Verifica-UD: a Verifier for Universal Dependencies Annotation for Portuguese

Lucelene Lopes[1], **Magali Sanches Duran**[1], **Thiago Alexandre Salgueiro Pardo**[1]

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

{lucelene,magali.duran}@gmail.com,

taspardo@icmc.usp.br

Abstract. *This paper presents Verifica-UD, a web-based tool to detect problems in Portuguese sentences annotated using Universal Dependencies (UD) standards in the form of a CoNLL-U file. The tool performs three levels of sentence verification: structural (to assess CoNLL-U compliance), morphosyntactic (to assess the part of speech tagging), and syntactic (to assess the parsing information). Verifica-UD also provides detailed help on Portuguese UD annotation directives. The benefits of this tool for reviewing annotated corpora are illustrated with an experiment.*

1. Introduction

The use of Universal Dependencies (UD) [de Marneffe et al. 2021, Nivre et al. 2020] as coding format for annotated corpora brings several advantages in terms of standardization among languages and clarity of concepts for users regardless of language.

CoNLL-U is the standard file format for the annotation of dependency treebanks in UD. Because of that it has been a format of interest for several annotators, and consequently also tool developers. This is the case of web-based visualization tools as Arborator-Grew [Guibon et al. 2020], but also some editors and searchers as UDeasy [Villa 2022] and UDConcord [Miranda and Pardo 2022].

Despite being a compact and relatively simple format, CoNLL-U may be somewhat confusing to be directly handled by human annotators, and often mistakes produced while editing a CoNLL-U file are hard to be spotted using a pure (non-rich) text editor. Some editors provide visualization tools with color tagging of CoNLL-U files, as the VSCode-conllu extension [Grobol 2021], but even such help is not enough when dealing with a large file. Besides the issues of data encoding, UD annotation may be a challenging task, as corpus annotation has shown in recent developments in the area. More than the directives of the original UD project, languages may need additional guidelines to deal with their specific linguistic phenomena.

The UD project offers a validation tool that has general restrictions and language dependent particularities. It is important to highlight that the language particularities included in the UD validation tool tends to include only definitions made based on the available corpora among the UD datasets, and not necessarily following general integrated language directives.

Given this panorama, we propose in this paper a web-based tool to verify CoNLL-U files following the general directives for annotation of Portuguese in UD stated in [Duran 2021, Duran 2022, Lopes et al. 2023a]. This tool, called Verifica-UD, performs the verification of the CoNLL-U file in three levels: structural, Part of Speech (PoS) tagging, and parsing. In fact, the rules applied in our tool reflect the more recent directives considered in the recent discussions of Portuguese annotation in UD and recent resources.

Next section describes the three levels of verification performed by Verifica-UD. The third section briefly presents the online tool usage. The fourth section presents a case study illustrating the potential of the tool to aid in human annotation. Finally, the conclusion section highlights the paper contributions and suggests future work.

2. Verification

This section describes the three levels of verification performed by Verifica-UD:

- Structural verification: analysis of the compliance with the CoNLL-U standards, plus the correct format of the dependency tree (e.g., a single root, connection of all nodes);
- Morphosyntactic verification: analysis of the fields lemma, PoS tag, and morphological features with respect to a lexical resource and general tagging rules;
- Dependency relations verification: analysis of the fields head and dependency relation (DEPREL) with respect to valid connections between head and dependents, as well as the corresponding tagging information of these tokens.

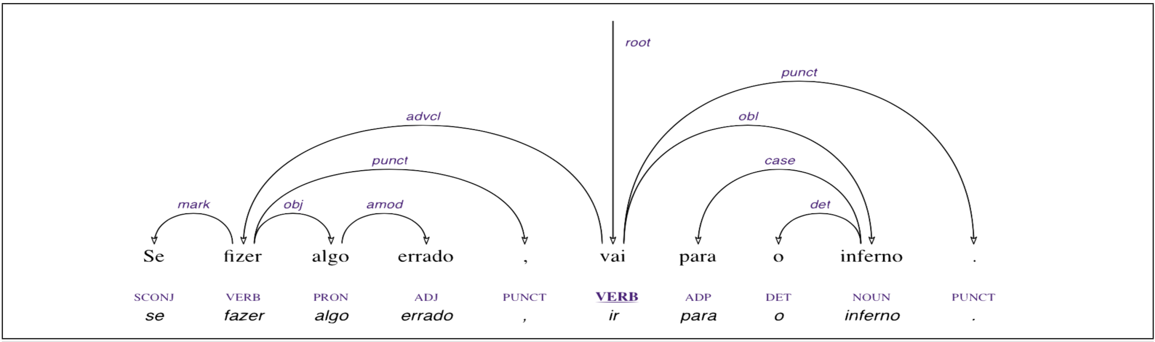
2.1. Structural Verification

The structural verification starts with tests of compliance with the CoNLL-U format, followed by verification of the dependency tree integrity. As such, the structural verification rules are independent of the language, which is not the case of the morphosyntatic and dependency relation rules that are tied to the Portuguese standards. The CoNLL-U format defines that each sentence requires two initial pieces of information:

