WAT-SL: A Customizable Web Annotation Tool for Segment Labeling

Johannes Kiesel and **Henning Wachsmuth** and **Khalid Al-Khatib** and **Benno Stein**
Faculty of Media
Bauhaus-Universität Weimar
99423 Weimar, Germany
`<first name>.<last name>@uni-weimar.de`

Abstract

A frequent type of annotations in text corpora are labeled text segments. General-purpose annotation tools tend to be overly comprehensive, often making the annotation process slower and more error-prone. We present *WAT-SL*, a new web-based tool that is dedicated to segment labeling and highly customizable to the labeling task at hand. We outline its main features and exemplify how we used it for a crowdsourced corpus with labeled argument units.

1 Introduction

Human-annotated corpora are essential for the development and evaluation of natural language processing methods. However, creating such corpora is expensive and time-consuming. While remote annotation processes on the web such as crowdsourcing provide a way to obtain numerous annotations in short time, the bottleneck often lies in the used annotation tool and the required training of annotators. In particular, most popular annotation tools aim to be general purpose, such as the built-in editors of GATE (Cunningham, 2002) or web-based tools like BRAT (Stenetorp et al., 2012) and WebAnno (Yimam et al., 2014). Their comprehensiveness comes at the cost of higher interface complexity, which often also decreases the readability of the texts to be annotated.

Most of the typical annotation work is very focused, though, requiring only few annotation functionalities. An example is *segment labeling*, i.e., the assignment of one predefined label to each segment of a given text. Such labels may, e.g., capture clause-level argument units, sentence-level sentiment, or paragraph-level information extraction events. We argue that, for segment labeling, a dedicated tool is favorable in order to speed up the annotator training and the annotation process. While crowdsourcing platforms such as *mturk.com* or *crowdflower.com* follow a similar approach, their interfaces are either hardly customizable at all or need to be implemented from scratch.

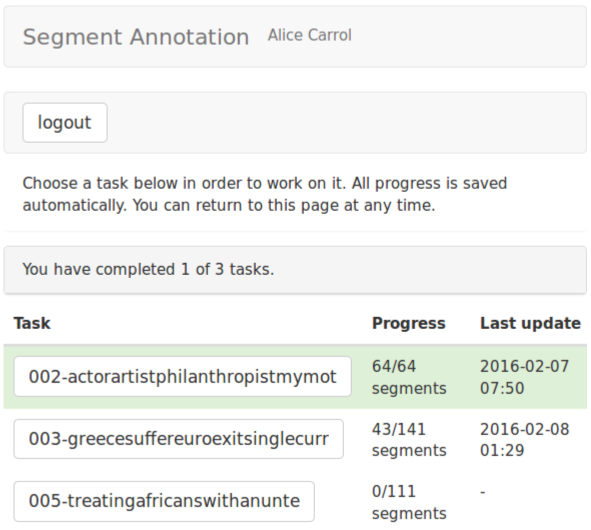
This paper presents *WAT-SL* (Web Annotation Tool for Segment Labeling), an open-source web-based annotation tool dedicated to segment labeling.[1] WAT-SL provides all functionalities to efficiently run and manage segment labeling projects. Its self-descriptive annotation interface requires only a web browser, making it particularly convenient for remote annotation processes. The interface can be easily tailored to the requirements of the project using standard web technologies in order to focus on the specific segment labels at hand and to match the layout expectations of the annotators. At the same time, it ensures that the texts to be labeled remain readable during the whole annotation process. This process is server-based and preemptable at any point. The annotator's progress can be constantly monitored, as all relevant interactions of the annotators are logged in a simple key-value based plain text format.

In Section 2, we detail the main functionalities of WAT-SL, and we explain its general usage. Section 3 then outlines how we customized and used WAT-SL ourselves in previous work to label over 35,000 argumentative segments in a corpus with 300 news editorials (Al-Khatib et al., 2016).

2 Segment Labeling with WAT-SL

WAT-SL is a ready-to-use and easily customizable web-based annotation tool that is dedicated to segment labeling and that puts the focus on easy usage for all involved parties: annotators, annotation curators, and annotation project organizers.

[1] WAT-SL is available open source under a MIT license at: `https://github.com/webis-de/wat`


Segment Annotation Alice Carrol

logout

Choose a task below in order to work on it. All progress is saved automatically. You can return to this page at any time.

You have completed 1 of 3 tasks.

Task	Progress	Last update
002-actorartistphilanthropistmymot	64/64 segments	2016-02-07 07:50
003-greecesuffereuroexitsinglecurr	43/141 segments	2016-02-08 01:29
005-treatingafricanswithanunte	0/111 segments	-

Figure 1: Screenshot of an exemplary task selection page in WAT-SL, listing three assigned tasks.

