

A Web-based Collaborative Annotation and Consolidation Tool

Tobias Daudert

Insight Centre for Data Analytics
National University of Ireland, Galway
tobias.daudert@insight-centre.org

Abstract

Annotation tools are a valuable asset for the construction of labelled textual datasets. However, they tend to have a rigid structure, closed back-end and front-end, and are built in a non-user-friendly way. These downfalls difficult their use in annotation tasks requiring varied text formats, prevent researchers to optimise the tool to the annotation task, and impede people with little programming knowledge to easily modify the tool rendering it unusable for a large cohort. Targeting these needs, we present a web-based collaborative annotation and consolidation tool (AWOCATo), capable of supporting varied textual formats. AWOCATo is based on three pillars: (1) Simplicity, built with a modular architecture employing easy to use technologies; (2) Flexibility, the JSON configuration file allows an easy adaption to the annotation task; (3) Customizability, parameters such as labels, colours, or consolidation features can be easily customized. These features allow AWOCATo to support a range of tasks and domains, filling the gap left by the absence of annotation tools that can be used by people with and without programming knowledge, including those who wish to easily adapt a tool to less common tasks. AWOCATo is available for download at <https://github.com/TDaudert/AWOCATo>.

Keywords: annotation, consolidation, curation, tool, text, textual annotation, sentiment analysis, classification

1. Introduction

The continuous rise in data enabled new opportunities in computer science (Hey and Trefethen, 2003). Large volumes of data can be used in machine learning approaches for a variety of tasks such as text classification (Khan et al., 2010), image recognition (Joutou and Yanai, 2009), and disease prediction (Chen et al., 2017). Labelled datasets are fundamental for the evaluation of these models and for training on supervised approaches, however, they tend to be sparse (Jeong et al., 2009; He et al., 2007). Annotation is known to be a complex task, involving many people and stages (Finlayson and Erjavec, 2017); as aid, researchers can leverage annotation tools. Albeit facilitating and improving the efficiency of annotation tasks, annotation tools tend to have a rigid structure and a complex interface, thus, requiring a steep learning curve and discouraging the deployment of tailored tasks. In addition, this complexity deters potential users without programming skills to employ such tools in their annotations task.

To fulfil these needs, we present a web-based/online collaborative annotation and consolidation tool (in short, AWOCATo), capable of supporting varied textual formats. AWOCATo’s structure is based on three pillars:

1. **Simplicity** - built with Python and utilising MongoDB¹ as back-end.
2. **Flexibility** - annotation and curation task features are configurable in a JSON² file.
3. **Customizability** - fully adaptable front-end coded with HTML, CSS, and Javascript.

In this paper, we will deconstruct AWOCATo providing details on the back-end architecture, front-end, and, most importantly, the module which contributes the most to this

tool’s flexibility — the JSON configuration file. To demonstrate its usability, section 4. presents an annotation task where AWOCATo was used for a sentiment annotation task.

2. Related Work

The need for annotated corpora has been progressively increasing over the years. Nowadays, we can find a myriad of tools adapted to video (Lin et al., 2003), text (Yimam et al., 2013; Bontcheva et al., 2013), and multimodal data (Kipp, 2001; Wittenburg et al., 2006) annotation. Focusing on the annotation of textual corpora, crowd-sourcing systems (e.g., MTurk³, Figure Eight⁴) are routinely applied for classification tasks while for annotation targeting linguistics (e.g., syntax), recent tools such as LighTAG⁵ offer the management of annotators as well as the integration of evaluation supported by AI-based models as with Daturks⁶ and TagTog⁷. Over the years, annotation tools continued to evolve addressing the needs of the research community. In this section, we expose three downfalls of the currently available annotation tools for textual data.

