WebAnno: a flexible, web-based annotation tool for CLARIN

Richard Eckart de Castilho1  Chris Biemann2  Iryna Gurevych1,3  Seid Muhie Yimam2  
(1) Ubiquitous Knowledge Processing Lab (UKP-TUDA)  
Dept. of Computer Science, Technische Universität Darmstadt  
(2) FG Language Technology, Dept. of Computer Science, Technische Universität Darmstadt  
(3) Ubiquitous Knowledge Processing Lab (UKP-DIPF)  
German Institute for Educational Research and Educational Information  
http://www.{lt,ukp}.tu-darmstadt.de

Keywords: linguistic annotation, annotation tool, web application, text annotation, software

1. Introduction
We present WebAnno, a web-based annotation tool suitable for a wide range of text annotation tasks. The development of the tool was driven by the requirements of the CLARIN community, and the tool interacts with the CLARIN infrastructure. The ability to host multiple annotation projects being in parallel – yet isolated from each other – on a single installation of WebAnno makes it particularly attractive for research centers. The ability to fully configure projects via a web interface also enables non-technical staff to create and administer annotation projects. Further, it supports distributed teams of annotators, who are able to work remotely without having to install the software locally.

2. Related work
In this section, we shortly summarize related tools and contrast them to WebAnno. We distinguish between tools supporting distributed annotation, meaning that each team member works on their own annotation set, and collaborative annotation, meaning that all members of the team work on the same annotation set. Collaborative annotation can help better distributing the workload within the team, yet quality assurance is difficult – i.e., the annotation quality cannot be measured via inter-annotator agreement because annotations are not recorded for each annotator separately.

GATE Teamware (Bontcheva et al., 2013) is an annotation tool for distributed annotation teams. The management and monitoring user interfaces are web-based. Yet, contrary to WebAnno, annotations are done using a locally installed software. GATE Teamware allows the definition of complex annotation workflows that mix automatic analysis steps with manual annotation steps, e.g. to automatically annotate a corpus and then have it corrected or augmented by the annotation team.

The brat rapid annotation tool (Stenetorp et al., 2012) is another web-based annotation tool. Contrary to GATE Teamware, annotations can be made in the browser and it does not require the annotators to install any software locally. However, the configuration is mostly done through files, i.e. not web-based. Moreover, annotations are done collaboratively. If an annotator creates, modifies, or deletes an annotation, this change is immediately visible to all other annotators working on the same document.

The lack of a purely web-based generic annotation tool supporting distributed annotation spurred the development of WebAnno.

3. WebAnno
In this section, we describe the functionalities of WebAnno version 2.0 (June 2014), see also (Yimam et al., 2014). As in its previous version (Yimam et al., 2013), WebAnno supports a range of pre-defined annotation layers, such as part of speech, lemmata, named entities, dependency relations, and coreference chains. The new version additionally allows adding and configuring custom annotation layers as required for the annotation task at hand. WebAnno supports three basic annotation concepts: spans, relations between spans, and chains connecting sets of spans (Figure 1).

The complete process of setting up an annotation project, importing documents, configuring custom annotation types, distributing the workload, etc. can all be conveniently performed via a browser-based user interface. Import/export support for different corpus formats make WebAnno interoperable with several other platforms, including the CLARIN WebLicht (Hinrichs et al., 2010) via TCF support. Support for additional corpus formats, e.g. TEL, can be plugged in as reading and writing components compatible with the the DKPro Core component collection (Eckart de Castilho and Gurevych, 2014).

WebAnno offers dedicated support for specific types of annotation projects undertaken by an annotation team in which each member assumes one or more roles.
Roles  The available roles are:

- **Project manager** – configures the project, sets up annotation types, imports the documents in the project configuration interface, and assigns the workload to the annotators in the team in the monitoring interface.
- **Annotators** – create annotations on those documents assigned to them in the annotation interface. They can only see their own annotations and work in isolation from each other.
- **Curator** – reviews the annotations produced by the annotation team via the curation interface and merges them into a final result. A curator can also review the current state of the project in the monitoring interface.

Depending on the role, different components of WebAnno are accessible to the user, for example, annotators cannot change the project configuration.