- the sentence identifier, as `# sent_id =` <string>
- the textual sentence content, as `# text =` <string>

After that, all tokens must be placed in individual lines containing one numbered token per line, each with ten fields separated by tab characters. Contracted words should be split into individual tokens. In the CoNLL-U format, the contracted words must have an individual line to hold it preceding the lines of the split tokens. The ten fields of each token indicate, respectively: ID - the token number identifier; FORM - the token form; LEMMA - the token lemmatized form; POS - PoS tag; XPOS - language specific PoS tag (ignored by Verifica-UD); FEAT - the morphological features; HEAD - the identifier of the token head of the dependency relation; DEPREL - the dependency relation tag; DEPS - enhanced dependency graph (ignored by Verifica-UD); MISC - miscellaneous information. Figure 1 shows an example of a correct sentence[1] with both the CoNLL-U annotation and the corresponding dependency tree.

[1]"Se fizer algo errado, vai para o inferno" (If you do something wrong, you'll go to hell). This sentence may have another interpretation in Portuguese: "Se fizer errado algo" (If you do it wrong); in this case, "errado" (wrong) would be annotated as ADV advmod because it would be modifying the predicate itself and not the object.


root
punct
advcl
obl
punct
case
mark
obj
amod
det
Se fizer algo errado , vai para o inferno .
SCONJ VERB PRON ADJ PUNCT VERB ADP DET NOUN PUNCT
se fazer algo errado , ir para o inferno .

```
# sent_id = FOLHA_DOC009367_SENT031
# text = Se fizer algo errado, vai pro inferno.
1   Se      se      SCONJ   _   _   2   mark    _   _
2   fizer   fazer   VERB    _   Mood=Sub|Number=Sing|Person=3|Tense=Fut|VerbForm=Fin    6   advcl   _   _
3   algo    algo    PRON    _   Gender=Masc|Number=Sing|PronType=Ind    2   obj _   _
4   errado  errado  ADJ _   Gender=Masc|Number=Sing|VerbForm=Part   3   amod    _   SpaceAfter=No
5   ,   ,   PUNCT   _   _   2   punct   _   _
6   vai ir  VERB    _   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   0   root    _   _
7-8 pro _   _   _   _   _   _   _   _
7   para    para    ADP _   _   9   case    _   _
8   o   o   DET _   Definite=Def|Gender=Masc|Number=Sing|PronType=Art   9   det _   _
9   inferno inferno NOUN    _   Gender=Masc|Number=Sing 6   obl _   SpaceAfter=No
10  .   .   PUNCT   _   _   6   punct   _   _
```

Figure 1. CoNLL-U example of a sentence and corresponding dependency tree.

The structural verification detects 16 different error kinds. The full list of structural errors detected is available at a previous publication [Lopes et al. 2023b]. Figure 2 illustrates three error examples. The first one is a common error where the token #7 has no morphological features. This token is denoted without the sixth field with the empty indicator ("_") missing. Another error in this figure is in the last token that is incorrectly numbered with #11, instead of the expected #10. The third error in Figure 2 is the dependency tree malformation, since it has two **root** tokens (#2 and #6).

```
# sent_id = FOLHA_DOC009367_SENT031
# text = Se fizer algo errado, vai pro inferno.
1   Se      se      SCONJ   _   _   2   mark    _   _
2   fizer   fazer   VERB    _   Mood=Sub|Number=Sing|Person=3|Tense=Fut|VerbForm=Fin    0   root    _   _
3   algo    algo    PRON    _   Gender=Masc|Number=Sing|PronType=Ind    2   obj _   _
4   errado  errado  ADJ _   Gender=Masc|Number=Sing|VerbForm=Part   3   amod    _   SpaceAfter=No
5   ,   ,   PUNCT   _   _   2   punct   _   _
6   vai ir  VERB    _   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   0   root    _   _
7-8 pro _   _   _   _   _   _   _   _
7   para    para    ADP _       9   case    _   _
8   o   o   DET _   Definite=Def|Gender=Masc|Number=Sing|PronType=Art   9   det _   _
9   inferno inferno NOUN    _   Gender=Masc|Number=Sing 6   obl _   SpaceAfter=No
11  .   .   PUNCT   _   _   6   punct   _   _
```

Figure 2. Examples of structural problems, with (a) the sixth field missing error for token #7, (b) last token incorrectly numbered, and (c) malformed dependency tree with two root tokens (#2 and #6).

2.2. Morphosyntactic Verification

Morphosyntactic verification applies to tokens individually. Each token must have consistent information for the fields FORM, LEMMA, POS, and FEAT. In PoS tag annotation for Portuguese, we follow the directives defined at [Duran 2021], that establishes that, from the 17 original PoS tags, only 16 are employed (the PART tag is not used). The PoS tags ADP, AUX, CCONJ, DET, PRON, and SCONJ are closed classes and, as such, they have, in principle, all possible forms known. The PoS tags ADV and NUM have a defined closed subset, as all primitive adverbs (in Portuguese, adverbs not ending with "*-mente*" - similar to English "-ly") that form a closed subclass, and all numbers written in their extensive form (not with digits). Among the other (open) classes, NOUN, ADJ, VERB, and INTJ, plus the open subset of ADV, have an extensive representation and, together with the closed classes and closed subsets of NUM and ADV, are included in a lexical resource called PortiLexicon-UD [Lopes et al. 2022].