First, annotation tools tend to be complex. Although certain tasks (e.g., named-entity recognition (NER), part of speech tagging (POS)) can require a more complex interface as utilised in BRAT (Stenetorp et al., 2012), GATE Teamware (Bontcheva et al., 2013), WebAnno (Yimam et al., 2013), and Callisto (Day et al., 2004), this is not the case for simpler tasks such as the sentiment classification or text labelling. The complexity of these tools can make the annotation process overwhelming, thus, they are not suitable for a wide range of annotation tasks. Recent tools have tried to counteract this issue. For example, TagEditor⁸ is an annotation desktop application integrated with

¹<https://www.mongodb.com/>

²<https://www.json.org/>

³<https://www.mturk.com/>

⁴<https://www.figure-eight.com/>

⁵<https://www.lighttag.io>

⁶<https://dataturks.com/>

⁷<https://www.tagtog.net/>

⁸<https://github.com/d5555/TagEditor>

	Web-based	Consolidation/Curation Mode	Customizable*	Programming Knowledge Required **	Open Source
BRAT	Yes	Yes	No	Yes	Yes
Callisto	No	No	Yes	Yes	Yes
CoSACT	Yes	Yes	No	Yes	Yes
Datururks	Yes	Limited	Limited	No	No
Doccano	Yes	No	Limited	No	Yes
GATE	Yes	Yes	Yes	Yes	Yes
LightTAG	Yes	No	Limited	No	No
TagEditor	No	No	Limited	No	Yes
TagTog	Yes	Yes	Limited	No	No
WebAnno	Yes	Yes	Yes	Yes	Yes

*Stated as one of the major tool features. **Medium to High.

Table 1: Annotation tool comparison.

the Python spaCy library⁹ which allows the labelled dataset to be easily applied to machine learning models available with spaCy. It offers flexibility to choose the type of annotation and labels as well as several other options during the annotation (*e.g.*, sentence marking, break line and white space deletion). However, it does not support multiple parallel annotators nor it is deployed in a server, thus, lacking the ability to track the annotator’s progress and to flexibly work on different machines. As it limited to the same local machine, each annotator is also the tool’s manager, hence, responsible for administrating it on their machine. Furthermore, TagEditor offers a rigid interface structure and limited adaption to the annotation purpose. Another example is Doccano (Nakayama et al., 2018), an online annotation tool based on the creation of annotation projects; annotator log-in credentials are associated with a project, thus, securely supporting multiple annotators at the same time. Doccano also allows for the setting of task-specific labels. However, only categorical labels are supported and the customization of these is also limited to annotation tasks with similar label requirements such as NER, sentiment analysis, and translation. In addition, both TagEditor and Doccano do not support the consolidation of the labelled data. CoSACT (Daudert et al., 2019), a recent tool applied in the Social Sentiment Indexes Project (SSIX)¹⁰ and in the SemEval 2017 Task 5 (Cortis et al., 2017), provides a consolidation mode, however, similarly to Doccano and TagEditor, its inflexible structure prevents it from being used to other tasks apart from sentiment analysis. Furthermore, CoSACT was developed in the context of a European project, without aspiration to become a widely used and publicly available annotation tool. This relates to the second identified problem, the rigid structure of the annotation tools which impedes users to adapt them to the needs of their tasks. Although for most tasks an out-of-the-box solution suffices, other users prefer to adapt the tool according to their needs. The third downfall respects the lack of customization available in these tools which tends to be absent or limited to a few design choices. Besides choosing the right tool to com-

ply with the annotation aim, users also need to customize class labels, the color of the buttons, or amend fundamental variables such as the standard deviation for the consolidation, for example. Table 1 presents a comparison between the mentioned tools.

To address the need for a user-friendly annotation tool capable of being utilised by people from a wide range of domains, AWOCATo was developed. This web-based annotation and consolidation tool supports multiple annotators and annotation tasks, which can also be conducted in parallel. Furthermore, its simple architecture based on Python and design in HTML, CSS, and Javascript can be easily customized.

3. Tool Description

AWOCATo is a fully customizable server-based annotation tool only requiring a browser (*e.g.*, Chrome¹¹, Firefox¹²) to run. It was created on a simple architecture incorporating a Python back-end, MongoDB for data storage, HTML/CSS/Javascript front-end, and a JSON file for defining the annotation task and consolidation features. This tool also provides a consolidation/curation mode in which the annotation can be automatically or manually consolidated, if necessary. As AWOCATo is a server-based tool, the user can comply with local general data protection and privacy regulations which is not guaranteed by third-party tools. Furthermore, the user has complete control over AWOCATo and can deploy it to any computing device supporting the required technologies.

The architecture for this tool, presented in figure 1, depicts the front-end, back-end, and feature configuration. To allow for an in-depth view of AWOCATo, below are reported further details on the features, back-end, and front-end. For simplicity, we refer to *user* as the person applying AWOCATo in an annotation task, and the *annotator* as the person performing the annotation task.

