasks of each step provided a solid basis for the next step. A secondary observation is that the bulk of demands was done in the PoS tag annotation step, both in terms of time taken, as well as the number of people involved. Finally, it is worthy mentioning that the automatic steps were very effective, delivering a good accuracy, and they can be easily replicated to large amounts of text.

9 Final Remarks

This technical report defines a process to build revised corpora and delivers a corpus in Portuguese annotated with morphological information using UD format and tagset. The process can be reproduced by other similar initiatives in Portuguese or in other languages, requiring the natural adaptation of language specific information (list of closed class words, expressions, lexical resource, etc.). The produced corpus with 8,420 sentences (168,399 tokens) can be effective for several training tasks or benchmarks.

Future work includes the annotation of syntactic information (columns HEAD and DEPREL of the CoNLL-U file, as described in Figure 2), following the same modus operandi. The annotation of other sentences of the Folha Kaggle repository must significantly increase the size of the desired treebank in UD for Portuguese.

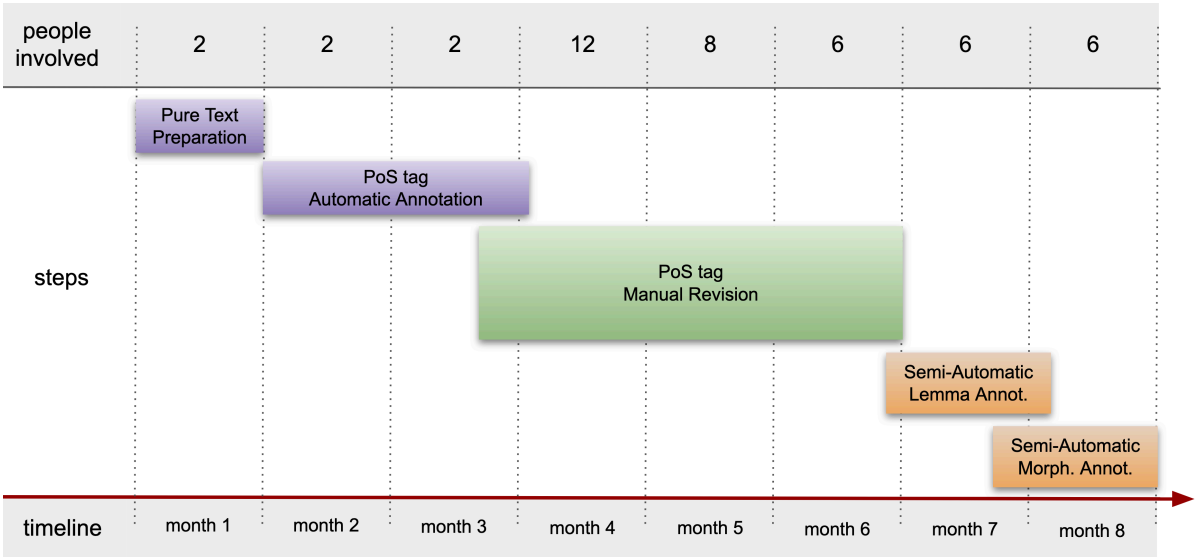

Figure 3: Timeline of the practical experiment and number of people involved in each step.

Acknowledgments

The authors are grateful to the Center for Artificial Intelligence (C4AI) of the University of São Paulo, supported by IBM and FAPESP (grant #2019/07665-4).

References

[1] Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4):555–596, dec 2008.

[2] Delphine Bernhard, Anne-Laure Ligozat, Myriam Bras, Fanny Martin, Marianne Vergez-Couret, Pascale Erhart, Jean Sibille, Amalia Todirascu, Philippe Boula de Mareüil, and Dominique Huck. Collecting and annotating corpora for three under-resourced languages of france: Methodological issues. *Language Documentation & Conservation*, 15:42, 2021.