In this way, the morphosyntactic verification is done for each token tagged in the CoNLL-U with tags ADP, ADV (primitive adverbs subset), AUX, CCONJ, DET, NUM (written subset), PRON, and SCONJ, verifying its form against the lexical resource. If present in the lexical resource with the lemma and morphological annotated option, the token is considered correct. Otherwise, general annotation rules are verified, for example, a token tagged as PRON or DET must have a **PronType** feature. If the general annotation rules are valid, a warning is issued stating that the token belongs to a closed class, but it is not present in the lexical resource. However, if the general annotation rules are violated, the token is considered incorrect and an error is issued.

For tokens tagged in the CoNLL-U with tags ADJ, ADV (except primitive ones), INTJ, NOUN, and VERB, the token form is verified against the lexical resource. If present in the lexical resource, the annotation must be one of the options, otherwise a warning is issued. However, if absent in the lexical resource, the general annotation rules are verified and, if the rules are violated, an error is issued, otherwise, a warning is issued.

For tokens tagged in the CoNLL-U with tags PROPN, PUNCT, X, SYM, and NUM (with digits in the FORM field), general annotation rules are verified, and, if the annotation is not one of the expected possibilities, an error is issued.

In Figure 3, three example situations are indicated. An error is indicated for the lemma in token #1 because it mismatches the lexical entry for the SCONJ "*se*", since lemmas of common words must not be capitalized. Another error is found in token #3 that is a known PRON, but it is missing the morphological feature **PronType=Ind**, since all PRON tokens require the **PronType** feature. Token #9, the NOUN "*capetódromo*", is absent in the lexical resource, but, since NOUN is an open class, a warning is issued.

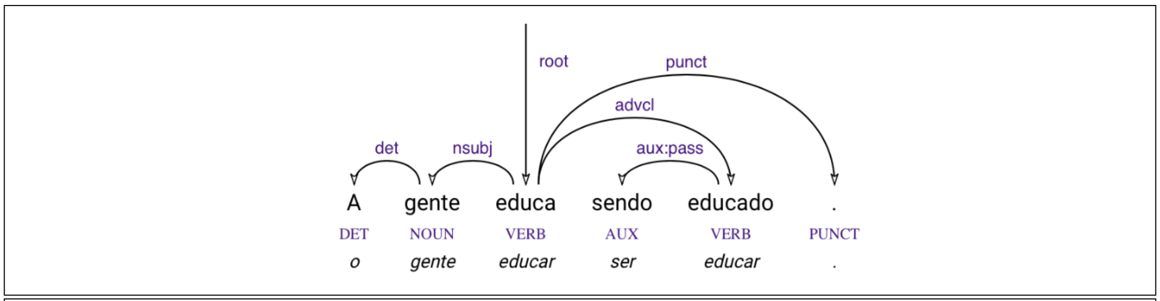
```
# sent_id = FOLHA_DOC009367_SENT031
# text = Se fizer algo errado, vai pro capetódromo.
1    Se   Se   SCONJ   _   _   2   mark   _   _
2    fizer   fazer   VERB   _   Mood=Sub|Number=Sing|Person=3|Tense=Fut|VerbForm=Fin   6   advcl   _   _
3    algo   algo   PRON   _   Gender=Masc|Number=Sing 2   obj _   _
4    errado  errado  ADJ _   Gender=Masc|Number=Sing|VerbForm=Part   3   amod   _   SpaceAfter=No
5    ,   ,   PUNCT   _   _   2   punct   _   _
6    vai ir   VERB   _   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   0   root   _   _
7-8  pro _   _   _   _   _   _   _   _   _
7    para   para   ADP _   _   9   case   _   _
8    o   o   DET _   Definite=Def|Gender=Masc|Number=Sing|PronType=Art   9   det _   _
9    capetódromo capetódromo NOUN   _   Gender=Masc|Number=Sing 6   obl _   SpaceAfter=No
10   .   .   PUNCT   _   _   6   punct   _   _
```

Figure 3. Examples of morphosyntactic problems, with (a) the wrong lemma for token #1 (it should be "*se*"), (b) the PronType=Ind is missing from FEATS of token #3, (c) token #9 "*capetódromo*" is absent as NOUN in the lexical resource (warning).

In total, the morphosyntactic verification may issue 29 possible errors and 14 possible warnings. The list of rules is available at a previous publication [Lopes et al. 2023b].

2.3. Dependency Relation Verification

Dependency relation verification applies to token relations, therefore, to token sequences belonging to the same branch of the dependency tree, and it concerns primarily the fields HEAD and DEPREL, but also their eventual relations with the fields FORM, LEMMA, POS, and FEAT. In CoNLL-U encoding, each DEPREL tag represents a dependency relation associating a dependent token, where the tag is, to a head token to which the token ID is in the field HEAD. For example, in Figure 4[2], token #2 "*gente*" is the dependent of the dependency relation **nsubj** that has the token #3 "*educa*" as the head of the relation.