3.1. Features

The flexibility provided by the configurable JSON file is the core of AWOCATo. This architecture delivers complete

⁹<https://spacy.io/>

¹⁰<https://ssix-project.eu/>

¹¹<https://www.google.com/chrome/>

¹²<https://www.mozilla.org/firefox/>

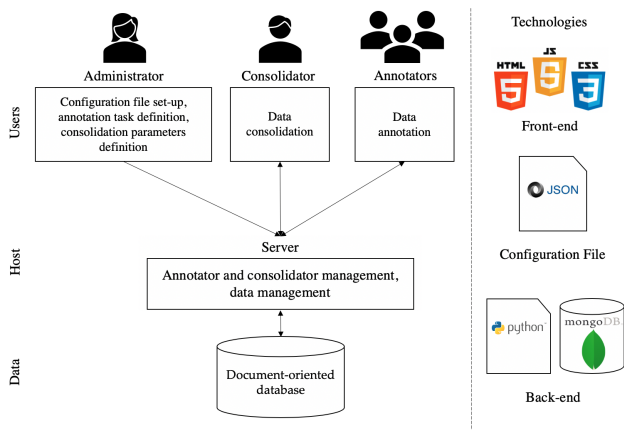


Figure 1: Scheme of the tool (left) and the applied technologies (right).

control to the user allowing AWOCATo to be tailored to the required annotation task in a simple manner. The file is utilised to define:

1. **Data provided and storage format:** The user is able to set the database, the collection, and the fields from MongoDB that are used to provide the data for the annotation. The user is also able to define the batch size for the queried records (*i.e.*, texts for annotation), thus, balancing the number of queries with the size of data provided to the front-end. The annotation is then stored in user-defined fields; thus, the user designs their own structure for the data storage.
2. **Security and access management:** The JSON configuration file also holds the account information for the annotators and consolidators. The access to the tool is secured by a username and password combination defined by the user. During the communication between the front-end and back-end, the exchange of the login credentials is encrypted preventing the loss of confidential information. Furthermore, as AWOCATo is deployed on a server, data privacy and control is guaranteed. AWOCATo also includes features designed to protect the displayed data by impeding the ability to copy and screenshot.
3. **Annotation task requirements/modes:** The user is able to define the parameters necessary for a successful annotation based on the task at hand. For example, the presence of a sentiment scale for a sentiment analysis annotations task. AWOCATo contains by default seven annotation modes: basic sentiment annotation, news sentiment annotation (*i.e.*, integrates the title and content), four configurations for categorical annotation (*e.g.*, binary, ternary), and free-text annotation. In addition, the user can further customize each mode by changing the placement of the buttons and scales, and attributing new colors; figure 4 and 5 represent a ternary categorical annotation example and figure 6 a free-text annotation. Due to AWOCATo's modular

architecture, modifications and additions to one mode are independent from changes to any other mode.

4. **Consolidation settings:** The configuration file also stores the parameters vital for the consolidation mode. As AWOCATo can automatically consolidate annotations, it requires at least two basic settings: the number of required annotations (*i.e.*, three annotators per annotated text), and the stipulated standard deviation (*i.e.*, maximum allowed dispersion of the data). Manual consolidation is performed when at least one of the selected settings is not satisfied; in this scenario the consolidator manually reviews the annotations and attributes a curated score. For specific modes, such as the sentiment one (see figure 2), the user can also define the standard deviation for the relevance annotation or the minimum number of annotators agreeing on the polarity. Switching between the consolidation and the annotation mode simply requires the change of one boolean variable in the configuration file (see *consolidation.mode* in figure 7).

5. **Annotation guidelines:** All annotation tasks require specific guidelines with which the annotator must familiarise themselves. AWOCATo includes a customizable guideline page in HTML. However, to cater to users with limited HTML knowledge, the annotation guidelines can be created, for example, in Google Docs¹³, exported as HTML and stored in a predefined folder. Users with little programming skills will not need to write a single line of code and knowledgeable users can directly write their own HTML guideline page. The configuration file also provides the ability to choose if the guidelines display at the start of each annotation/consolidation session.

As an example, figure 7 shows the structure of the configuration JSON file. In case the user wishes to add new features to the tool, they can simply utilise the same architecture and employ the necessary changes in AWOCATo's back-end and front-end.

