ArboratorGrew: Collaborative Curation for Treebank Manipulation

Khensa Daoudi[1], **Kim Gerdes**[2], **Gaël Guibon**[1], **Bruno Guillaume** [1], **Kirian Guiller**[3]
[1] LORIA, Inria, Université de Lorraine, CNRS, 54000 Nancy, France
[2] Lisn (CNRS), Université Paris-Saclay, France
[3] Modyco (CNRS), Université Paris-Nanterre, France

Relevant UniDive working groups: WG1, WG3

1 Introduction

Nowadays, the development of dependency treebanks is experiencing significant progress. Multilingual treebanks have been created as a part of the Universal Dependency (UD) (McDonald et al., 2013) and Surface Syntactic Universal Dependencies (SUD) (Gerdes et al., 2019) projects. The annotation and the maintenance of these treebanks requires a collective effort of researchers from diverse backgrounds in order to provide a larger understanding of linguistic phenomena. Different annotation tools can be integrated in the annotation process in order to enhance the quality of treebanks and ensure an effective collaboration between collaborators. ArboratorGrew[1] is a powerful software that assists the different phases of the annotation workflow. It is a web-based collaborative tool for dependency treebank annotation, it incorporates Grew (Bonfante et al., 2018) as a pattern search tool which is a valuable asset that facilitates the mining and the correction of the potential errors in the existing dependency treebanks. ArboratorGrew is undergoing continuous development. In this abstract, we will explore the different features that have been developed over the past few years. Through this exploration, we highlight the importance of these features in the development of high-quality dependency treebanks.

2 Collaborative Features

The treebank annotation is a laborious and time-consuming task that requires the participation of multiple persons. Managing this collaboration becomes a major feature of any annotation tool (Cucurnia et al., 2021; Pei et al., 2022). ArboratorGrew provides a set of features that facilitate the collaboration and ensure an effective coordination between annotators during the annotation process. Additionally, the collaborative aspect allows ArboratorGrew to be used as an educational tool for syntax teaching.

Projects Organization. ArboratorGrew's corpora are organized into projects. These projects can be either private, visible without modification access or public. The tool also provides multiple user roles as follows:

- Admin: they set up the project, provide the annotation schema. They invite and define the roles of the collaborators, they are also in charge of preparing and uploading the corpus to be annotated.
- Validator: the project's linguist, they track the quality of the annotations made by annotators, correct the errors and validate the gold standard annotation of each sentence.
- Annotator: they can browse and edit the treebank by saving a modified tree under their name. Every sample has a list of annotators.
- Guest: they are users with read-only access in the private projects.

Labeling System. ArboratorGrew's labeling system is a key feature that helps annotators in categorizing and organizing their work. It is possible for users to add labels to their trees by either using predefined labels or creating new ones. Labels are mainly utilized to indicate the progress of the annotation process and can prioritize the sentences by designing those to be annotated at the future stages. The labels serve as means of communication between the annotators. This feature improves the efficiency of the validation phase, as the validator can examine annotations that have been marked as completed using the tag system.

Reference-less Annotation Mode. In this mode the users do not have access to other users' trees. The annotations are hidden to avoid bias and to assess the quality of the corpus through inter-annotator agreement. This mode is suitable for teaching students syntactic annotation. It allows admins to create exercises with different levels

[1] https://arborator.github.io

of complexity: **1:reference visible**, when editing, the annotators can see the reference tree, the differences between the reference tree and the user annotation are highlighted in red and the pourcentage of the correct annotation is provided. **2:local feedback**, here the reference tree is not visible but the differences and the statistics are still given. **3:global feedback**, only the statistics are available and **4:no feedback** where only the validator can see the annotation, calculate and export the annotator's scores. ArboratorGrew offers a way to perform pairwise annotators comparison by providing a *difference mode* feature that helps the validator to compare between two annotations by highlighting the differences. It is a useful tool that helps the validators or the admins of a project to better identify the things that need more clarification and training.

