

(GraPAT の紹介であって
取扱説明書ではない、)

“class-room-annotation”

はチェックしてやるべし。

GraPAT: a Tool for Graph Annotations

Jonathan Sonntag, Manfred Stede

Applied Computational Linguistics

EB Cognitive Science

University of Potsdam / Germany

sonntag@uni-potsdam.de, stede@uni-potsdam.de

sec2: 既存の1-1に新しい3要素

sec3: GraPAT について

sec4: 他のツールとの比較

Abstract

We introduce GraPAT, a web-based annotation tool for building graph structures over text. Graphs have been demonstrated to be relevant in a variety of quite diverse annotation efforts and in different NLP applications, and they serve to model annotators' intuitions quite closely. In particular, in this paper we discuss the implementation of graph annotations for sentiment analysis, argumentation structure, and rhetorical text structures. All of these scenarios can create certain problems for existing annotation tools, and we show how GraPAT can help to overcome such difficulties.

Keywords: Annotation, Graph structures, Sentiment

1. Introduction

We present a new tool for the annotation of graphs upon text. GraPAT (**Graph-based Potsdam Annotation Tool**) is web-based and provides annotators with a natural visualisation of their annotations, and thus supports the intuitions of annotators to follow graph-based annotation schemes.

The paper is structured as follows: In Section 2., we provide the motivation for building this tool by discussing three use cases, where existing tools do not have all the desired functionality. Then, Section 3. describes the design decisions and the implementation of GraPAT, and Section 4. gives comparisons to related work, i.e., to similar annotation tools.

2. Use Cases

The annotation tool has been developed for the annotation of graph structures upon text. Graph structures are useful in annotation, since a) the data to be annotated can be enriched by most types of automatic analysis, i.e. dependency parsing, coreference analysis, etc., and b), of course, the annotation itself is more expressive than token based annotation. It is also of importance to display these graph structures explicitly in order to help annotators follow annotation guidelines.

We now describe three different annotation efforts which lead to the development of the tool. At present, our central use case is sentiment analysis, so we discuss it in most detail; the two others are rhetorical text structure, and argumentation structure.

2.1. Sentiment analysis

The annotation of sentiment on a sub-sentence granularity typically involves the assignment of *sources* or *opinion holders*, *targets* and words which induce a sentiment (Wiebe et al., 2005; Clematide et al., 2012). A source is an entity expressing a polarity towards something, which is the target. The inducing words are then a label on the relation between source and target.

Prominent sentiment annotation guidelines (such as the guidelines of the MPQA corpus (Wiebe et al., 2005) or (Wilson, 2008)) use frames to model sentiment. It is worth

noting that these frame-based representations can be interpreted as graphs as well. A requirement for the interpretation of a frame as a relation, and thus, a graph for all frames, is a given target which, in MPQA, was only annotated for a few instances.

“Sentiment” in newspaper articles is traditionally hard to annotate and even to interpret (Balahur and Steinberger, 2009) because the subjectivity is much less clear than in other genres such as product reviews, and unclear assignments of sources and targets creates confusion for the annotators.

Larger annotation efforts such as Seki et al. (2007), Seki et al. (2008), Seki et al. (2010) and Wiebe et al. (2005) display major difficulties in annotating sentiment on newspaper articles. Wiebe et al. (2005) report a *kappa*-value of 0.77 for the differentiation between objective and subjective sentences. The text span to which the relation is attributed to, is identified with a *kappa*-value of 0.67. No inter-annotator agreement is reported for the polarity of the frames. For this work, the annotation has been carried out by non-expert annotators and they received a training of about 40 hours. The NTCIR sentiment analysis shared tasks kept their tasks' setup which makes them comparable between different iterations. Different subtasks include the decision on whether a sentence is opinionated or not, which polarity an opinion has, if a sentence is relevant with respect to some predefined question, and, finally, detecting sources and targets of opinions. Seki et al. (2007) describe a pilot annotation task in which the *kappa*-agreement for English is 0.2947 to determine whether sentences are opinionated or not and 0.3380 for agreement on the polarity of sentences. The agreement increases to 0.7309 and 0.7069 respectively for NTCIR-8 (Seki et al., 2010). Although this is a substantial increase, it still means that a lot of disagreement exists, although the annotation was closely monitored by the authors, and the annotation guidelines have been fine-tuned and improved in many iterations. The inter-annotator agreement is also an upper limit for the performance of sentiment analysis tools as measured by the test data. Therefore, increasing inter-annotator agreement for human annotations is vital to improve the performances of sentiment

analysis systems.