3.2. Front-end

AWOCATo's simple aesthetics are designed with HTML, CSS, and Javascript. All these languages are common in the computer science field and easy to use, hence, if the user wishes to update the tool's appearance they can easily do so. Otherwise, when no programming background is present, AWOCATo can be used as-is providing a clean environment for the annotators to work on. This is in line with the fundamentals of providing an easy to use and open source annotation tool for textual data. Incorporated by default in AWOCATo's front-end, the user can find a feature which enables a pop-up window when the annotator does not select an option for the annotation (*e.g.*, sentiment class or score, label). This feature was set-up to motivate the annotator to correctly perform the task.

¹³<https://www.google.com/docs/about/>

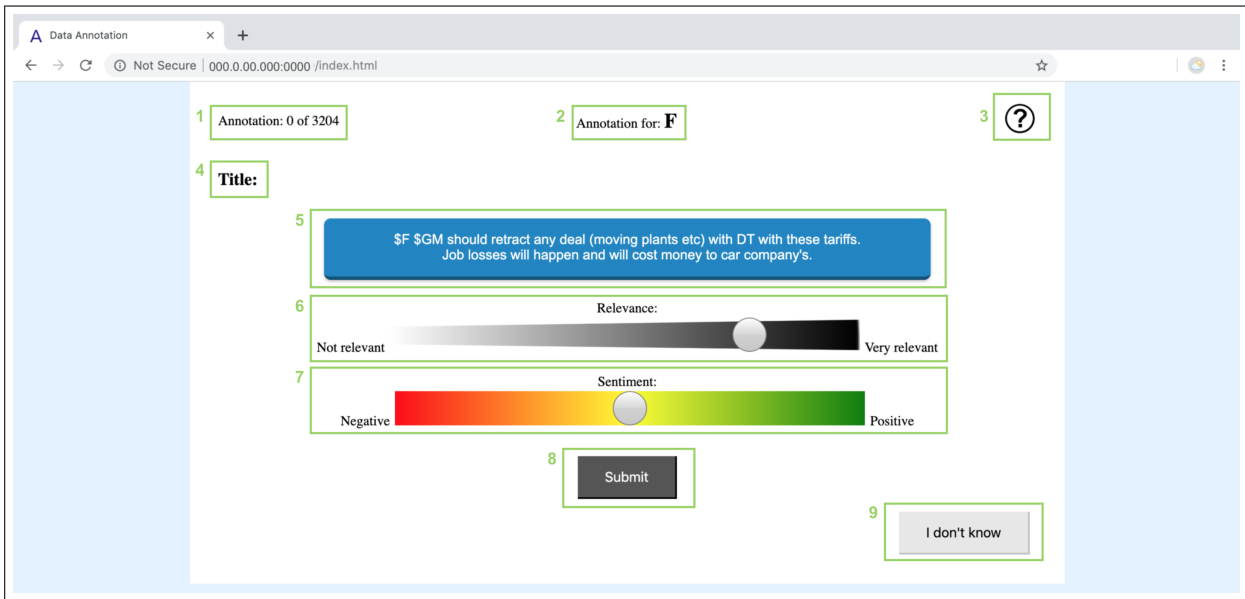


Figure 2: Example of AWOCATo in the annotation mode for sentiment analysis with numbered boxes.

