Collaborative development of annotation guidelines with application to Universal Dependencies

Sampo Pyysalo, Filip Ginter

Department of Information Technology
University of Turku, Finland
sampo@pyysalo.net, filip.ginter@utu.fi

Abstract

We introduce an online linguistic annotation guideline development system that supports simple authoring, visualization of complex structured annotation, version control, and rich collaboration features. The system is presented in the context of a project aiming to develop universally applicable treebank annotation guidelines and apply them to a broad range of languages. The system is open source and available from http://spyysalo.github.io/annodoc/, and the Universal Dependencies documentation is available from http://universaldependencies.github.io/docs/.

1. Introduction

The development and maintenance of comprehensive, up-to-date guidelines with detailed examples is a challenging but necessary part of any larger linguistic annotation effort. Web-based document management systems have many potential benefits for collaboration and access, but there is a lack of solutions that combine ease of editing with explicit support for annotation guideline development.

The Universal Dependencies (UD) project is a collaborative effort building on Stanford Dependencies (de Marneffe et al., 2006; de Marneffe et al., 2014), Google universal part-of-speech tags (Petrov et al., 2012), the Interset morphosyntactic tagsets (Zeman, 2008), and related recent proposals (McDonald et al., 2013; Tsarfaty, 2013; Rosa et al., 2014) to develop universally applicable treebank annotation guidelines and to apply these to create cross-linguistically consistent treebank annotation for many languages. At present, the project involves members from more than 10 institutions and targets 16 languages. Collaboration is carried out almost exclusively online.

To support the needs of the UD effort, we combined existing and newly developed components into a novel guideline development system that provides simple authoring, visualization of complex structured annotation, version control, and rich collaboration features.

2. Features

The core function of the introduced system is to take plain text-like documents that include annotation examples in easy-to-write formats such as the Stanford Dependency (SD) format, and to generate corresponding web pages with vector graphics-based annotation visualizations (Figure 1). The system thus allows the source documents to remain simple and removes the need for authors to be familiar with technologies such as HTML, SVG and JavaScript.

To facilitate collaboration between multiple authors and to organize different versions of developing guidelines, the system is implemented within a distributed version control system that tracks changes and resolves any conflicts that may arise from two or more editors working on the same document. The generation of the web documents from the source is then automated so that a new version of the guidelines is automatically published whenever the version-controlled source is updated.

Other properties of the system include

- Online editing integrated with version control
- Full Unicode, supporting any language and writing system
- Inline HTML support, for complex source documents
- Basic scripting facilities, for e.g. automating document listings
- Fully configurable visualization with support for nearly any form of text annotation
- Multiple annotation formats
- Issue tracking and discussion features
- Compatibility with browser-based export to PDF

3. Implementation

We next briefly present the technologies used to implement the system.

Git version control system with a focus on distributed development (http://git-scm.com/). Provides revision management, conflict resolution, etc.

GitHub web-based hosting service for Git repositories (http://github.com/). Provides storage, online editing, and facilities for issue tracking and discussion.

Markdown plain text-based document format emphasizing readability and ease of writing, designed for automatic conversion into HTML (http://daringfireball.net/projects/markdown/).

Jekyll static website generator that supports Markdown and the templating and scripting language Liquid, generating HTML documents (http://jekyllrb.com).

Scalable Vector Graphics (SVG) an XML-based format for two-dimensional graphics, supported by modern browsers.

BRAT an online tool for annotation visualization and editing (http://brat.nlplab.org) (Stenetorp et al., 2012). Generates SVG-based annotation visualizations.

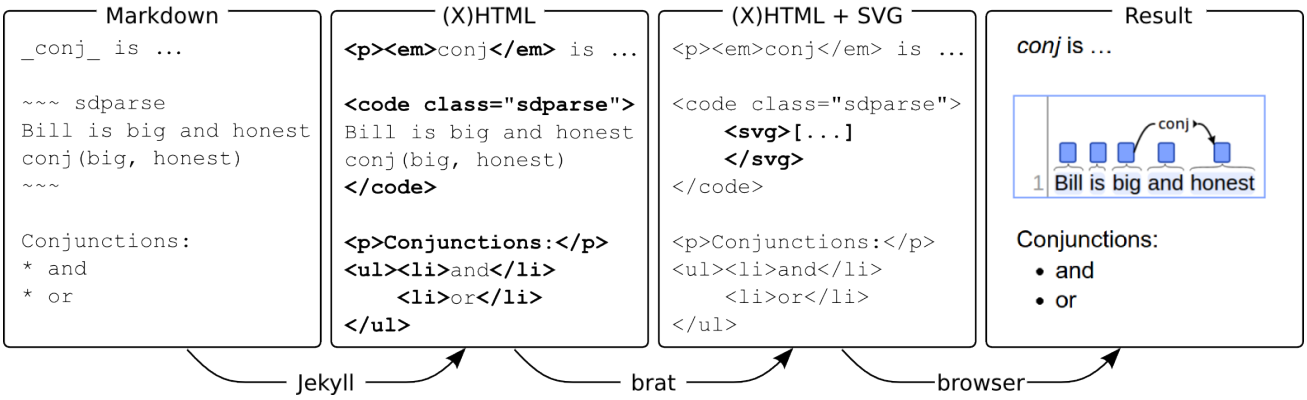

Figure 1: Processing from Markdown input into document with visualizations.