Apart from improving annotation guidelines, training and monitoring annotators, one route left for an improvement of inter-annotator agreement is the annotation tool itself. GraPAT thus tries to enhance the annotation process, with the goal of increasing the agreement to higher levels.

- (1) a. We disapprove that, while, sadly, Moscow appreciated it.
- b. To Moscows regret, Washington expressed its resentments as well.

A further advantage of an annotation of sentiment using GraPAT is that the nodes in the annotation area correspond to discourse referents and not only to a surface form which allows for a tighter modeling of sentiment. An annotation of Sentence 1a can be seen in Figure 1 and the continuation of it in Figure 4. As can be seen, old concepts and discourse referents remain and can be further annotated. Sentence 1b illustrates the need for this functionality.

Through this process of creation and update of a graph, synchronised with the sentences responsible for it, we obtain an incremental growth of the graph. This incremental annotation and growths of the annotation graph becomes clear when comparing the additions to the graph through Sentence 1b.

A pilot study for an annotation project using the SALTO tool (Burchardt et al., 2006) showed that the annotation guidelines for the sentiment annotation and the tools workflow and visualisation conflicted with each other. This counter-intuitive behaviour of the tool lead to “wrong” annotations although the intuitions of the annotator turned out to be correct in a discussion with the annotator. The annotator was not able to express his intuitions using SALTO (although, strictly, it would have been possible). During the development of GraPAT, the annotator was confronted with the same problem again and was able to serialise his intuitions.

2.2. Rhetorical text structure

A second use-case for GraPAT is the annotation of text structure, where one popular theory that has been applied to many different sorts of texts is Rhetorical Structure Theory (Mann and Thompson, 1988). RST posits trees as representational structure, and there is a widely-used, dedicated annotation tool for this purpose, RSTTool¹. While in general it works well, it does not allow for handling phenomena of segment embedding, as these violate tree constraints by requiring crossing edges. Embedding can occur with speaker attribution but also for “conventional” RST relations, as the following two examples illustrate.

- (2) There is a need, as the president remarked, to increase our level of confidence.
- (3) Tom decided, even though his mother had advised against it, to purchase the car.

Cases like this cannot be satisfactorily handled with RSTTool. Moreover, some other theories of text structure disagree a priori with the tree constraints that RST assumes,

such as the work around the Discourse GraphBank (Wolf and Gibson, 2005). For analyses of this kind, more versatile tools are needed. GraPAT allows for annotating and representing the graph structures that are required. Concerning RST, our present first version of GraPAT does not match the elegance of creating and displaying “well-behaved” RST structures as realized in RSTTool, but we see an RST-specific extension of GraPAT as a step for future work.

2.3. Argumentation structure

Somewhat similar to the RST discussion, the annotation of argumentation structure according to the schema of Peldszus and Stede (2013) makes requirements that go beyond trees. Discourse segments are being related to each other in terms of argumentative support and attack, which may involve fairly complex configurations. In particular, the schema includes the possibility of arbitrary node creation, as well as “edges on edges”. This is used when one text segment attacks not a different text segment, but a support relation that has been marked between two other segments. In other words, it is not the validity of a statement that is being attacked, but the role of a statement for supporting another one. An illustration of handling such structures is given in Figure 2.

Argumentation structure, in this schema, does not necessarily lead to a complete analysis of a text, since not any segment needs to play a role in the core argumentation. (It may just provide background information, for example.) Therefore, the structures are partial, and this is one more reason why a tool such as RSTTool would be inappropriate.

Peldszus and Stede (2013) also describe a “class-room-annotation” scenario, which requires special attention to inter-annotator agreement measures. They show that the performance of annotators can differ significantly in such scenarios. Since web browser-based annotation tools are ideal for annotation efforts with many annotators, such as class-room-annotations, the clustering and ranking methods described by Peldszus and Stede will be included into the tool, in order to provide an overview of annotators’ performances and to detect outliers (which most probably are wrong annotations).