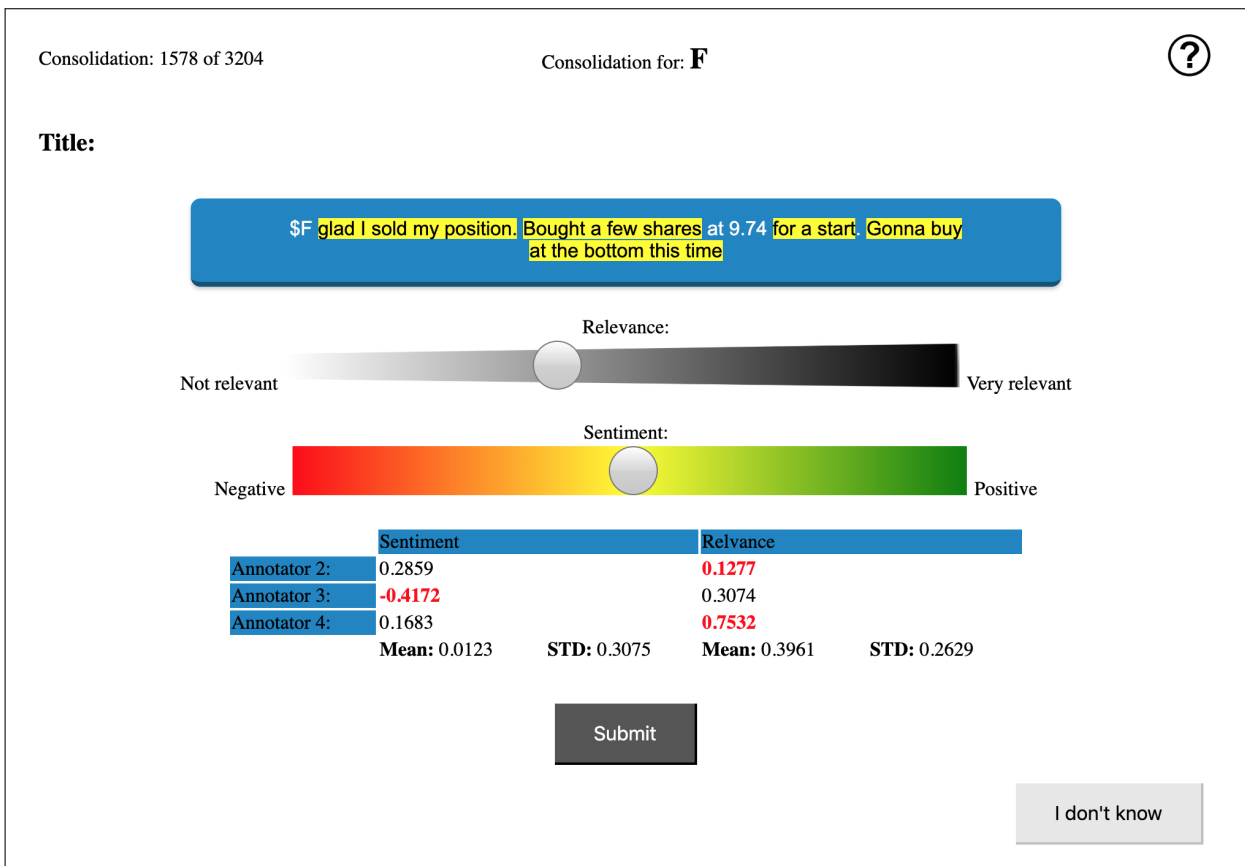


Figure 3: Close-up of the consolidation mode for sentiment analysis in AWOCATo.

3.3. Back-end

AWOCATo is built with the Python programming language. This is the most popular language in computer science¹⁴ partly due to its extensive library support, third-party mod-

ules integration, user-friendly data structures, and simple-to-learn syntax (Oliphant, 2007). The Python back-end establishes the server, implements the desired preparation to the textual data, and secures the storage in a database which can be either deployed on the same machine or external to it.

¹⁴<https://www.statista.com/chart/16567/popular-programming-languages/>

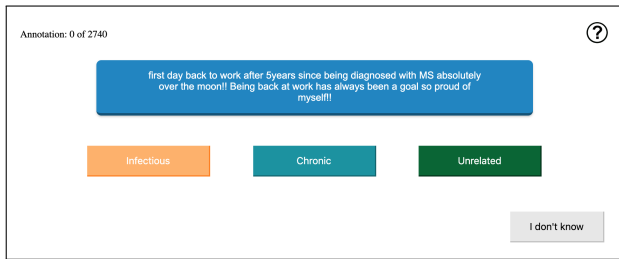


Figure 4: Categorical annotation example.

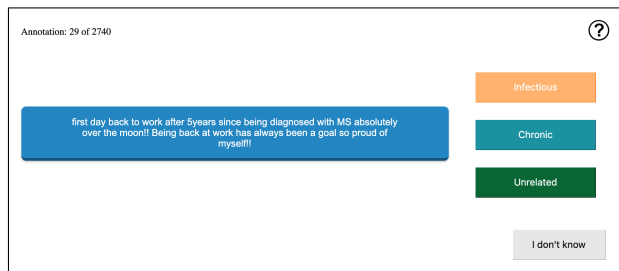


Figure 5: Categorical annotation with side buttons example.

