CL-CONLLU
Universal Dependencies in Common Lisp

Alexandre Rademaker[1,2], Fabricio Chalub[1]
, Bruno Cuconato[2], Henrique Muniz[1,2], and Guilherme Passos[1,3]

[1] IBM Research
[2] FGV/EMAp
[3] COPPE/UFRJ

Abstract. The growing interest in the *Universal Dependencies* project for creating corpora in different languages, using a common morphological and syntactic tags, motivate different research groups involved in the creation and maintenance of corpora the demand for tools for editing, correction and display of syntactic trees. Here we present `cl-conllu`, a Common Lisp library for manipulating CoNLL-U files, the file format used by the Universal Dependencies project.

1 Introduction

The use of different tags for morphological and syntactic annotations, as well as different annotation conventions, makes it difficult to develop multi-language syntax analysis tools and to study common linguistic phenomena between different languages [5]. To solve this problem, the Universal Dependencies (UD) project aims to create consistent linguistic annotations between different languages. Its annotation scheme is based on the Interset [11], Google Universal Tagset [7] as well as the Stanford Dependencies [3]. Recently, the UD project launched version 2.0 of its treebanks [6], already used in the shared task of the Conference on Computational Natural Language Learning (CoNLL 2017).

The advancement of the UD project demands tools for helping on corpora maintenance. In particular, as part of the UD-Portuguese-Bosque [9] corpus maintenance effort, we developed a library for manipulating the CoNLL-U files. The library provides features such as reading and writing CoNNL-U files, annotation validation, batch transformations, queries and the production of different views of syntax trees.

2 The CoNLL-U format

Following a syntactic model of dependencies, UD considers that each word is dependent on some other (except for the root of the phrase), and dictates its head, through a specific dependency relation. Besides, by its adoption of lexicalism, in UD the basic units of annotation are syntactic *words* (not spelling or phonological words) [5]. Hence, contractions and clitics are divided, for example, *do* is broken into two tokens *de+o*.

For the representation of annotations following these principles, the CoNLL-U format was developed. This format is an evolution of the CoNLL-X format, which in turn was an extension of the Malt-TAB format of Joakim Nivre [1]. Each file can contain multiple phrases, separated by a blank line. Each sentence starts with one or more lines containing metadata, such as the identifier and its original text. Then the words (tokens) are described one on each line. Also, multi-word tokens (orthographic tokens that have been broken into more than one word) receive a line of their own. Each word or tokens contains ten fields separated by a single tab character. These are: sequential numbering (ID); original form in the text (FORM); the lemma (LEMMA); the UD PoS tag (UPOS); language-specific grammatical class (XPOSTAG); morphological attributes (FEATS); the word index of its mother token (HEAD); the universal dependency relation (DEPREL); *enhanced* dependency relation (DEPS); and miscellaneous information (MISC). For the root token of the sentence, the dependency relation is `root` and its `head` is zero. Some fields, such as the FEATS field, support a list of values, separated by '|.'

3 The `cl-conllu` library

The `cl-conllu` library was developed in the Common Lisp (CL) language. It aims to make it easier to work with files in the CoNLL-U format. The library has essential features as reading and writing files in the format in question, conversion to other formats, queries over sentences loaded in memory, functions for evaluating annotations and comparing different annotations of the same sentence.

The primary data structures of the library are the sentence, token, and multiword token classes. A sentence has as its attribute ('slot 'in CL) the main list of its tokens and multiword tokens. We chose to keep the tokens and multi-word tokens in separate lists to facilitate the use of these structures by various other library functions.

The functions `read-conllu` and `write-conllu` are the functions for reading and writing CoNNL-U files, respectively. The first one, receives a 'string' or an object of class `pathname`.[4] It returns a linked list of objects of the class `sentence`. The `write-conllu` function receives a linked list of `sentence` objects and a file name and writes the sentences to the file. Among format conversions, the library currently supports the conversion of CoNLL-U files to Prolog and RDF.

Three significant recent additions to the library are: (1) a rule language to facilitate batch transformations; (2) visualization of the syntactic trees; and (3) a standard query language in syntax trees. These recent additions are the focus of this article.

[4] In CL, the 'pathname' class represents a path in the operating system's file system [10].