3. Annotation Tool

The interface of GraPAT is split up in different parts: a menu to save, log in, etc.; an annotation area; and the text area which displays the current sentence, paragraph or text. The annotation area serves to create new nodes, which can be typed (i.e., for argumentation structure: proponent vs. opponent), to connect different nodes with edges, which can also be typed (i.e., for sentiment analysis: negative vs. positive) and to delete nodes and edges. When a node or an edge is being created, a small pop-up window contains annotation choices for the element.

One guiding principle for the development of GraPAT was to automate as much work for the annotators as possible. For sentiment annotation as described in the previous section, for example, GraPAT automatically names the nodes of the annotation depending on which textual part the node belongs to. This approach differs from that of the BRAT tool (see below) by trading variability in the annotation

¹<http://www.wagsoft.com/rsttool>

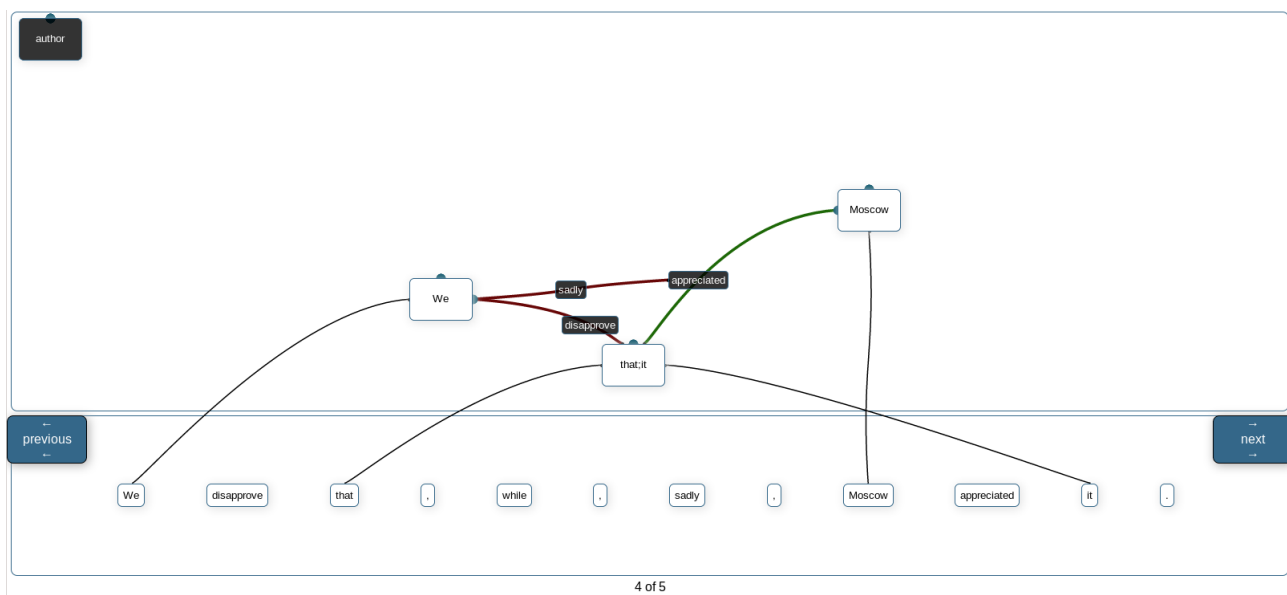


Figure 1: Screenshot from the annotation tool showing a relation between an edge and another relation. Relations with negative polarity are depicted in red (“sadly”, “disapprove”) and with positive polarities in green (“appreciated”).