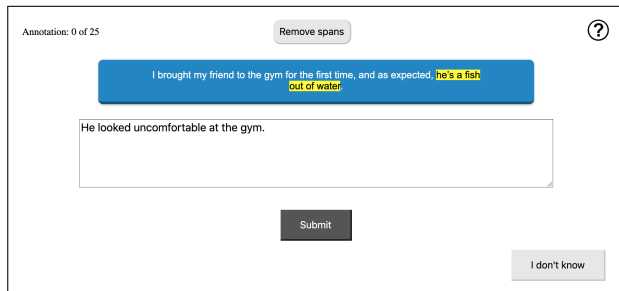


Figure 6: Free-text annotation example.

The storage of the data is guaranteed by MongoDB, its document-oriented construction, efficiency in querying large quantities of documents and scalability make this a suitable choice for AWOCATo. MongoDB is licensed as free software and has several Graphical User Interfaces (GUI) which facilitates its use by non-experts. Furthermore, MongoDB is suitable for storing unstructured data also supporting the simplicity and flexibility of the tool. MongoDB can be deployed on the same computing device which the tool is run on if the user wishes to use store the data locally. Alternatively, the tool can be connected to an IP address to facilitate the connection to a remote MongoDB instance. The data is sequentially provided to the annotator in the same order as the user stored it in the database, thus, maintaining the chronological order.

4. Annotation Experience

AWOCATo was originally designed for a sentiment annotation task also focusing on the contribution of certain text segments for the sentiment score. The annotators were tasked to 1) identify if the presented text was relevant for the given entity; 2) identify the sentiment expressed towards that entity; and 3) mark the spans contributing to the sentiment.

To begin an annotation session, the annotator must go to a specific URL where it is shown the log-in front page (figure

```

"port_number": 8080,
"accounts": [
  {"username": "Ruth", "password": "1234"},
  {"username": "Alan", "password": "abcd"}
],
"mongodb_settings": {
  "url": "localhost",
  "port": 27027,
  "database": "my_database",
  "collection": "health_texts",
  "items_per_query": 10
},
"consolidation_mode": false,
"guideline_at_start": true,
"annotation_mode": "bi-classification",
"mode_settings": {
  "bi-classification": {
    "data_specifications": {...},
    "interface_settings": {
      "idontknow_value": 4,
      "class_names": [
        {
          "class_name": "Infectious",
          "annotation_value": 1,
          "button_color_bg": "#FFB266",
          "button_color_border": "#FF8000"
        }, {
          "class_name": "Chronic",
          "annotation_value": 2,
          "button_color_bg": "#0E92A1",
          "button_color_border": "#09656F"
        }
      ]
    },
    "button_at_side": false
  }
},
"tri-classification": {...},
"sentiment": {...},
"sentiment-news": {...},
"textual-annotation": {...}
},
"consolidation_settings": {
  "minimum_annotations": 3,
  "standard_deviation": 0.31,
  "accepted_annotators": [0,1]
}

```

Figure 7: Example of the configuration JSON file structure in AWOCATo.

9).

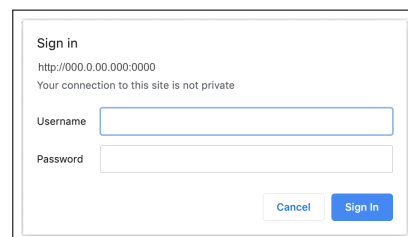


Figure 9: Log-in pop-up in AWOCATo.