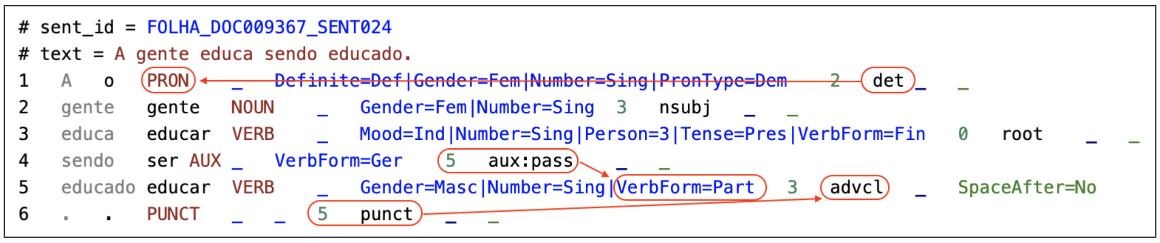
```
# sent_id = FOLHA_DOC009367_SENT024
# text = A gente educa sendo educado.
1   A     o     DET _    Definite=Def|Gender=Fem|Number=Sing|PronType=Art    2   det _   _
2   gente   gente   NOUN    _    Gender=Fem|Number=Sing  3   nsubj   _   _
3   educa   educar  VERB    _    Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   0   root    _   _
4   sendo   ser AUX _    VerbForm=Ger    5   aux:pass    _   _
5   educado educar  VERB    _    Gender=Masc|Number=Sing|VerbForm=Part|Voice=Pass    3   advcl   _   SpaceAfter=No
6   .   .   PUNCT   _   _   3   punct   _   _
```

Figure 4. CoNLL-U example of a sentence and corresponding dependency tree.

Because of this token relation aspect, the dependency relation rules are of different nature than the previous ones, often establishing restrictions to which kind of PoS tags can be dependent, or head of specific relations, or even establishing the need for some specific morphological feature to some dependency relations. Also at this dependency relation level, some situations are not necessarily errors, but they are unusual, and, as such, may provoke the indication of warnings. For example, the CoNLL-U representation in Figure 5 has three situations indicated. The first token provokes an error as Token #1 "*a*" has PoS tag PRON, but it is the dependent of a **det** relation, and one of the dependency relation rules states that all dependents of **det** relation need to be DET. Token #4 also indicates an error, as a token head of dependency relation **aux:pass** needs to be a VERB holding the morphological features **VerbForm=Part** and **Voice=Pass**. Finally, the last token (#6) provokes a warning as there is a dependency rule stating that final punctuation is usually dependent of a relation **punct** with the head being the sentence's **root**.

The dependency relation rules define 61 errors and 8 warnings. The full list of dependency rules is available at a previous publication [Lopes et al. 2023b].

[2]"A gente educa sendo educado" (We educate being educated or We educate being polite). Depending on the interpretation, "educado" may be a VERB in passive voice or an ADJ, and the AUX "sendo" (being) may be aux:pass ou cop, respectively.



```
# sent_id = FOLHA_DOC009367_SENT024
# text = A gente educa sendo educado.
1   A       o       PRON    _   Definite=Def|Gender=Fem|Number=Sing|PronType=Dem     2   det   _   _
2   gente   gente   NOUN    _   Gender=Fem|Number=Sing   3   nsubj   _   _
3   educa   educar  VERB    _   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   0   root   _   _
4   sendo   ser AUX _   VerbForm=Ger    5   aux:pass   _   _
5   educado educar  VERB    _   Gender=Masc|Number=Sing|VerbForm=Part   3   advcl   _   SpaceAfter=No
6   .   .   PUNCT   _   _   5   punct   _   _
```

Figure 5. Examples of syntactic problems, with (a) a relation det involving token #1 that has PoS tag PRON, (b) the token #4 is dependent of relation aux:pass that requires as head a token VERB with features VerbForm=Part and Voice=Pass, (c) token #6, the final punctuation, should have relation punct with the root.

3. Verifica-UD Online Tool

Verifica-UD implements the three level rules for errors and warnings as defined in the previous section through a webservice tool[3]. This tool was developed with Python, implementing the server using REST API [Richardson and Ruby 2007], while the web interface was implemented using a HTML/CSS/PHP technology. The basic tool operation consists in uploading a CoNLL-U file (.conllu) that is automatically verified. Once uploaded, the CoNLL-U file is analyzed and a list of errors and warnings grouped by sentences sorted in alphabetical order is displayed as depicted in Figure 6. After performing the verification and displaying the errors and warnings, the tool offers the possibility of exporting a report that lists the sentences, the tokens, and the errors or warnings that were found.