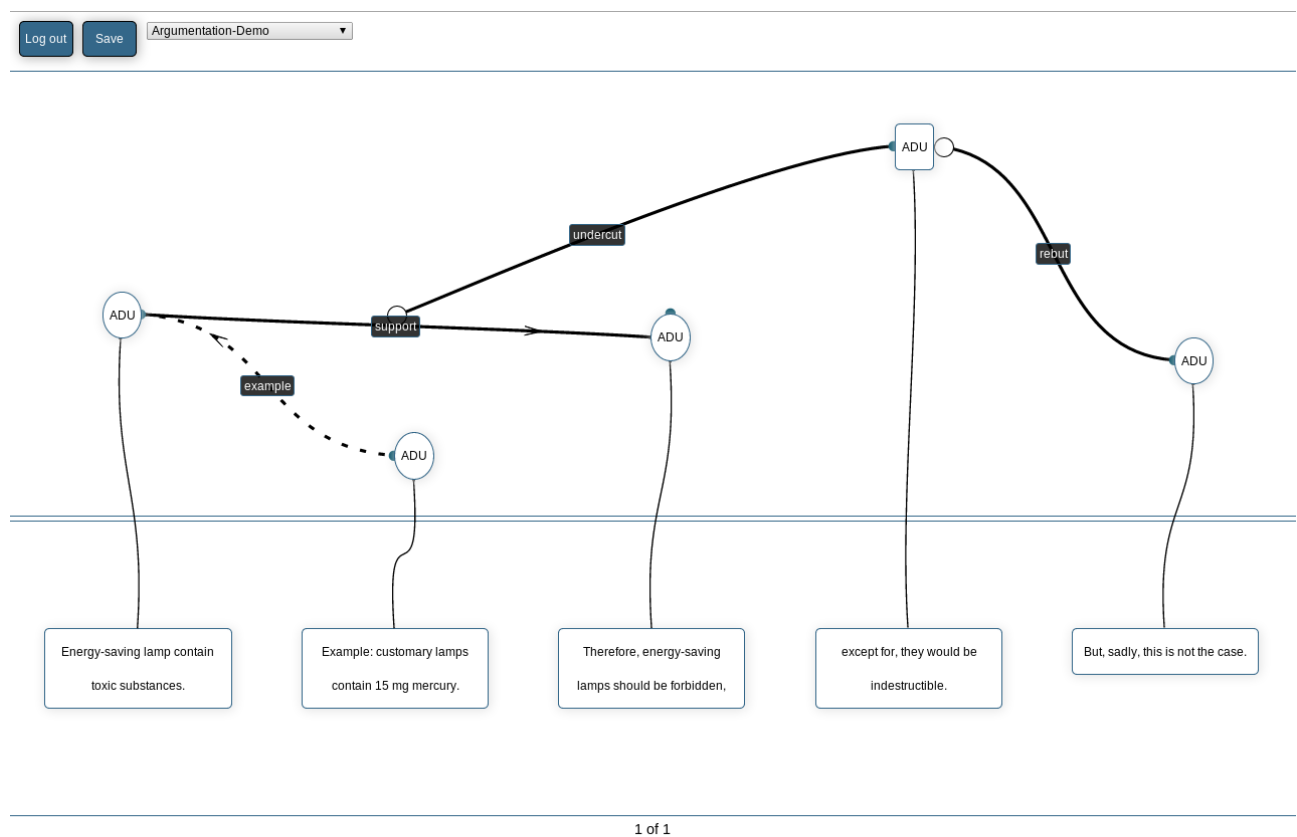


Figure 2: Graph showing how different text segments combine into argumentation units with different relations .

scheme against annotation speed and comfort. In our view, both issues – variability and annotation speed/comfort – are equally important. To achieve both, task-specific tools can be better-suited than general-purpose tools designed for a wide variety of annotation tasks. Task-specific tools (in the spirit of RSTTool), however, are relatively rare, and to this end we see GraPAT as a contribution.

The graphs that can be described by GraPAT are directed (weighted) multigraphs with the ability to create “edges on edges”. Since this is not a standard concept of graphs, the mathematical representation for edges on edges is that the abstract weight² of an edge is a tuple containing an attribute

²An abstract weight can be a rational number, an integer, a label or any other mathematical object.