After a successful log-in, the guidelines page is presented

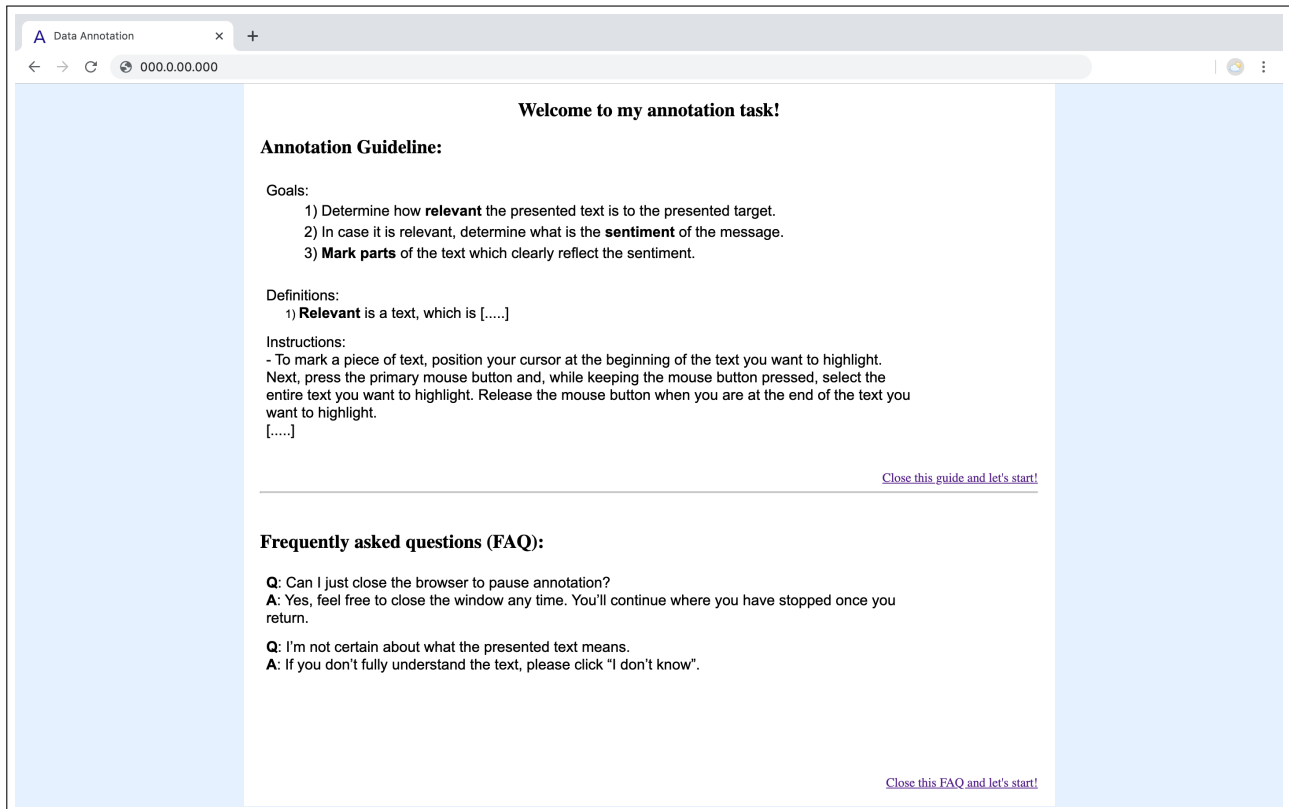


Figure 8: Portion of the guidelines page in AWOCATo.

(figure 8); depending on the number of sessions, the annotator is either introduced to the guidelines or can use these to refresh their memory regarding the task. Present throughout the guidelines page are several links stating “Close this guide and let’s start!” where the annotator can proceed at any point to the next step. In the annotation page (figure 2), the annotator finds a blue box with the text to annotate (box 5), and above this, the targeted entity (box 2). They must select how relevant the text is for the given entity using the first slider (box 6) followed by the selection of the sentiment (box 7). Note that the sentiment slider only appears after the relevance was determined; this design was implemented to avoid creating unnecessary distractions to the annotators, thus, keeping them fully focused in each step of the annotation. The sliders default is 0 *i.e.* not relevant and neutral, respectively. The option to provide a sentiment slider was based on past established research (Daudert et al., 2019; Vasiliu et al., 2016; Cortis et al., 2017; Gaillat et al., 2018).

Then, as per instructed in the guidelines, the annotator can mark the text spans contributing to the sentiment (in box 5; yellow highlighted text in figure 3). Upon reading the text, if the annotator cannot confidently annotate it they can click on “I don’t know” (box 9) to move to the next annotation. In this example, a continuous scale is used (boxes 6 and 7), however, the user can choose to apply the categorical annotation mode to ensure sentiment classes. Furthermore, the number of classes can be defined in the JSON configuration file. To track the progress, the top left corner shows the current annotation number and the total number of anno-

tations. At the top right corner, the annotator can click on the question mark (box 3) to check the guidelines page at any point during the annotation. As the progress is continuously saved with each annotation submitted, the annotator can simply close the browser to end the session.