In WAT-SL, the annotation process is split into *tasks*, usually corresponding to single texts. Together, these tasks form a *project*.

2.1 Annotating Segments

The annotation interface is designed with a focus on easy and efficient usage. It can be accessed with any modern web browser. In order to start, annotators require only a login and a password.

When logging in, an annotator sees the *task selection page* that lists all assigned tasks, including the annotator's current progress in terms of the number of labeled segments (Figure 1). Completed tasks are marked green, and the web page automatically scrolls down to the first uncompleted task. This allows annotators to seamless interrupt and continue the annotation process.

After selecting a task to work on, the annotator sees the main *annotation interface* (Figure 2). The design of the interface seeks for clarity and self-descriptiveness, following the templates of today's most popular framework for responsive web sites, *Bootstrap*. As a result, we expect that many annotators will feel familiar with the style.

The central panel of the annotation interface shows the text to be labeled and its title. One design objective was to obtain a non-intrusive annotation interface that remains close to just displaying the text in order to maximize readability. As shown in Figure 2, we decided to indicate segments only by a shaded background and a small button at the end. To ensure a natural text flow, line breaks are possible within segments. The button reveals a menu for selecting the label. When moving the mouse cursor over a label, the label description is displayed to prevent a faulty selection. Once a segment is labeled, its background color changes, and the button displays an abbreviation of the respective label. To assist the annotators in forming a mental model of the annotation interface, the background colors of labeled segments match the label colors in the menu. All labels are saved automatically, avoiding any data loss in case of power outages, connection issues, or similar.

In some cases, texts might be over-segmented, for example due to an automatic segmentation. If this is the case, WAT-SL allows annotators to mark a segment as being continued in the next segment. The interface will then visually connect these segments (cf. the buttons showing "->" in Figure 2).

Finally, the annotation interface includes a text box for leaving comments to the project organizers. To simplify the formulation of comments, each segment is numbered, with the number being shown when the mouse cursor is moved over it.

