Agile creation of multi-layer corpora with corpus-tools.org

Stephan Druskat[1], Thomas Krause[1] and Carolin Odebrecht[1]
[1]Humboldt-Universität zu Berlin, Dept. of German Studies and Linguistics, Berlin, Germany
[1]{stephan.druskat, krauseto, carolin.odebrecht}@hu-berlin.de

Agile corpus creation [3] is a methodology which replaces the linear-phase process of traditional corpus creation with iterative cycles of corpus query, annotation scheme edits, annotation and analysis. We demonstrate corpus-tools.org, a suite of generic tools tailored to the agile creation of multi-layer corpora. It consists of Salt [4], a graph-based meta model and API for linguistic data; Pepper [5], a conversion tool and platform for linguistic data; Atomic [1], an extensible annotation software; and ANNIS [2], a search and visualization architecture for multi-layer corpora. We demonstrate our tools in the correction of syntax trees in a multi-layer corpus, using an extension to Atomic which re-uses ANNIS to query complex annotation subgraphs containing errors.

While Atomic can potentially be used for various different annotation concepts (tokens, spans, trees, relations) due to its extensibility via plugins, and is therefore generally a suitable tool for annotation cycles over multi-layer corpora, it does lack search capabilities to properly cater for truly agile workflows.

ANNIS, on the other hand, provides an existing search and visualisation system, based on annotation graphs similar to Salt – which in turn is used by Atomic for internal representation of data. ANNIS also provides the expressive ANNIS Query Language (AQL). However, ANNIS is optimised for a static creation workflow, where analysis is a singular process at the very end of the workflow: A complete corpus is converted with Pepper and imported into ANNIS once, as opposed to recurring updates of single documents. As of yet, and in contrast to Atomic, ANNIS is also not self-contained, i.e., it depends on a separate installation of the PostgreSQL relational database[1], which complicates its setup for end-users.

These drawbacks will be compensated for by a new, C++-based implementation of ANNIS: graphANNIS[2]. While still under development, it already supports a large subset of AQL, and its data representation is much more closely aligned with the Salt data model than the implementation using the relational database. graphANNIS has a Java API[3], and is distributed as a self-contained library via the Maven build and dependency system. Its encapsulation allows for graphANNIS to be embedded in other tools and pipelines, and thus it will be integrated into Atomic as its search engine. Atomic will still be responsible for the storage of corpus data, while graphANNIS provides an additional index which is updated whenever a document is changed by the user. For search tasks, Atomic will provide a GUI section for entering AQL queries. These will then be parsed by the ANNIS AQL parser and passed to the graphANNIS search system. graphANNIS will subsequently return solely the Salt IDs of the matched nodes, which Atomic will use to display the results in its views and editors. This setup will provide corpus-tools.org with the capabilities needed for truly agile multi-layer corpus creation workflows.

[1] Stephan Druskat, Lennart Bierkandt, Volker Gast, Christoph Rzymski, and Florian Zipser. Atomic: an open-source software platform for multi-level corpus annotation. In *Proceedings of the 12th Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2014)*, pages 228–234, Hildesheim, Germany, 2014.

[2] Thomas Krause and Amir Zeldes. ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1):118–139, 2016. URL `http://dx.doi.org/10.1093/llc/fqu057`.

[3] Holger Voormann and Ulrike Gut. Agile corpus creation. *Corpus Linguistics and Linguistic Theory*, 4(2):235–251, 2008. doi: 10.1515/CLLT.2008.010.

[4] Florian Zipser and Laurent Romary. A model oriented approach to the mapping of annotation formats using standards. In *Proceedings of the Workshop on Language Resource and Language Technology Standards*, 2010. Seventh International Conference on Language Resources and Evaluation (LREC 2010), Valletta, Malta.

[5] Florian Zipser, Amir Zeldes, Julia Ritz, Laurent Romary, and Ulf Leser. Pepper: Handling a multiverse of formats, 2011. Poster presented at 33. Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft, 24 February, Göttingen University, Göttingen, Germany.

[1] `http://postgresql.org/`

[2] `https://github.com/thomaskrause/graphANNIS`

[3] C++ and Java are bridged by the JavaCPP library (`https://github.com/bytedeco/javacpp`), which also allows to package native binaries together with the system-independent JAR-files.