discoursegraphs: A graph-based merging tool and converter for multilayer annotated corpora

Arne Neumann
Applied Computational Linguistics
SFB 632 / EB Cognitive Science
Universität Potsdam, Germany
arne.neumann@uni-potsdam.de

Abstract

`discoursegraphs` is a Python-based converter for linguistic annotation formats which facilitates the combination of several, heterogeneous layers of annotation of a document into a unified graph representation. The library supports a range of syntax and discourse-related formats and was successfully used to revise and merge a multilayered corpus (Stede and Neumann, 2014).

1 Introduction

In an ideal world, we would like to have an easy-to-use annotation tool that supports a wide range of annotation tasks, uses a standard-compliant interchange format and which can be easily extended – in a novice friendly programming language. While there has arguably been progress in the field of general-purpose annotation software in recent years (e.g. `brat` (Stenetorp et al., 2012) and `WebAnno` (Yimam et al., 2013)), hierarchical and higher order annotation remains the domain of specialised programs (e.g. `RSTTool` (O'Donnell, 2000) and `MMAX2` (Müller and Strube, 2006)) using idiosyncratic file formats, written and last maintained by brave colleagues in the dark ages of computer history.

To honor the contributions of these fellow minds, I have implemented a simple and easily extendable toolkit called `discoursegraphs`, which can convert a number of syntax and discourse-related annotation formats and is able to merge these annotations into a single graph for further exploration or transformation into other, more sustainable formats. The library is free and open-source software and is available from its repository[1]. It can also be installed directly via Python's `pip` package manager[2].

2 Related Work

There are numerous converters for linguistic annotations, but they usually only convert between a limited set of file formats and are geared towards specific projects or focus on one type of annotation (e.g. *treetools*[3] for Treebank formats). To the best of my knowledge, there's only one other off-the-shelf converter that supports merging heterogeneous annotations into a unified data structure: `SaltNPepper` (Zipser et al., 2010; Zipser et al., 2014). Despite its wide range of import and export formats (and its recent addition of merging capabilities), I chose to write my own toolkit for the sake of simplicity and maintainability.[4]

3 System Architecture

`discoursegraphs` is implemented in Python 2.7 and uses the `NetworkX` library (Hagberg et al., 2008) to represent annotated documents as graphs.

`DisourseDocumentGraph` is the fundamental data structure of the library. It is a directed graph with (possibly) multiple edges between nodes. Each token in a document is represented by a node with token-level features (e.g. part-of-speech tag and lemma) encoded as attribute-value pairs.

All nodes and edges belong to at least one annotation layer (with possible sub-layers, e.g. `'syntax'` vs. `'syntax:category'`, `'syntax:token'` or

[1] `https://github.com/arne-cl/discoursegraphs`

[2] `https://pip.pypa.io`

[3] `https://github.com/wmaier/treetools`

[4] *SaltNPepper* is a versatile, mature library – there's even an annotation tool based on it (Druskat et al., 2014) – but it is also rather heavy-weight. The core of *SaltNPepper* (not including importers and exporters) already consists of roughly 60,000 lines of Java, while *discoursegraphs*' core consists of only 750 lines of Python.

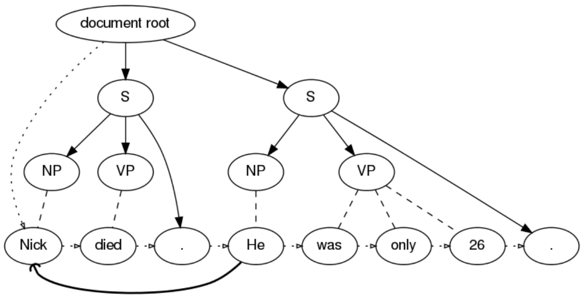

Figure 1: Example document containing two sentences with syntax and coreference annotation.

'rst' vs. 'rst:nucleus'), which they can be easily queried for.

Annotations are expressed as additional nodes (e.g. for elements in a constituency parse tree) and directed edges between them. Both annotation nodes and edges can have additional attributes stored in attribute-value pairs. Namespaces are used in order to allow conflicting annotations to be merged. For example, a token node may have two part-of-speech annotations associated with it (e.g. 'penn:vbz' and 'brown:doz').