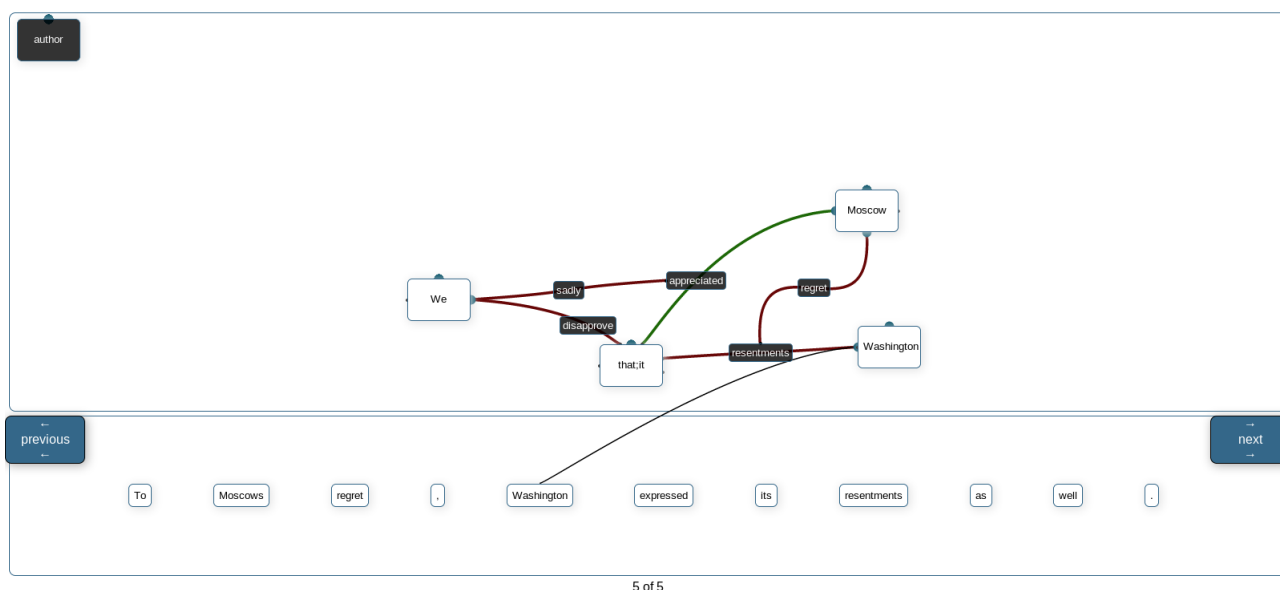


Figure 4: A screenshot showing the continued annotation of the sentences from a text. The discourse referents from previous sentences are still visible as are their relations.

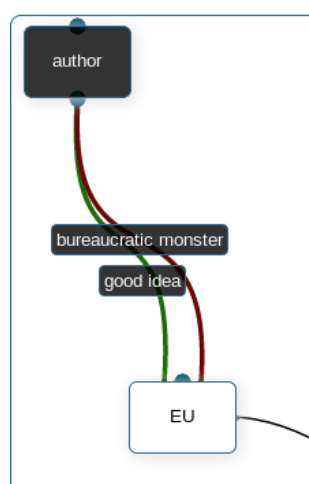


Figure 3: Part of a graph showing multiple edges between nodes. Following their polarities, the edge labeled “good idea” is painted in green and the edge “bureaucratic monster” in red.

value matrix and a node which represents the edge.

Our annotation tool is web-browser-based and thus does not require any installation on the annotators side who often do not have a sufficient technical background to do complicated installations. GraPAT relies on the graph library “jsPlumb”³ for JavaScript, which is publicly available. To avoid heavy load on the server side, most of the implementation relies on client side code using JQuery and JavaScript, and the server controls the data and the results using Java Servlets and JSP. The annotations are saved into a MySQL database in the back-end and can be exported from an administration page at the front-end.

³<http://jsplumbtoolkit.com/home/jquery.html>

4. Related Work

A tool similar to ours is WebAnno (Yimam et al., 2013), which is also web-based and provides the possibility to annotate text based on tokens. A major focus of their work is the management of annotators, which includes functions to monitor the annotation progress, and the measurement of inter-annotator agreement with Kappa as well as a function to define annotation schemes individually.

WebAnno is based on “BRAT” (Stenetorp et al., 2012), a web-based tool for the annotation of NLP phenomena. The authors provide a user-friendly tool without any installation efforts for the annotators, and rich visualisation is possible. On top of its annotation capacities, BRAT provides a search function to search for specific keywords or even for relations between text spans. BRAT also features a semi-automatic annotation mode for semantic class annotation, which proposes results to the annotators and thereby reduces the times annotators spent on a decision.