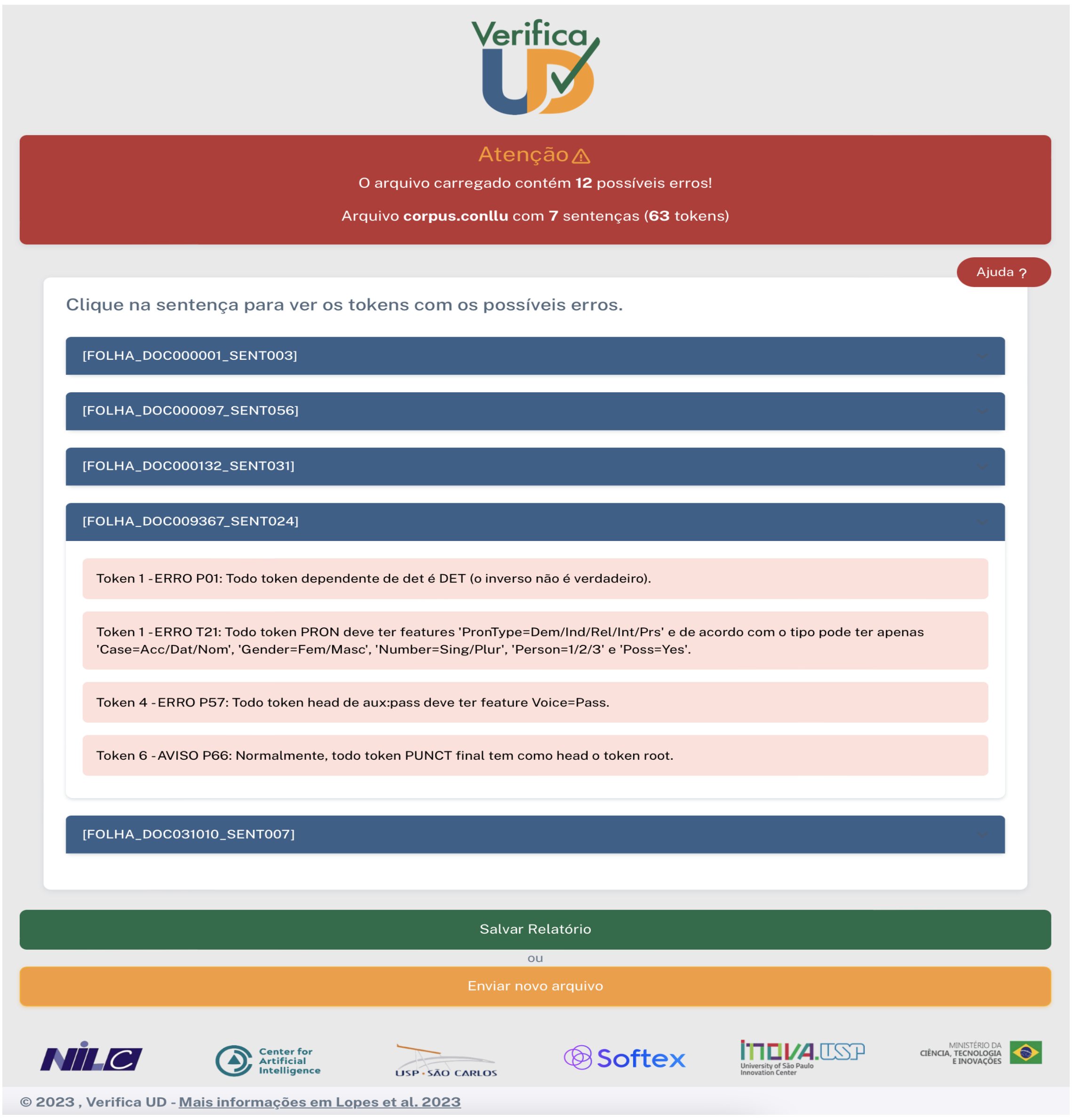
Another feature of the interface is a set of help pages that allows the user to learn more about CoNLL-U annotation for Portuguese according to the directives in [Duran 2021, Duran 2022, Lopes et al. 2023a]. At the help interface, it is also possible to see the full list of errors and warnings currently detected by Verifica-UD grouped by the structural, morphosyntatic, and dependency relation level, as shown in Figures 7 and 8.

4. Evaluation of the Tool

To exemplify the benefits brought by the use of Verifica-UD, we conducted an experiment over a small corpus of 300 sentences (5,818 tokens) automatically annotated using UDPipe 2.0 parser [Straka 2018]. This set of automatically annotated sentences was previously submitted to a human revision that limit the corrections to the fields POS, HEAD, and DEPREL (ignoring LEMMA and FEAT fields of the automatically annotated CoNLL-U). We analyzed the initial corpus and the corpus revised through Verifica-UD to estimate the potential benefits of our tool to improve the human revision.

The human reviewer edited 173 out of the 300 sentences (58%), making amendments to the automatic annotation. We applied Verifica-UD over the original corpus, ignoring all rules that inspected the fields LEMMA and FEAT, as those fields were not contemplated by the human revision. This resulted in Verifica-UD indicating problems for 87 sentences over the 300 sentences (29%). The intersection of the human edited sentence set with Verifica-UD results was of 72 sentences. Observing these 72 edited sentences, we applied Verifica-UD to the human edited version and it turned out that 22

[3]Verifica-UD may be found at the POeTiSA project website (https://sites.google.com/icmc.usp.br/poetisa), but also at the links http://verificaud.icmc.usp.br:24080/verificaud/ and http://verificaud.icmc.usp.br/.