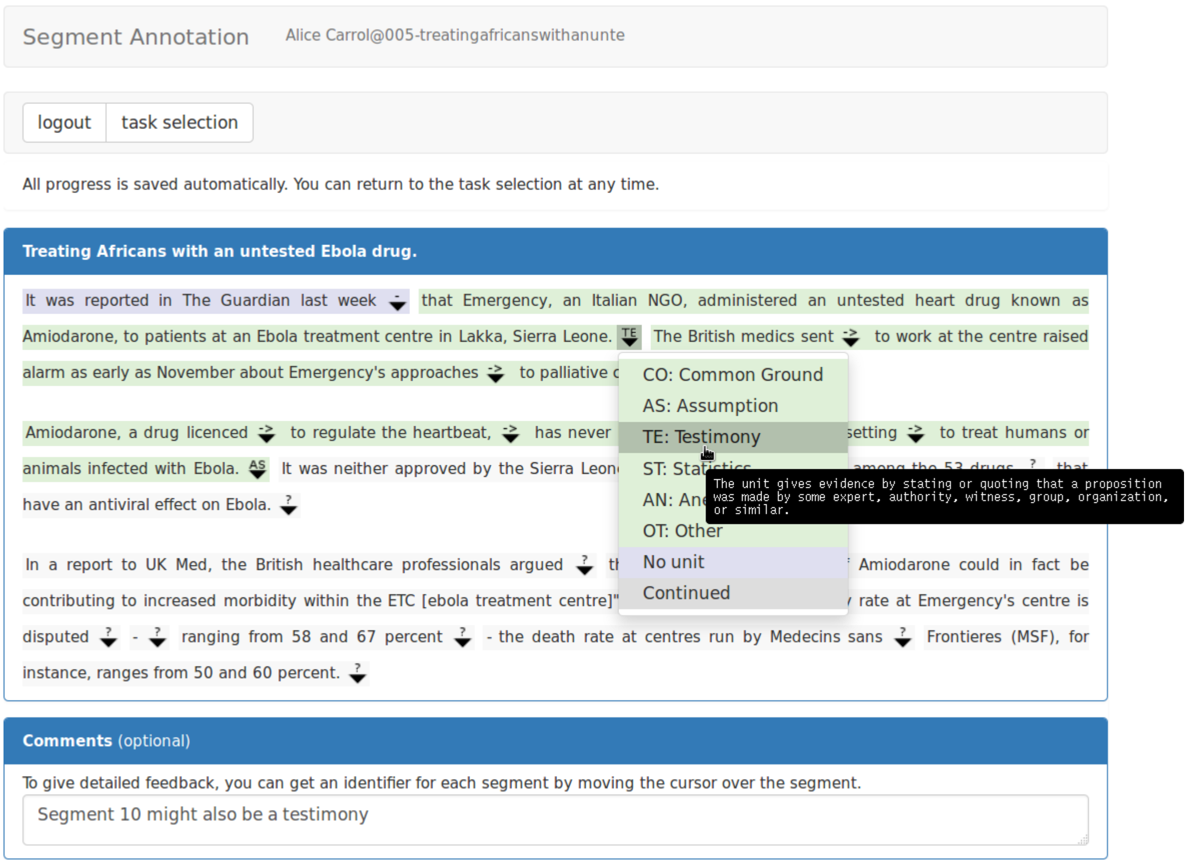
2.2 Curating Annotations

After an annotation process is completed, a curation phase usually follows where the annotations of different annotators are consolidated into one. The WAT-SL *curation interface* enables an efficient curation by mimicking the annotation interface with three adjustments (Figure 3): First, segments for which the majority of annotators agreed on a label are pre-labeled accordingly. Second, the menu shows for each label how many annotators chose it. And third, the label description shows (anonymized) which annotator chose the label, so that curators can interpret each label in its context.

The curation may be accessed under the same URL as the annotation in order to allow annotators of some tasks being curators of other tasks.

2.3 Running an Annotation Project

WAT-SL is a platform-independent and easily deployable standalone Java application, with few configurations stored in a simple "key = value" file. Among others, annotators are managed in this file by assigning a login, a password, and a set of tasks to each of them. For each task, the organizer of an annotation project creates a directory (see below). WAT-SL uses the directory name as the task name in all occasions. Once the Java archive file we provide is then executed, it reads all configurations and starts a server. The server is immediately ready to accept requests from the annotators.


Segment Annotation Alice Carrol@005-treatingafricanswithanunte

logout | task selection

All progress is saved automatically. You can return to the task selection at any time.

Treating Africans with an untested Ebola drug.

It was reported in The Guardian last week | that Emergency, an Italian NGO, administered an untested heart drug known as Amiodarone, to patients at an Ebola treatment centre in Lakka, Sierra Leone. TE | The British medics sent | to work at the centre raised alarm as early as November about Emergency's approaches | to palliative c

Amiodarone, a drug licenced | to regulate the heartbeat, | has never | setting | to treat humans or animals infected with Ebola. AS | It was neither approved by the Sierra Leon | among the 53 drugs ? | that have an antiviral effect on Ebola. ?

In a report to UK Med, the British healthcare professionals argued ? | th | Amiodarone could in fact be contributing to increased morbidity within the ETC [ebola treatment centre]" | rate at Emergency's centre is disputed ? | - ? | ranging from 58 and 67 percent ? | - the death rate at centres run by Medecins sans ? | Frontieres (MSF), for instance, ranges from 50 and 60 percent. ?

CO: Common Ground
AS: Assumption
TE: Testimony
ST: Statistics
AN: Anecdote
OT: Other
No unit
Continued

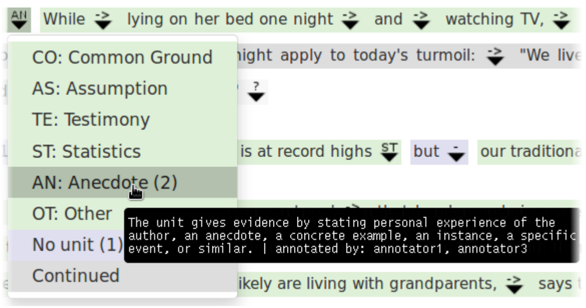
The unit gives evidence by stating or quoting that a proposition was made by some expert, authority, witness, group, organization, or similar.

Comments (optional)

To give detailed feedback, you can get an identifier for each segment by moving the cursor over the segment.

Segment 10 might also be a testimony

Figure 2: Screenshot of the annotation interface of WAT-SL, capturing the annotation of one news editorial within the argumentative segment labeling project described in Section 3. In particular, the screenshot illustrates how the annotator selects a label (*Testimony*) for one segment of the news editorial.


All | While | lying on her bed one night | and | watching TV, | ight apply to today's turmoil: | "We live | ? | is at record highs ST | but | our tradition | ikely are living with grandparents, | says

CO: Common Ground
AS: Assumption
TE: Testimony
ST: Statistics
AN: Anecdote (2)
OT: Other
No unit (1)
Continued

The unit gives evidence by stating personal experience of the author, an anecdote, a concrete example, an instance, a specific event, or similar. | annotated by: annotator1, annotator3

Figure 3: Screenshot of the curation interface, illustrating how a label (*Anecdote*) is selected based on counts of all labels the annotators selected for the respective segment (*Anecdote (2)*, *No unit (1)*).