Starting with the visualization, the function `tree-sentence` receives a sentence and a ‘stream’ and produces a nice vertical tree showing the tokens connections. This function has been inspired by similar function in the UDAPI library [8].

To allow batch transformations, the `apply-rules-from-file` function has been implemented. This functionality was inspired by the program ‘Corte e Costura’ [4]. The function receives a list of rules, a CoNLL-U file to be read and a CoNLL-U to be generated. The function also produces a log file of the rule applications. The Listing 1.1 presents a rule with more than one pattern, with variables, in the left-hand side followed by a list of conditions. The variables are identifiers CL beginning with the character ”?”. The conditions are formed by an operator, a token field that we are interested in testing, and a string that can be a regular expression. The complete list of operators is presented in the library documentation on the Github repository.

Listing 1.1. Example of rule

```
(=> ((?a (match lemma ”[aA]te”))
     (?b (= lemma ”entao”)))
    ((?a (set upostag ”ADV”))
     (?b (set upostag ”ADV”))))
```

The query function operates over the trees. It was created to facilitate the localization of sentences given a pattern in the corpus. And the rule processor was built for batch correction of annotations (syntactic and morphological). The query language was inspired by [2].

Listing 1.2. query example

```
CL> (query '(nsubj
                (advcl
                    (and (upostag ”VERB”) (lemma ”correr”))
                    (upostag ”VERB”))
                (upostag ”PROP”)) *sentences*)
```

4 Conclusion

We intend to continue adding features to the library, such as: (1) better support for sentence validation; (2) expansion of the rule language with support for variables over expressions and not just variables for tokens, possibly combining the query language with the rules language; and (3) support for sentence editing interactively and other forms of syntactic tree visualization. Finally, we intend to add even more test cases to increase the robustness of the library. The library and its source code can be downloaded from the `http://github.com/own-pt/cl-conllu` repository and an initial documentation is in the same repository.

References

1. Buchholz, S., Marsi, E.: Conll-x shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language

Learning. pp. 149–164. CoNLL-X '06, Association for Computational Linguistics, Stroudsburg, PA, USA (2006), http://dl.acm.org/citation.cfm?id=1596276.1596305

2. Luotolahti, J., Kanerva, J., Pyysalo, S., Ginter, F.: Sets: Scalable and efficient tree search in dependency graphs. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. pp. 51–55. Association for Computational Linguistics (2015), https://aclweb.org/anthology/N/N15/N15-3011.pdf
3. de Marneffe, M., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C.D.: Universal stanford dependencies: A cross-linguistic typology. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014. pp. 4585–4592. European Language Resources Association (ELRA) (2014), http://www.lrec-conf.org/proceedings/lrec2014/summaries/1062.html
4. Mota, C., Santos, D.: Corte e costura no ac/dc: auxiliando a melhoria da anotação nos corpos. Setembro de (2009)
5. Nivre, J., de Marneffe, M., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R.T., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal dependencies v1: A multilingual treebank collection. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016. European Language Resources Association (ELRA) (2016), http://www.lrec-conf.org/proceedings/lrec2016/summaries/348.html
6. Nivre, J.e.a.: Universal dependencies 2.0 (2017), http://hdl.handle.net/11234/1-1983, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University
7. Petrov, S., Das, D., McDonald, R.T.: A universal part-of-speech tagset. In: Calzolari, N., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012. pp. 2089–2096. European Language Resources Association (ELRA) (2012), http://www.lrec-conf.org/proceedings/lrec2012/summaries/274.html
8. Popel, M., Žabokrtskỳ, Z., Vojtek, M.: Udapi: Universal api for universal dependencies. In: Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies, 22 May, Gothenburg Sweden. pp. 96–101. Linköping University Electronic Press (2017)
9. Rademaker, A., Chalub, F., Real, L., Freitas, C., Bick, E., de Paiva Universal Dependencies for Portuguese, V.: Universal dependencies for portuguese. In: Proceedings of the Fourth International Conference on Dependency Linguistics (Depling). pp. 197–206. Pisa, Italy (Sep 2017)
10. Steele Jr, G.L., Common, L.: the language. Digital Press **20**, 124 (1984)
11. Zeman, D.: Reusable tagset conversion using tagset drivers. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco. European Language Resources Association (2008), http://www.lrec-conf.org/proceedings/lrec2008/summaries/66.html