Atenção ⚠

O arquivo carregado contém **12** possíveis erros!

Arquivo **corpus.conllu** com **7** sentenças (**63** tokens)

Clique na sentença para ver os tokens com os possíveis erros.

[FOLHA_DOC000001_SENT003]

[FOLHA_DOC000097_SENT056]

[FOLHA_DOC000132_SENT031]

[FOLHA_DOC009367_SENT024]

Token 1 -ERRO P01: Todo token dependente de det é DET (o inverso não é verdadeiro).

Token 1 -ERRO T21: Todo token PRON deve ter features 'PronType=Dem/Ind/Rel/Int/Prs' e de acordo com o tipo pode ter apenas 'Case=Acc/Dat/Nom', 'Gender=Fem/Masc', 'Number=Sing/Plur', 'Person=1/2/3' e 'Poss=Yes'.

Token 4 -ERRO P57: Todo token head de aux:pass deve ter feature Voice=Pass.

Token 6 -AVISO P66: Normalmente, todo token PUNCT final tem como head o token root.

[FOLHA_DOC031010_SENT007]

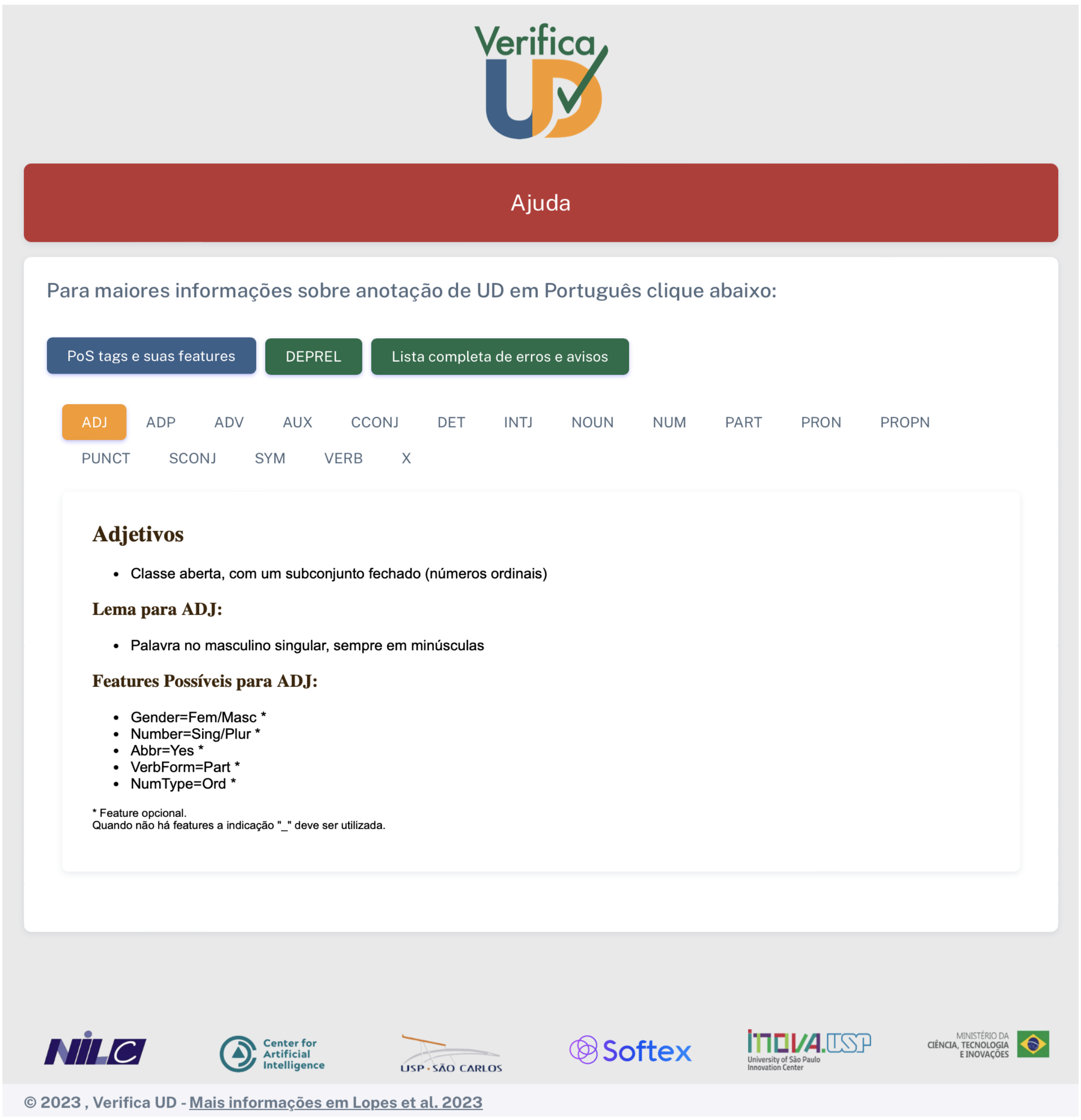
© 2023 , Verifica UD - Mais informações em Lopes et al. 2023

Figure 6. Verifica-UD Interface displaying detected errors and warnings.

of those sentences still presented problems according to our rules. We also noticed that 101 sentences were edited by the reviewer without indication of problems by Verifica-UD when applied to the original corpus. In 10 of these 101 edited sentences, the human edition corrected the HEAD error but forgot to adjust the DEPREL, thus generating new errors and warnings by our tool.