WAT-SL logs all relevant annotator interactions. Whenever an annotator changes a segment label, the new label is immediately sent to the server. This prevents data loss and allows to monitor the progress. In addition to the new label, WAT-SL logs the current date, and the time offset and IP address of the annotator. With these logs, researchers can analyze the annotators' behavior, e.g., to identify hard cases where annotators needed much time or changed the label multiple times.

Annotating Segments Only few steps are needed to set up the segment labeling tasks: (1) List the labels and their descriptions in the configuration file, and place the button images in the corresponding directory. (2) Set the title displayed above each text (see Figure 2) in a separate configuration file in the respective task directory, or project-wide in the same file as the labels. (3) Finally, put the texts in the task directories. For easy usage, the required text format is as simple as possible: one segment per line and empty lines for paragraph breaks. Optionally, organizers can add Cascading Style Sheet and JavaScript files to customize the interface.

Curating Annotations To curate a task, an organizer duplicates the annotation task and then copies the annotation logs into the new task directory. The organizer then specifies curators for the curation task analog to assigning annotators.

Result In addition to the logs, the web interface also allows the organizer to see the final labels without history in a simple "key = value" format, which is useful when distributing the annotations.

3 Case Study: Labeling Argument Units

WAT-SL was already used successfully in the past, namely, for creating the *Webis-Editorials-16 corpus* with 300 news editorials split into a total of 35,665 segments, each labeled by three annotators and finally curated by the authors (Al-Khatib et al., 2016). In particular, each segment was assigned one of eight labels, where the labels include six types of argument units (e.g., *assumption* and *anecdote*), a label for non-argumentative segments, and a label indicating that a unit is continued in the next segment (see the "->" label in Section 2). On average, each editorial contains 957 tokens in 118 segments.

The annotation project of the Webis-Editorials-16 corpus included one task per editorial. The editorial's text had been pre-segmented with a heuristic unit segmentation algorithm before. This algorithm was tuned towards oversegmenting a text in case of doubt, i.e., to avoid false negatives (segments that should have been split further) at the cost of more false positives (segments that need to be merged). Note that WAT-SL allows fixing such false positives using "->".

For annotation, we hired four workers from the crowdsourcing platform *upwork.com*. Given the segmented editorials, each worker iteratively chose one assigned editorial (see Figure 1), read it completely, and then selected the appropriate label for each segment in the editorial (see Figure 2). This annotation process was repeated for all editorials, with some annotators interrupting their work on an editorial and returning to it later on. All editorials were labeled by three workers, resulting in 106,995 annotations in total. The average time per editorial taken by a worker was ~20 minutes.

To create the final version of the corpus, we curated each editorial using WAT-SL. In particular, we automatically kept all labels with majority agreement, and let one expert decide on the others. Also, difficult cases where annotators tend to disagree were identified in the curation phase.

From the perspective of WAT-SL, the annotation of the Webis-Editorials-16 corpus served as a case study, which provided evidence that the tool is easy to learn and master. From the beginning, the workers used WAT-SL without noteworthy problems, as far as we could see from monitoring the interaction logs. Also, the comment area turned out to be useful, i.e., the workers left several valuable suggestions and questions there.

4 Conclusion and Future Work

This paper has presented WAT-SL, an open-source web annotation tool dedicated to segment labeling. WAT-SL is designed for easily configuration and deployment, while allowing project organizers to tailor its annotation interface to the project needs using standard web technologies. WAT-SL aims for simplicity by featuring a self-explanatory interface that does not distract from the text to be annotated, as well as by always showing the annotation progress at a glance and saving it automatically. The curation of the annotations can be done using exactly the same interface. In case of oversegmented texts, annotators can merge segments. Moreover, all relevant annotator interactions are logged, allowing projects organizers to monitor and analyze the annotation process.

Recently, WAT-SL was used successfully on a corpus with 300 news editorials, where four remote annotators labeled 35,000 argumentative segments. In the future, we plan to create more dedicated annotation tools in the spirit of WAT-SL for other annotation types (e.g., relation identification). In particular, we plan to improve the annotator management functionalities of WAT-SL and reuse them for these other tools.

References

Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. A News Editorial Corpus for Mining Argumentation Strategies. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3433–3443.

Hamish Cunningham. 2002. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36(2):223–254.

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 at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96.