The toolkit relies on four basic types of edges (Figure 1) to model linguistic annotations ranging from syntax to semantics, discourse phenomena and information structure:

- **spanning relation**: one span root node with outgoing edges to all (token) nodes the span covers – signifies a contiguous span of tokens, e.g. a phrase or a named entity [dashed line without arrow]
- **dominance relation**: a hierarchical annotation, e.g. from a noun phrase to a noun in a constituent structure [solid line with black arrow]
- **pointing relation**: a non-hierarchical relation, e.g. for linking coreferent entities [bold solid line with curved arrow]
- **precedence relation**: a path, starting from the document root node through all tokens in the order they occur in the document and ending at the last token [dotted line with unfilled arrow]

While typed edges are not strictly necessary to represent linguistically annotated data in graphs[5], they avoid ambiguity – especially when working with unknown corpora or when multiple tools have to work on the same dataset, cf. Neumann et al. (2013).

3.1 Importers

discoursegraphs includes importers for the following tools and formats: (i) constituent and dependency structures: Tiger-XML (Mengel and Lezius, 2000), Penn Treebank (Prasad et al., 2008) and CoNLL 2009/2010 (Hajič et al., 2009; Farkas et al., 2010), (ii) rhetorical structure: RSTTool's (O'Donnell, 2000) rs3 and rst/dis formats, (iii) pointing relations (e.g. coreference, connectives): MMAX2 (Müller and Strube, 2006) and ConAno (Stede and Heintze, 2004), and (iv) annotations of spans of text: EXMARaLDA (Schmidt, 2004).

Additional importers can easily be implemented by parsing an input format (e.g. with lxml[6]) and adding its tokens as nodes to a DisourseDocumentGraph. Afterwards, annotation nodes and edges can be added. To simplify the development of complex converters, you can add annotations iteratively and use the library's visualisation and document statistics functions (cf. Section 4) to check if the resulting graph matches your expectations.

3.2 Exporters

The library also provides a number of exporters for (i) general purpose graph formats like dot (Ellson et al., 2002), GEFX[7], GML[8] and GraphML (Brandes et al., 2013), (ii) the linguistic interchange formats CoNLL 2009 and PAULA XML 1.1 (Zeldes et al., 2013), (iii) the neo4j graph database[9] – both regular export via the geoff format, as well as live upload of annotated graphs to a running neo4j instance, and (iv) EXMARaLDA's exb format.

4 Usage

The API of the library has been kept deliberately simple. These five lines are all it takes to parse a document with two different annotation layers (syntax and rhetorical structure) into document graphs, merge them and convert them into a format that can be read by neo4j:

[5] For example, the ISO-standardised *Linguistic Annotation Framework* (ISO 24612, 2012) does allow type annotations on edges, but does not require them.

[6] http://lxml.de/

[7] http://gexf.net/format/

[8] http://www.fim.uni-passau.de/en/fim/faculty/chairs/theoretische-informatik/projects.html

[9] http://neo4j.com/

```
import discoursegraphs as dg
docgraph = dg.read_tiger('in.xml')
rstgraph = dg.read_rs3('in.rs3')
docgraph.merge_graphs(rstgraph)
dg.write_geoff(docgraph, 'out.geoff')
```

Document conversion and annotation merging is also available via a command-line interface. Beyond merging, the API provides functions for basic document statistics and graph visualisations (using the browser-based IPython (Pérez and Granger, 2007) notebook[10] and its dot plugin[11]).

discoursegraphs provides functions to select nodes and edges based on their properties (e.g. membership in a layer, edge type, annotations etc.). Combined with the graph manipulation capabilities of *NetworkX*, this e.g. allows the user to extract meaningful substructures from multi-level annotated documents or to create trees that combine syntactic and discourse information for kernel-based machine learning, as in Joty and Moschitti (2014).