The direct benefits of Verifica-UD can be illustrated in 47 out of the 300 sentences:

- the 15 sentences with problems identified by Verifica-UD, but unnoticed by the reviewer;
- the 22 sentences where the reviewer edited, but problems remained, revealing that the problem detected by the tool was not the same as the one corrected by the human annotator; plus
- the 10 edited sentences that resulted in new verification issues.


Verifica UD
Ajuda
PoS tags e suas features
DEPREL
Lista completa de erros e avisos
ADJ
ADP
ADV
AUX
CCONJ
DET
INTJ
NOUN
NUM
PART
PRON
PROPN
PUNCT
SCONJ
SYM
VERB
X
NILC
Center for Artificial Intelligence
USP · SÃO CARLOS
Softex
INOVA USP University of São Paulo Innovation Center
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

Para maiores informações sobre anotação de UD em Português clique abaixo:

Adjetivos

- Classe aberta, com um subconjunto fechado (números ordinais)

Lema para ADJ:

- Palavra no masculino singular, sempre em minúsculas

Features Possíveis para ADJ:

- Gender=Fem/Masc *
- Number=Sing/Plur *
- Abbr=Yes *
- VerbForm=Part *
- NumType=Ord *

* Feature opcional.
Quando não há features a indicação "_" deve ser utilizada.

© 2023 , Verifica UD - Mais informações em Lopes et al. 2023

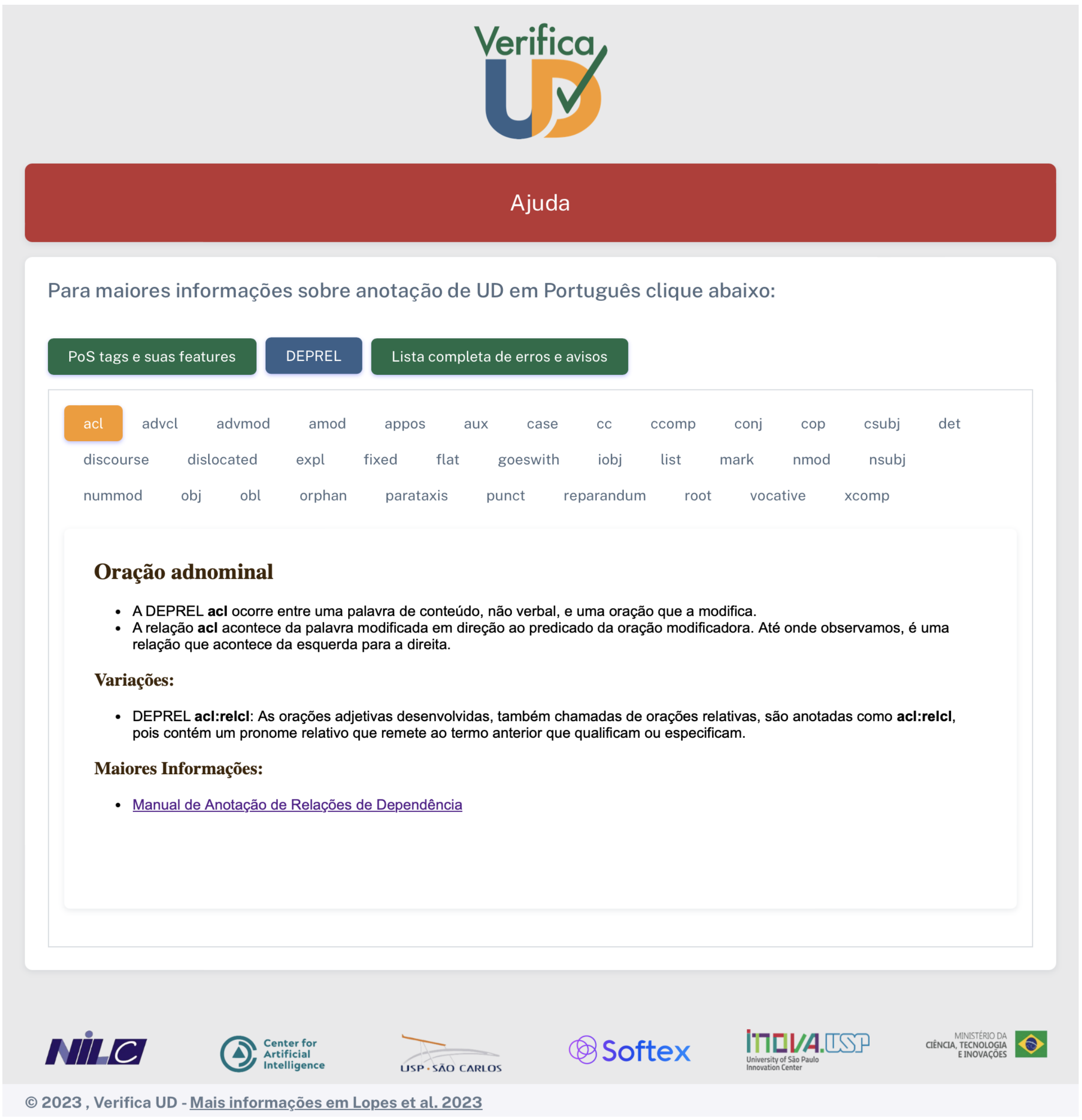
Figure 7. Verifica-UD Interface displaying help topics for tagging level.

