

# FAST: Fast Annotation tool for Smart devices

Shunyo Kawamoto, Yu Sawai\*, Kohei Wakimoto\*, Peinan Zhang\*

CyberAgent, Inc.

{kawamoto\_shunyo, sawai\_yu, wakimoto\_kohei, zhang\_peinan}@cyberagent.co.jp

## Abstract

Working with a wide range of annotators with the same attributes is crucial, as in real-world applications. Although such application cases often use crowd-sourcing mechanisms to gather a variety of annotators, most real-world users use mobile devices. In this paper, we propose “FAST,” an annotation tool for application tasks that focuses on the user experience of mobile devices, which has not yet been focused on thus far. We designed FAST as a web application for use on any device with a flexible interface that can be customized to fit various tasks. In our experiments, we conducted crowd-sourced annotation for a sentiment analysis task with several annotators and evaluated annotation metrics such as speed, quality, and ease of use from the tool’s logs and user surveys. Based on the results of our experiments, we conclude that our system can annotate faster than existing methods while maintaining the annotation quality.

## 1 Introduction

In the annotation of application tasks, it is important to work with a wide range of annotators as in real-world situations such as in the evaluation of the outputs of natural language generation (NLG) systems or sentiment analysis for user reviews. For instance, when evaluating the outputs of an NLG system for textual-ad creatives, the annotators, often called workers in a crowd-sourcing context, are usually required to annotate whether the generated text is fluent or not<sup>1</sup>.

Although such applications often use crowd-sourcing to gather a wide variety of annotators, statistics have shown that a large percentage of real-world users currently use mobile devices such as smartphones (Economic Research Office, 2020).

The application task itself, such as sentiment annotation and fluency assessment of the NLG system outputs, is simple. However, it is important to improve the operation interface and evaluate its contribution in detail. This is because a large amount of data is annotated by annotators with various operation proficiency levels.

In addition to the level of proficiency, mobile devices are more likely than desktop devices to be used on the go, during spare time, and in parallel with other tasks (Economic Research Office, 2020). Annotation using mobile devices is expected to collect more data as the time available for work is increased and the effort is reduced.

We propose a novel annotation tool called “FAST”<sup>2</sup>, and its contributions are summarized below.

- We propose and develop an annotation tool for the tasks that focuses on the user experience (UX) on mobile devices, which is important but has not yet been addressed in previous studies.
- We demonstrate that our tool is scalable, extensible, and customizable, and can be applied not only to the tasks described in this paper, but also to many other tasks.
- To evaluate the contribution of the tool to the improvement in the UX, we conducted an evaluation experiment with multiple annotators in a setting close to the practical use and obtained the metrics of annotation efficiency. We also conducted quantitative evaluations, such as inter-annotator agreement and subjective evaluations of UX.

\*These authors contributed equally.

<sup>1</sup>We use the term “annotation” to refer to the procedure of labeling a single instance.

<sup>2</sup>A short introduction video and the source codes are available at <https://github.com/CyberAgent/fast-annotation-tool>

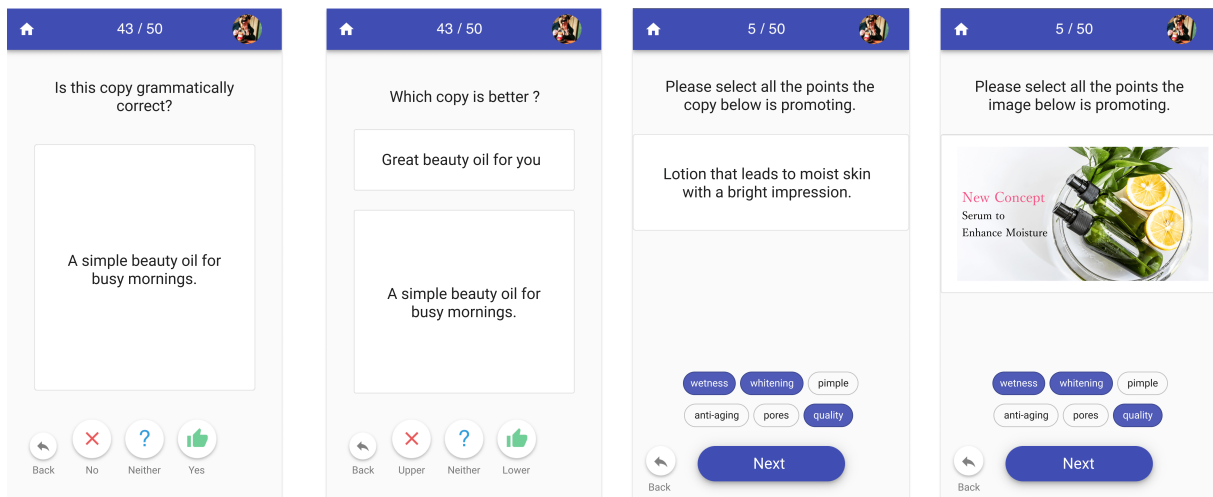


Figure 1: Use-case examples of FAST. In FAST, both texts and images are supported, and users can freely set questions and answers.