[3] Flavio Massimiliano Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. Udante: First steps towards the universal dependencies treebank of dante's latin works. In Johanna Monti, Felice dell'Orletta, and Fabio Tamburini, editors, *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020*, volume 2769 of *CEUR Workshop Proceedings*, Bologna, Italy, 2020. CEUR-WS.org.

[4] Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. Building Universal Dependency treebanks in Korean. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[5] Magali Sanches Duran, Lucelene Lopes, Thiago Alexandre Salgueiro Pardo, and Maria das Graças Volpe Nunes. Sobre anotação de corpus e anotadores humanos: questões científicas e humanas. *Linguamática*, 2021. (submitted).

[6] Dan Gillick. Sentence boundary detection and the problem with the U.S. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[7] Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. When collaborative treebank curation meets graph grammars. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5291–5300, Marseille, France, May 2020. European Language Resources Association.

[8] Lucelene Lopes, Magali Sanches Duran, and Thiago Alexandre Salgueiro Pardo. Universal dependencies-based pos tagging refinement through linguistic resources. In *Proceedings of the 10th Brazilian Conference on Intelligent System*, BRACIS'21, 2021.

[9] Favaro Manuel, Biffi Marco, and Montemagni Simonetta. Risorse linguistiche di varietà storiche di italiano: il progetto travasi. In *Proceedings of 7th Italian Conference on Computational Linguistics (CLiC-it)*, page 9, Bologna, Italy (Online), March 2021. CEUR-WS.

[10] Aleksandra Miletic, Myriam Bras, Marianne Vergez-Couret, Louise Esher, Clamença Poujade, and Jean Sibille. Building a Universal Dependencies treebank for Occitan. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2932–2939, Marseille, France, May 2020. European Language Resources Association.

[11] Marcelo Caetano Martins Muniz. A construção de recursos linguístico-computacionais para o português do brasil: o projeto unitex-pb. Master's thesis, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo - ICMC/USP, 2004.

[12] Joakim Nivre. Towards a universal grammar for natural language processing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 3–16, Cham, 2015. Springer International Publishing.

[13] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020. European Language Resources Association.

[14] Thiago Pardo, Magali Duran, Lucelene Lopes, Ariani Felippo, Norton Roman, and Maria Nunes. Porttinari - a large multi-genre treebank for brazilian portuguese. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 1–10, Porto Alegre, RS, Brasil, 2021. SBC.

[15] Mohammad Sadegh Rasooli, Pegah Safari, Amirsaeid Moloodi, and Alireza Nourian. The persian dependency treebank made universal, 2020.

[16] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

[17] Marlesson Santana. Kaggle - news of the brazilian newspaper. `https://www.kaggle.com/marlesson/news-of-the-site-folhauol`. Accessed: 2021-06-14.

[18] Djamé Seddah, Farah Essaidi, Amal Fethi, Matthieu Futeral, Benjamin Muller, Pedro Javier Ortiz Suárez, Benoît Sagot, and Abhishek Srivastava. Building a user-generated content North-African

Arabizi treebank: Tackling hell. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1139–1150, Online, July 2020. Association for Computational Linguistics.

[19] Milan Straka. 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, 2018.

[20] Milan Straka, Jan Hajič, and Jana Straková. 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, May 2016. European Language Resources Association (ELRA).

[21] Umut Sulubacak. Implementing universal dependency, morphology, and multiword expression annotation standards for turkish language processing. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(3):1662–1672, 2018.

[22] Utku Türk, Kaan Bayar, Ayşegül Dilara Özercan, Görkem Yiğit Öztürk, and Şaziye Betül Özateş. First steps towards Universal Dependencies for Laz. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 189–194, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics.

[23] Universal Dependencies. UD Portuguese Bosque - UD version 2. `https://universaldependencies.org/treebanks/pt\_bosque/index.html`. Accessed: 2021-06-14.

[24] Alina Wróblewska. Towards the conversion of National Corpus of Polish to Universal Dependencies. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5308–5315, Marseille, France, May 2020. European Language Resources Association.