

Querying GrAF data in linguistic analysis

Peter Bouda, Centro Interdisciplinar de Documentação Linguística e Social, Minde/Portugal,
pbouda@cidles.eu

The “Graph Annotation Framework” (GrAF) defines an API and an XML format to store and query linguistic annotations as annotation graphs. The format was standardized as ISO 24612 in 2012¹, and was explicitly developed as an underlying data model for linguistic annotations in a radical stand-off approach² ([Ide and Suderman 2007]). The basic data structures are annotation graphs as proposed in [Bird and Liberman 2001], and are general and expressive enough to encode all known varieties of annotation in linguistics and other “annotation-based” disciplines. Although GrAF is not a TEI-compatible format, both standards share a certain technological foundation and grew in a similar ecosystem, but with slightly different applications in mind. In our talk we will show the connections between TEI and GrAF, propose an option to convert between the „two worlds“, and demonstrate a query system for GrAF data that we already use in typological analysis of annotated data from language documentation projects.

Historically, GrAF was developed as a standoff version of the “Corpus Encoding Standard for XML” (XCES), which is a TEI application for computational linguistics. In contrast to XCES, the development of the GrAF standard took place under the ISO umbrella from the beginning. In our project we use a Python implementation³ of GrAF 1.0. The American National Corpus developed the schemata (Relax NG, W3C and DTD) for GrAF 1.0 with the TEI Roma program and ODD files, so that GrAF and TEI share at least the basic data types in its schema definitions⁴. Furthermore, the feature structures of annotations in GrAF are TEI compliant. On top of GrAF, we developed a library Poio API⁵ which primarily maps between file formats used in language documentation projects and GrAF data structures, and adds an advanced query system on top of the GrAF API. Our goal with Poio API is to use GrAF as pivot data structure to

- 1) convert between the supported file formats, and
- 2) to have a unified entry point to linguistic analysis pipelines.

At the moment we are implementing parsers and writers for TEI-compatible files, so that TEI data can later be used in the same or similar linguistic workflows that we currently apply on other file formats.

As the GrAF standard already defines an API we are ready to use this interface to query the data from annotation graphs. We decided not to use XSLT or XQuery in our project (although there exists an GrAF/XML serialization of our data, see [Blumtritt et. al. 2013]), as we see programming an API in Python or Java as much more powerful, extensible and easier to teach than the equivalent XML technologies. We have several scientific tools and libraries that already support GrAF data, for example the UIMA and GATE workflow systems for Java programmers⁶ and the whole ecosystem of “Scientific Python” with linguistic, numerical and statistical packages. In addition to this, Python and Java provide means to quickly develop user interfaces for desktop computers and the web, so that we are able to hide the complexity of any API or query language from users, which in our case do normally not have any computational background.

There are two main options to query GrAF data in Poio API:

- 1) Via directly accessing the GrAF API or
- 2) through filters and filter chains in Poio API.

In the first case, the programmer uses the semantics of annotations graphs to apply graph-traversal

1 http://www.iso.org/iso/catalogue_detail.htm?csnumber=37326, accessed 21.3.2013

2 In contrast to TEI, which does still not support standoff annotation without markup; see discussions in [Cayless and Soroka 2010] and [Bański and Przepiórkowski 2009] for an in-depth analysis of the problem.

3 <http://media.cidles.eu/poio/graf-python/>, accessed 26.8.2013

4 <http://www.xces.org/ns/GrAF/1.0/>, accessed 26.8.2013

5 <http://media.cidles.eu/poio/poio-api/>, accessed 26.8.2013

6 <http://www.anc.org/software/>, accessed 26.8.2013

techniques or other graph-based methods to query and analyze the data. The search is memory based, the user might decide to load the full graph or only a part of it. We will present an example how we transform the annotation graphs to a general network in order to provide an overview over dictionary entries in several dozens of native South-American languages. This method was also successfully used to measure semantic similarities between dictionary entries in the data.

In the second case the user or programmer stays within the semantics of the tier-based data, which is much more common for our end users. Here, for each of the tiers (like “utterance”, “translation”, “morpho-syntactic”, etc.), the user might enter search terms and regular expressions to filter the data, i.e. only receive a subset of the data that contains matches for each search term on each tier. The user can then apply another filter on this subset, creating a filter chain to further narrow down the search result. We will present a desktop GUI called “Poio Analyzer”⁷ that is already in use in analysis projects and that demonstrates the filters and filter chains. An initial idea of Poio API was to ease the creation of custom annotation and edit tools for individual research projects. We did not want to build the “swiss-army knife” for linguists, but allow researchers to adapt the software to their current needs. The code of Poio Analyzer consists of only a few hundred lines of Python code and can easily be adapted. Based on Poio Analyzer we are currently implementing a web application called CLASS⁸ (Cologne Language Archive Services) that will contain this and other functionality.

References

[Bański and Przepiórkowski 2009] Bański, Piotr and Przepiórkowski, Adam. 2009. Stand-off TEI annotation: the case of the National Corpus of Polish. In: *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pp. 64–67, Singapore.

[Bird and Liberman 2001] Steven Bird and Mark Liberman. 2001. A formal framework for linguistic annotation. *Speech Commun.* 33, 1-2 (January 2001), 23-60.

[Blumtritt et. al. 2013] Blumtritt, Jonathan, Peter Bouda and Felix Rau. 2013. [Poio API and GraF-XML: A radical stand-off approach in language documentation and language typology](#). Presented at Balisage: The Markup Conference 2013, Montréal, Canada, August 6 - 9, 2013. In *Proceedings of Balisage: The Markup Conference 2013*. Balisage Series on Markup Technologies, vol. 10. (URL: <http://www.balisage.net/Proceedings/vol10/html/Bouda01/BalisageVol10-Bouda01.html>, accessed 26.8.2013)

[Cayless and Soroka 2010] Cayless, Hugh A. and Soroka, Adam. 2010. [On implementing string-range\(\) for TEI](#). In: *Proceedings of Balisage: The Markup Conference 2010* (URL: <http://www.balisage.net/Proceedings/vol5/html/Cayless01/BalisageVol5-Cayless01.html>, accessed 26.8.2013)

[Ide and Suderman 2007] Ide, Nancy and Suderman, Keith. 2007. GrAF: A graph-based format for linguistic annotations. In: *Proceedings of the Linguistic Annotation Workshop*, pp. 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

7 <http://media.cidles.eu/poio/poio-analyzer/>, accessed 26.8.2013

8 <http://class.uni-koeln.de>, accessed 26.8.2013