5 Future Work

I plan to extend discoursegraphs with im- and exporters for further interchange formats, i.e. GrAF (Ide and Suderman, 2007), FoLiA (van Gompel and Reynaert, 2013) and especially Salt (Zipser et al., 2010), in order to leverage SaltNPepper's broader variety of supported formats, which would in turn also allow users to use merged corpora in the ANNIS linguistic query and visualisation tool (Krause and Zeldes, 2014).

Acknowledgments

This work was supported by Deutsche Forschungsgemeinschaft as part of the funding for project D1 in the Collaborative Research Center 632 'Information Structure' (Univ. Potsdam, HU Berlin and FU Berlin). I would also like to thank Matthias Bisping, Yulia Grishina and Tatjana Scheffler for testing the software and reporting bugs.

References

Ulrik Brandes, Markus Eiglsperger, Jürgen Lerner, and Christian Pich. 2013. Graph markup language (GraphML). In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*. CRC Press.

Stephan Druskat, Lennart Bierkandt, Volker Gast, Christoph Rzymski, and Florian Zipser. 2014. Atomic: an open-source software platform for multi-level corpus annotation. In *Proceedings of the 12th edition of the KONVENS conference Vol. 1*. Universität Hildesheim.

John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull. 2002. Graphviz–open source graph drawing tools. In *Graph Drawing*, pages 483–484. Springer.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, pages 1–12. Association for Computational Linguistics.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

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

ISO 24612. 2012. *Language Resource Management – Linguistic Annotation Framework*. International Standards Organization, Geneva, Switzerland.

Shafiq Joty and Alessandro Moschitti. 2014. Discriminative Reranking of Discourse Parses Using Tree Kernels. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2049–2060. Association for Computational Linguistics.

Thomas Krause and Amir Zeldes. 2014. ANNIS3: A new architecture for generic corpus query and visualization. *Literary and Linguistic Computing*.

Andreas Mengel and Wolfgang Lezius. 2000. An XML-based Representation Format for Syntactically Annotated Corpora. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In

[10] http://ipython.org/notebook.html

[11] https://github.com/cjdrake/ipython-magic

Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus technology and language pedagogy: New resources, new tools, new methods*, pages 197–214. Peter Lang.

Arne Neumann, Nancy Ide, and Manfred Stede. 2013. Importing MASC into the ANNIS linguistic database: A case study of mapping GrAF. In *Proceedings of the Seventh Linguistic Annotation Workshop (LAW)*, pages 98–102. Association for Computational Linguistics.

Michael O'Donnell. 2000. RSTTool 2.4: a markup tool for Rhetorical Structure Theory. In *Proceedings of the 1st International Conference on Natural Language Generation (INLG 2000)*, pages 253–256. Association for Computational Linguistics.

Fernando Pérez and Brian E. Granger. 2007. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of LREC 2008*.

Thomas Schmidt. 2004. Transcribing and annotating spoken language with EXMARaLDA. In *Proceedings of the LREC-Workshop on XML based richly annotated corpora, Lisbon*, pages 69–74.

Manfred Stede and Silvan Heintze. 2004. Machine-assisted rhetorical structure annotation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 425. Association for Computational Linguistics.

Manfred Stede and Arne Neumann. 2014. Potsdam Commentary Corpus 2.0: Annotation for Discourse Research. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

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 Session at EACL 2012*, Avignon, France. Association for Computational Linguistics.

Maarten van Gompel and Martin Reynaert. 2013. FoLiA: A practical XML Format for Linguistic Annotation-a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3:63–81.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *ACL (Conference System Demonstrations)*, pages 1–6.

Amir Zeldes, Florian Zipser, and Arne Neumann. 2013. PAULA XML Documentation: Format Version 1.1. Research Report, hal-00783716, https://hal.inria.fr/hal-00783716.

Florian Zipser, Laurent Romary, et al. 2010. A model oriented approach to the mapping of annotation formats using standards. In *Workshop on Language Resource and Language Technology Standards, LREC 2010*.

Florian Zipser, Mario Frank, and Jakob Schmolling. 2014. Merging data, the essence of creation of multi-layer corpora. In *Postersession der Sektion Computerlingustik auf der 36. Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft (DGfS)*.