The key processing stages are illustrated in Figure 1. In brief, the source documents are formatted as Markdown extended with visualizations (e.g. in the SD format). These documents are kept in a Git repository that can be accessed either directly or via GitHub online editing facilities. On any update to the source, Jekyll is executed to generate (X)HTML, attaching visualization-related JavaScript code, and the resulting documents are made available online. When any of these documents are opened in a browser, our custom JavaScript component executes, replacing elements with visualizable content with embedded BRAT and the appropriate input. Finally, the embedded BRAT client runs in the browser to generate the SVG corresponding to the visualization, creating the final document.

We implemented new functionality to allow client-side embedded BRAT to visualize annotations in the SD, CoNLL-X and CoNLL-U formats in addition to the native `.ann` format of the tool. We additionally implemented automatic BRAT embedding and visualization, eliminating the need for document authors to write JavaScript to embed annotation visualizations. Excepting for GitHub, all of the used technologies are open source. Further, the use of GitHub could be replaced with an open source system providing similar functionality, such as GitLab (`http://gitlab.com`).

4. Universal Dependencies

The UD project for which the documentation system was originally developed also serves as a demonstration of the scalability of the system in supporting a collaborative effort involving 20 contributors working on treebanks for a number of languages whose documentation is tightly interlinked. The UD project released the first stable version of its general guidelines on October 1st, 2014, defining 17 POS tags building on the Google universal tagset, 17 features using the Interset inventory, a 40 dependency relation variant of the universal Stanford Dependencies, and a new CoNLL format extension, CoNLL-U. Work is now focusing on creating language-specific annotation guidelines and converting existing treebanks for various languages to the universal standard, using resources such as the English Web Treebank (Petrov and McDonald, 2012), the Turku Dependency Treebank (Haverinen et al., 2013), and the Swedish Treebank (Nivre and Megyesi, 2007). A first release of treebank data, aiming to cover at least 10 different languages, is planned for January 2015.

In addition to extensive documentation for the universal POS tags, features and dependency relations, initial drafts of language-specific documentation for English and Finnish have so far been introduced using the documentation system. In total, these sets of documentation currently consist of approximately 40,000 words as well as 386 visualized annotation examples: 218 for UD, 91 for English, and 77 for Finnish. As of this writing, the system is further set up with templates for language-specific documentation for Basque, Bulgarian, Czech, French, German, Greek, Hebrew, Hungarian, Irish, Italian, Korean, Persian, and Spanish. These templates cover in total over 1500 markdown pages, which can be converted into HTML in a few minutes on a standard desktop machine. The system can thus scale to large documentation efforts also in terms of its computational efficiency.

5. Conclusions

We have introduced a distributed guideline development and documentation system, addressing the current lack of an open-source, user-friendly, web-based solution. Being deployed at GitHub, the system also allows contributions from third parties in a controlled fashion and protection against any form of data loss. Although initially implemented for the Universal Dependencies effort, the system is fully general and can be readily used in other projects and for forms of annotation other than dependency syntax. The system supports a range of formats allowing annotations to express arbitrary text span annotations and any relation structure, and can thus be readily applied to named entity recognition, chunking, coreference, relation extraction, and event extraction tasks, among others.

Acknowledgements

This paper builds on joint work with Jinho Choi, Tim Dozat, Yoav Goldberg, Jan Hajič, Christopher Manning, Marie-Catherine de Marneffe, Ryan McDonald, Joakim Nivre, Slav Petrov, Natalia Silveira, Reut Tsarfaty, and Dan Zeman.

References

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, volume 6, pages 449–454.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, volume 14, pages 4585–4592.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2013. Building the essential resources for finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, pages 1–39.

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 (ACL)*, pages 92–97.

Joakim Nivre and Beata Megyesi. 2007. Bootstrapping a swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*, pages 97–102.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, volume 59.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, pages 2089–2096.

Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zdeněk Žabokrtský. 2014. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 2334–2341.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 102–107.

Reut Tsarfaty. 2013. A unified morpho-syntactic scheme of stanford dependencies. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 578–584.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pages 213–218.