TASSY—a free and open-source web-based system for creating surveys that include text annotation tasks.

II. RELATED WORK

NLP text annotation tools typically offer templates for common tasks like labeling named entities. The templates propose a workflow, GUI layout, and labels annotators should use. The focus is on minimizing the time for creating annotation tasks and maximizing annotator efficiency. To this end, the tools offer search and navigation functions that help users find relevant text parts.

The tools employ artificial intelligence to reduce annotation time by recommending labels. Text annotation tools commonly allow customizing the task-specific templates or create new templates from scratch to support other annotation tasks. However, none of the tools offers ready-to-use functionality to create survey questions.

The ability to create, distribute, and analyze large-scale surveys characterizes another class of tools. Survey tools are predominantly web platforms that allow users to create surveys by customizing question templates and configuring the survey logic, e.g., specifying follow-up questions that depend on the answer to a previous question. Despite the typically large number of features in survey tools, we could not find a tool that offered a text annotation question type.

TABLE I summarizes three of the most widely-used text annotation and survey tools to highlight their complementary strengths that triggered the development of our tool, TASSY.

TABLE I. WIDELY-USED TEXT ANNOTATION AND SURVEY TOOLS.

| Criterion                  | TASSY | Prodigy | Tag-tog | Doccano | SoGo | Survey | Monkey | Lime Survey |
|----------------------------|-------|---------|---------|---------|------|--------|--------|------------|
| Text annotations           | ✔     | ✔       | ✔       | ✔       | ✔    | ✔      | ✔      | ✔          |
| Survey questions           | ✔     | ✔       | ✔       | ✔       | ✔    | ✔      | ✔      | ✔          |
| Delivery                   | SWA   | SWA     | SWA     | SWA     | Cloud | Cloud  | Cloud  | Cloud / SWA |
| Costs                      | Free  | $390–$490 | $0–$99 | p.m.    | $0–$99 | $0–$499 | $0–$849 | p.a.       |
| License                    | MIT Commercial | MIT Commercial | MIT Commercial | MIT Commercial | GNU Commercial | GNU GNU | GNU GNU |

Legend: SWA—Self-hosted web application
III. SYSTEM OVERVIEW

TASSY is a progressive web application implemented using the Python Flask and Vue.js frameworks that support a MySQL/MariaDB or SQLite database. The tool’s appearance and functionality are highly customizable via a Python API.

The source code (MIT license) is available at https://bit.ly/35GNqOd. A demo system showcasing the survey on media bias that triggered the development of TASSY is available at http://tassy.blind-review.org.

Fig. 1 shows a question in TASSY that combines a text annotation task with a single-select input. The order of the questions can be configured to be identical for all participants or chosen randomly. The text that participants shall annotate is given in italic font ❶. The question or task description ❷ can contain instructions ❸. If desired, a link to detailed instructions, e.g., a coding book, can be inserted. Clicking the link will open the instructions in a modal window. Participants can select parts of the provided text at will. Ending the selection or pausing it for one second triggers the system to store the selected text as an annotation. Annotated text parts are highlighted in yellow and shown below the instructions ❹. Participants can reverse any annotation by clicking the x-button associated with each annotation. Administrators can configure lower and upper bounds on the length of the text parts that users may select. In the demo application, the maximum number of words that participants can select for one annotation is set to six. Exceeding the threshold will show an error overlay, informing the user that this selection is invalid. The lower part of the screen shows a single-select survey input ❺. Administrators can configure which inputs are mandatory. In the demo application, annotating parts of the text is optional, while completing two single-select questions is mandatory. Note that Fig. 1 only shows one of the two questions. Providing the required inputs activates the next button at the top of the page that allows proceeding to the next question ❻. A configurable section label and progress indicator inform the participants about the progression of the survey ❼.

The system includes several other input types that allow creating versatile surveys. Fig. 2 exemplifies a simple numeric input to record the age of participants. Fig. 3 shows a slider input that enables participants to indicate their political views on a discretized scale ranging from very liberal to very conservative. TASSY also enables open-ended questions, for which users can input text freely or as an extension to existing answers. Fig. 4 shows an example of the latter. The user can select one or multiple answers from the provided list and optionally enter additional answers for the shown question. For all input types, administrators can configure the inputs required for a) activating the next button that allows proceeding to the next question and b) the submit button that enables completing a section of the survey (for the example shown in Fig. 4, the section on participants’ news consumption).

IV. FUTURE WORK

In the future, we plan to enable the customization and administration of text annotation surveys via a graphical user interface rather than a Python API. Furthermore, we plan to offer a free hosted version of TASSY to eliminate the need to install the system locally. The planned extensions will increase the usability for less tech-savvy users.

![Fig. 1. Survey question requiring text annotation.](image1)

![Fig. 2. Example of a numeric input.](image2)

![Fig. 3. Example of a slider input.](image3)

![Fig. 4. Example of an extensible multi-select input.](image4)
REFERENCES

[1] T. Spinde, D. Krieger, M. Plank, and B. Gipp, “Towards A Reliable Ground-Truth For Biased Language Detection,” Sep. 2021.

[2] T. Spinde, L. Rudnickaia, F. Hamborg, and B. and Gipp, “Identification of Biased Terms in News Articles by Comparison of Outlet-specific Word Embeddings,” presented at the 16th International Conference (iConference 2021), Mar. 2021.

[3] T. Spinde et al., “Automated identification of bias inducing words in news articles using linguistic and context-oriented features,” Information Processing & Management, vol. 58, no. 3, p. 102505, 2021, doi: https://doi.org/10.1016/j.ipm.2021.102505.

[4] T. Spinde, L. Rudnickaia, Sinha Kanishka, F. Hamborg, B. and Gipp, and K. Donnay, “MBIC – A Media Bias Annotation Dataset Including Annotator Characteristics,” presented at the 16th International Conference (iConference 2021), Mar. 2021.

[5] M. D. White and E. E. Marsh, “Content analysis: A flexible methodology,” Library trends, vol. 55, no. 1, pp. 22–45, 2006.

How to cite this paper:
T.Spinde, K. Sinha, N. Meuschke, B. Gipp, “TASSY—A Text Annotation Survey System”, in Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2021.

BibTex:
@InProceedings{Spinde2021c,
title = {TASSY - A Text Annotation Survey System},
booktitle = {Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)},
author = {Spinde, Timo and Sinha, Kanishka and Meuschke, Norman and Gipp, Bela},
year = {2021},
month = {Sep.},
topic = {misc},
}