Github Synchronization. Github[2] is designed for collaboration on coding projects. However, it can also be a valuable tool for researchers to store and share data publicly. Github is a well-known and largely used tool in the Natural Language Processing (NLP) community for storing and managing treebanks. ArboratorGrew offers a full synchronization with Github, this allows users to track the progress of their work and revert to previous changes using the versioning system. In addition, it expands ArboratorGrew's collaborative environment by allowing users to work with colleagues who are not necessarily members of the ArboratorGrew project. To use this feature, users must authenticate themselves by using github social authentication. The admin can choose either to work with the main branch or to create and work with a new dedicated branch which will be used only for commits and pulls made with ArboratorGrew. This option allows users to preserve the main codebase integrity. All changes made by annotators are counted, and when the admin is ready, they can push directly to Github. Meanwhile, ArboratorGrew listens to changes made in the Github repo, the admin can directly retrieve these changes to their ArboratorGrew projects.

3 Treebank Curation

ArboratorGrew integrates a query system called Grew. This integration allows annotators to make requests in their corpora and to detect potential errors or annotation inconsistencies. In addition to Grew search, ArboratorGrew also includes other features for efficient error detection and correction. Here are the main ones:

Rule-based Annotation Editing. In order to fix recurrent errors or to make systematic annotation revision, it is possible to directly edit treebanks using Grew edition rules. Sentences that match the specified request will be modified and the modified part is highlighted. To save these modifications, users can choose to either select individual sentences or opt for all.

Lexicon. Dictionaries and lexicons are an important resource for many NLP applications (Catelli et al., 2022; Ahia et al., 2023). ArboratorGrew allows lexicon generation based on some subset of features, POS, lemma and/or form. Moreover, ArboratorGrew offers the ability to explore ambiguous entries in the lexicon by choosing a sublist of features where the same entries have different combinations of the chosen features.

4 Parser

ArboratorGrew offers the possibility to use the parser BertForDeprel (Guiller, 2020), which helps linguists to start their work with pre-annotation. The integrated parser has given encouraging results on low resource languages. The users of ArboratorGrew can either parse the data using the existing pretrained models or train and parse using their own data. All operations can be done from the web interface.

5 Conclusion

ArboratorGrew is currently under continuous development, the features that have been added or improved are a result of various user interactions and recommendations. We aim to enhance the user experience by supporting keyboard-only annotation to improve the annotator's productivity. We plan to create an intelligent system that will assist the user by suggesting relevant choices while annotating. Finally, we are willing to integrate validation scripts that will be applied on the treebanks in order to help detecting the errors and ensure the consistency of the annotation work.

Acknowledgements

This project has received funding from the French National Research Agency's grant Autogramm ANR-21-CE38-0017.

[2] https://github.com

References

Orevaoghene Ahia, Hila Gonen, Vidhisha Balachandran, Yulia Tsvetkov, and Noah A. Smith. 2023. LEXPLAIN: Improving model explanations via lexicon supervision. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 207–216, Toronto, Canada. Association for Computational Linguistics.

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*, volume 1 of *Logic, Linguistics and Computer Science Set*. ISTE Wiley.

Rosario Catelli, Serena Pelosi, and Massimo Esposito. 2022. Lexicon-based vs. bert-based sentiment analysis: A comparative study in italian. *Electronics*, 11(3):374.

Davide Cucurnia, Nikolai Rozanov, Irene Sucameli, Augusto Ciuffoletti, and Maria Simi. 2021. Matilda-multi-annotator multi-language interactivelight-weight dialogue annotator. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 32–39.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2019. Improving Surface-syntactic Universal Dependencies (SUD): MWEs and deep syntactic features. pages 126–132.

Kirian Guiller. 2020. Analyse syntaxique automatique du pidgin-créole du Nigeria à l'aide d'un transformer (BERT) : Méthodes et Résultats.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. 2022. Potato: The portable text annotation tool. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 327–337.