Just as WebAnno, BRAT can be configured flexibly for annotation schemes, but it does not allow for an arbitrary creation of nodes, which are related but not equal to text passages. For this reason, neither BRAT nor WebAnno are fully suitable for our purposes. They do not display and annotate graph structures intuitively; annotators cannot create nodes where needed, and the visualisation of arcs between nodes is only suitable for dependency-like graphs – which is not the case in the scenarios described in Section 2..

In contrast to BRAT and WebAnno, the SALTO tool (Burcharth et al., 2006) is, in principle, capable of displaying graph-structures, including the arbitrary creation of nodes, and it also allows for the creation of new nodes that are independent of text passages. SALTO was created for the annotation of semantic roles. Since SALTO is not web-based, annotators need to install and configure the tool. Although the installation is not very complicated, it still poses an obstacle to non-tech-savvy annotators. Additionally, the annotation process is not as intuitive as, for example, in

BRAT, and the visualisation is lacking node and edge attribute representations.

5. Conclusions and Outlook

We presented a tool for the annotation of graphs (GraPAT) over text or other graphs which is web-based and intuitive to use. It supports the annotator by automatic enrichments of the annotation whenever such generic mechanisms are applicable. While this leads to faster and more precise annotations, it also reduces the variability in terms of annotation schemes and it requires effort to apply other annotation schemes. Therefore, further development will include reducing the effort for researchers to create custom annotation schemes.

Additionally, we will improve the performance of GraPAT for large graphs which is suboptimal at this point. Finally, we want to include inter-annotator agreement measures including complex measures suited for class-room-annotations.

Further development will include limited functions to manage annotators. Additionally, we want to enable users to specify their own annotation schemes more easily. Yet, it is not our goal to create an alternative to WebAnno or BRAT, but to provide a light-weight tool specifically for annotation projects which require graphs.

The tool is available for download on our project page⁴.

Acknowledgements

Parts of this research belong to the project e-Identity which is funded by the Federal Ministry of Education and Research (BMBF) under grant agreement 01UG1234.

6. References

- Balahur, A. and Steinberger, R. (2009). Rethinking Sentiment Analysis in the News: from Theory to Practice and back. *Proceeding of WOMSA*.
- Burchardt, A., Erk, K., Frank, A., Kowalski, A., Pado, S., and Pinkal, M. (2006). Salto—a versatile multi-level annotation tool. In *Proceedings of LREC 2006*, pages 517–520.
- Clematide, S., Gindl, S., Klenner, M., Petrakis, S., Remus, R., Ruppenhofer, J., Waltinger, U., and Wiegand, M. (2012). MLSA — A Multi-layered Reference Corpus for German Sentiment Analysis. In Chair), N. C. C., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May. European Language Resources Association (ELRA).
- Mann, W. and Thompson, S. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *TEXT*, 8:243–281.
- Peldszus, A. and Stede, M. (2013). Ranking the annotators: An agreement study on argumentation structure. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 196–204, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Seki, Y., Evans, D., Ku, L.-W., Chen, H.-H., Kando, N., and Lin, C.-Y. (2007). Overview of opinion analysis pilot task at ntcir-6. In *Proceedings of NTCIR-6 Workshop Meeting*, pages 265–278.
- Seki, Y., Evans, D., Ku, L.-W., Sun, L., Chen, H.-H., Kando, N., and Lin, C.-Y. (2008). Overview of multilingual opinion analysis task at NTCIR-7. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, pages 185–203.
- Seki, Y., Ku, L.-W., Sun, L., Chen, H.-H., and Kando, N. (2010). Overview of Multilingual Opinion Analysis Task at NTCIR-8: A Step Toward Cross Lingual Opinion Analysis. In *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pages 209–220.
- Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (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*, pages 102–107, Avignon, France, April. Association for Computational Linguistics.
- Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.
- Wilson, T. A. (2008). Fine-grained subjectivity and sentiment analysis: Recognizing the intensity, polarity, and attitudes of private states. June.
- Wolf, F. and Gibson, E. (2005). Representing discourse coherence: a corpus-based study. *Computational Linguistics*, 31(2):249–287.
- Yimam, S. M., Gurevych, I., Eckart de Castilho, R., and Biemann, C. (2013). WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria, August. Association for Computational Linguistics.

⁴<http://www.ling.uni-potsdam.de/acl-lab/Eidentity/main.html>