Pushing the analysis a little further, we applied Verifica-UD with the full set of rules (including LEMMA and FEAT related ones) over the corpus version edited by the reviewer, and we discovered 81 additional sentences with potential problems. Therefore, in practical terms, using Verifica-UD, the human reviewer can be more productive in correcting these remaining 128 (47+81) sentences with problems pointed out by our tool.

5. Final Remarks

In this paper, we presented Verifica-UD, a web-based tool to verify possible problems in UD-annotated sentences (in CoNLL-U format) according UD guidelines for Portuguese. The tool has the potential to boost the production of annotated corpora in Portuguese, as well as to promote enhancement of the UD annotation in the Brazilian NLP community.

Verifica-UD was developed within the POeTiSA project (`https://sites.google.com/icmc.usp.br/poetisa`) and, according to our experiment, it


Ajuda

Para maiores informações sobre anotação de UD em Português clique abaixo:

Oração adnominal

- A DEPREL **acl** ocorre entre uma palavra de conteúdo, não verbal, e uma oração que a modifica.
- A relação **acl** acontece da palavra modificada em direção ao predicado da oração modificadora. Até onde observamos, é uma relação que acontece da esquerda para a direita.

Variações:

- DEPREL **acl:relcl**: As orações adjetivas desenvolvidas, também chamadas de orações relativas, são anotadas como **acl:relcl**, pois contém um pronome relativo que remete ao termo anterior que qualificam ou especificam.

Maiores Informações:

- Manual de Anotação de Relações de Dependência

© 2023 , Verifica UD - Mais informações em Lopes et al. 2023

Figure 8. Verifica-UD Interface displaying help topics for parsing level.

presents a good performance in terms of problem detection. Additionally, the availability of detailed help pages facilitates the human correction of the issues.

Future work includes the addition of new verification rules. It is also our plan to extend the tool with new edition and visualization functions.

Acknowledgments

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

References

de Marneffe, M.-C., Manning, C. D., Nivre, J., and Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Duran, M. S. (2021). Manual de anotação de PoS tags: Orientações para anotação de etiquetas morfossintáticas em língua portuguesa, seguindo as diretrizes da abordagem universal dependencies (UD). Technical Report 434, ICMC-USP.

Duran, M. S. (2022). Manual de anotação de relações de dependência: Orientações para anotação de relações de dependência em língua portuguesa, seguindo as diretrizes da abordagem universal dependencies (UD). Technical Report 440, ICMC-USP.

Grobol, L. (2021). VSCode language support for CoNLL-U. `https://github.com/LoicGrobol/vscode-conllu/blob/master/README.md`. Accessed: 2023-06-26.

Guibon, G., Courtin, M., Gerdes, K., and Guillaume, B. (2020). When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, pages 5293–5302, Marseille, France. European Language Resources Association.

Lopes, L., Duran, M., Fernandes, P., and Pardo, T. (2022). PortiLexicon-UD: a Portuguese lexical resource according to Universal Dependencies model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)*, pages 6635–6643, Marseille, France. European Language Resources Association.

Lopes, L., Duran, M. S., and Pardo, T. A. S. (2023a). Atribuição de lemas e atributos morfológicos seguindo as decisões adotadas na anotação do córpus Porttinari-base dentro das diretrizes da Universal Dependencies (UD). Technical Report -, ICMC-USP. To appear.

Lopes, L., Duran, M. S., and Pardo, T. A. S. (2023b). Verifica-UD - uma ferramenta online para verificação de textos em português anotados no formato CoNLL-U segundo o padrão Universal Dependencies. Technical Report -, ICMC-USP. To appear.

Miranda, L. G. M. and Pardo, T. A. S. (2022). UDConcord: a concordancer for universal dependencies treebanks. In *Proceedings of the Universal Dependencies Brazilian Festival (UDFest-BR)*, pages 1–10. Association for Computational Linguistics.

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, Marseille, France. European Language Resources Association.

Richardson, L. and Ruby, S. (2007). *RESTful Web Services*. O'Reilly, Beijing.

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.

Villa, L. B. (2022). Udeasy: a tool for querying treebanks in conll-u format. In *Proc. of the Workshop on Challenges in the Management of Large Corpora (CMLC)*, pages 16–19, Marseille, France. European Language Resources Association.