5. Conclusions and Future Work

AWOCATo is an annotation tool supporting a variety of text formats; its construction was based on the principles of simplicity, flexibility, and customizability. These features allow AWOCATo to support a range of annotation tasks and user domains, filling the gap left by the absence of annotation tools that can be used by people with and without programming knowledge as well as those who wish to easily adapt a tool to less common annotation tasks.

Future work will focus on improving the flexibility of AWOCATo (*e.g.*, the optional use of a relational database) and customization. Additional features giving the user further customization choices in terms of the text appearance will be added to the JSON configuration file. We can also equip AWOCATo with further annotation modes such as named entity recognition and part-of-speech tagging. The aesthetics of the tool can also improve, making it more visually appealing albeit maintaining its simplistic and easy customization capability. Overall, the aim is to solidify AWOCATo as an easy-to-use, flexible, and customizable tool suitable for a broad range of annotation tasks and user backgrounds.

6. Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

7. Bibliographical References

- Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., and Gorrell, G. (2013). Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029.
- Chen, M., Hao, Y., Hwang, K., Wang, L., and Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879.
- Cortis, K., Freitas, A., Daudert, T., Huerlimann, M., Zarrouk, M., Handschuh, S., and Davis, B. (2017). Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval)*, pages 519–535. Association for Computational Linguistics.
- Daudert, T., Zarrouk, M., and Davis, B. (2019). CoSACT: A collaborative tool for fine-grained sentiment annotation and consolidation of text. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 34–39. Association for Computational Linguistics.
- Day, D., McHenry, C., Kozierek, R., and Riek, L. (2004). Callisto: A configurable annotation workbench. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal. European Language Resources Association.
- Finlayson, M. A. and Erjavec, T. (2017). Overview of annotation creation: Processes and tools. In *Handbook of Linguistic Annotation*, pages 167–191. Springer.
- Gaillat, T., Zarrouk, M., Freitas, A., and Davis, B. (2018). The ssix corpora: three gold standard corpora for sentiment analysis in english, spanish and german financial microblogs. In *The Language Resources and Evaluation Conference*. European Language Resources Association.
- He, J., Carbonell, J. G., and Liu, Y. (2007). Graph-based semi-supervised learning as a generative model. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 2492–2497. Association for Computing Machinery.
- Hey, T. and Trefethen, A. (2003). The data deluge: An e-science perspective. *Grid computing: Making the global infrastructure a reality*, pages 809–824.
- Jeong, M., Lin, C.-Y., and Lee, G. G. (2009). Semi-supervised speech act recognition in emails and forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1250–1259. Association for Computational Linguistics.
- Joutou, T. and Yanai, K. (2009). A food image recognition system with multiple kernel learning. In *Proceedings of the 16th IEEE international conference on Image processing (ICIP)*, pages 285–288. Institute of Electrical and Electronics Engineers.
- Khan, A., Baharudin, B., Lee, L. H., and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20.
- Kipp, M. (2001). Anvil-a generic annotation tool for multimodal dialogue. In *Proceedings of the Seventh European Conference on Speech Communication and Technology*. International Speech and Communication Association Archive.
- Lin, C.-Y., Tseng, B. L., and Smith, J. R. (2003). Videoanex: Ibm mpeg-7 annotation tool for multimedia indexing and concept learning. In *Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 203–212. Institute of Electrical and Electronics Engineers.
- Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., and Liang, X. (2018). doccano: Text annotation tool for human. Software available from <https://github.com/chakki-works/doccano>.
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20.
- 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. Association for Computational Linguistics.
- Vasiliu, L., Freitas, A., Caroli, F., McDermott, R., Zarrouk, M., Hürlimann, M., Davis, B., Daudert, T., Khaled, M. B., Byrne, D., et al. (2016). In or out? real-time monitoring of brexit sentiment on twitter. In *Proceedings of the 12th International Conference on Semantic Systems (SEMANTiCS)*. CEUR Workshop Proceedings.
- Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., and Sloetjes, H. (2006). Elan: a professional framework for multimodality research. In *5th International Conference on Language Resources and Evaluation (LREC)*, pages 1556–1559. European Language Resources Association.
- Yimam, S. M., Gurevych, I., de Castilho, R. E., 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. Association for Computational Linguistics.