Annotation interface  WebAnno offers different user interfaces for performing annotations. These depend on the project type:

- **Annotation project** (Figure 2) – a classic annotation project in which the annotation team creates new annotations. The whole screen is used by an annotation editor panel showing the document being annotated. It is possible to work on externally pre-annotated documents, which can be edited by the annotators.
- **Correction project** – in a correction project, the team reviews and corrects or augments annotations that are already present, e.g. as the result of an externally performed automatic annotation procedure. In this mode, the screen is horizontally split into an annotation panel and a suggestion panel. The externally created annotations are displayed in the suggestion panel and can be accepted (copied to the annotation panel) with a single click. The annotation panel then allows editing annotations, but also adding new annotations. In this way, we ensure that all pre-existing (automatically created) annotations are verified manually.
- **Automation project** (Figure 3) – an automation project is a combination of annotation and correction. It uses machine learning capabilities built into WebAnno to automatically suggest annotations in a suggestion panel. This mode can speed up annotation as the annotator can rapidly accept or reject suggestions made by the system that immediately learns from provided annotations. Automatic suggestions are currently only supported for span annotations.

Curation interface  (Figure 4) Curation is supported through a dedicated user interface. The system compares the annotations produced by each member of the annotation team sentence by sentence. If a difference between the annotations is detected, the sentence is highlighted in a sentence overview. Clicking on a sentence opens a detailed comparison view with an annotation panel in the upper part and annotation views below showing the sentence with annotations of each of the annotators. All annotations the team members agreed upon are automatically merged into the annotation panel. The remainder must be manually reviewed and merged by the curator.

Monitoring interface  (Figure 5) Through its browser-based interface, WebAnno supports annotation projects involving a distributed annotation team where annotators can work in parallel, yet isolated from each other. The quality of the annotations produced by the team can be determined based on the inter-annotator agreement. Through the monitoring interface in WebAnno, project managers and curators get an overview of the progress in the annotation projects...
and of the inter-annotator agreement. The interface also allows to distribute the workload in the team by assigning documents to annotators. Furthermore, the monitoring interface provides automation process feedback such as status of the automation, training error and F-measure values.

4. Relation to CLARIN

Made for CLARIN WebAnno was created to meet the requirements on an annotation tool in the context of CLARIN, in particular of the CLARIN-D F-AG 7. However, it was designed and implemented as a generic tool applicable to a wide range of annotation tasks. As such, it has already been used successfully by CLARIN researchers for the preparation of a new dataset for German Named Entity Recognition (Benikova et al., 2014) and for the semantic annotation of the Danish CLARIN reference corpus (Pedersen et al., 2014). By carrying out annotation projects already during the development of WebAnno, we made sure to incorporate early feedback by users.

New impulses to TCF To our knowledge, WebAnno is one of the first, if not even the first annotation tool to support TCF. The TCF format was designed for the interchange of annotated corpus data between web-services in CLARIN WebLicht, where each service consumes the output of previous services and adds new layers of annotation on top. Using TCF in an annotation tool is a different use-case, as annotations are not only added, but also updated or deleted by annotators. In collaboration with the maintainers of the TCF API wlfxb\(^1\), we drafted an extension to the API to support this use-case. The extension permits the preservation of arbitrary XML elements in the TCF stream (even elements that are not part of the TCF specification) and rewriting existing annotation layers, e.g. because they have been edited by the annotation team. The preservation of arbitrary XML elements is important to WebAnno and TCF users in this context, because it allows them to quickly correlate annotations edited in WebAnno or automatically created in WebLicht with extra project-specific annotations that are not (yet) part of the TCF specification.

5. Conclusion

We developed the WebAnno annotation tool driven by requirements from the CLARIN community. It was used in producing new language resources now offered by CLARIN and spurred discussions around the TCF format, being one of the first annotation tools supporting this format. WebAnno is suited for a wide range of annotation tasks, easily configurable via web interfaces and provided as open source software\(^2\) under a permissive license.

In the future, we will make further refinements to WebAnno, driven by the needs of the CLARIN community.

6. Acknowledgements

The work presented in this paper was funded by a German BMBF grant to the CLARIN-D project, the Hessian LOEWE research excellence program as part of the research center “Digital Humanities” and by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. 1/82806.

7. References


\(^1\)https://github.com/weblicht/wlfxb
\(^2\)http://webanno.googlecode.com