## 2 Related Work

In this section, we review the work related to the issues addressed by FAST. First, we note that FAST is not aimed at performing complete linguistic annotations; rather it focuses on applied tasks such as quality assessment of NLG systems. More specifically, FAST is designed for the annotation of individual tokens and sentences and not for the annotation of relations between tokens and sentences. In addition, we expect the annotators to be users with a wide range of attributes, such as those employed in crowd-sourcing.

Examples of tools that support extensive and detailed linguistic annotation are Brat (Stenetorp et al., 2012), MAE (Rim, 2016), INCEpTION (Klie et al., 2018), and Anafora (Chen and Styler, 2013). Recently, there have also been open-source software (OSS) tools such as doccano (Nakayama et al., 2018) that are suitable for applied tasks such as multi-label classification. Doccano is an OSS tool with mobile support and is currently one of the most popular annotation tools, with more than 4,000 stars on GitHub. This tool is focused on industrial targets such as sentiment analysis done by general users, rather than inter-entity structural annotation done by experts e.g. dependency structure annotations or coreference annotations. In this respect, Doccano and FAST have been developed based on the same motivation, and the target tasks are also similar. Although Doccano supports mobile devices, it essentially has the same UI structure as its desktop version and thus implementing extensible UI system on the top of it requires ef-

forts. Therefore, we developed FAST from scratch and introduced user interfaces dedicated to mobile devices and mechanisms for the easy custom annotation of interface elements which are required for simulating end-users' environment as close as possible.

### User Interface and Experience

There are several studies on improving the efficiency of annotation tasks. As a prominent example, SLATE (Kummerfeld, 2019) aims to improve the efficiency of workers who are skilled in keyboard input by focusing on the command line interface. Conversely, in a crowd-sourcing situation, there is a large variation in the operating skills of the workers; therefore, FAST adopts a graphical user interface (UI) and aims to improve efficiency by devising a new UI.

In application tasks, manual evaluation by a large number of non-domain experts is crucial. For example, in the evaluation of the generation quality of NLG systems, which is one of the tasks envisioned by the proposed tool, it is vital to have a group of evaluators similar to the user population that will see the generated sentences (van der Lee et al., 2019). To accommodate a wide variety of workers, the environment in which the tool operates must be versatile; in other words, it must have a web-based interface or a mobile interface. Although there are tools that support mobile devices, such as Doccano mentioned earlier, tools that have a user interface for mobile devices as their primary focus are scarce. In particular, the mobile UIs of the existing tools are the same as those of their PC versions;

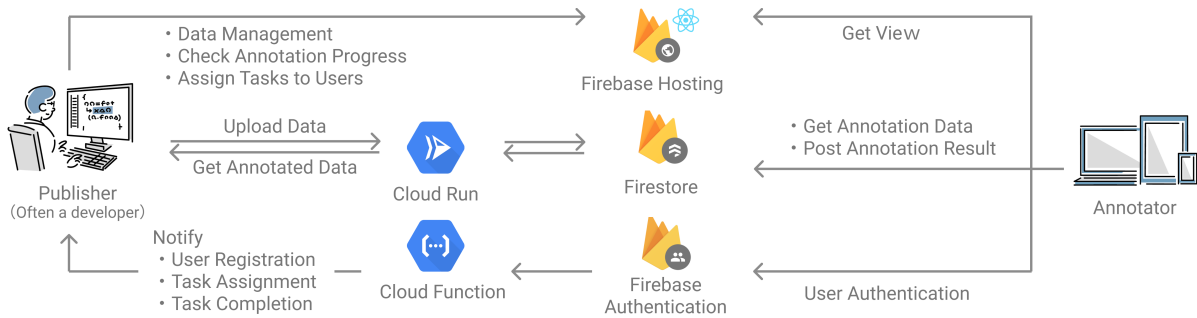


Figure 2: System architecture overview of FAST. The tool consists of serverless backend services and a user-facing interface for annotators, which can run on both PC and mobile devices.

thus, there is significant room for improving the efficiency of mobile interfaces. The proposed tool, FAST, aims to enable users who are not confident in operating PC terminals or those who are familiar with the mobile environment to work comfortably using a standard UI for mobile software.

### Evaluation for Annotation Tools

We conclude this chapter with related work on methods to evaluate the contribution of annotation tools. A comparison at the functional level is often performed for tools with a large feature set, such as RedCoat (Stewart et al., 2019). Conversely, there are situations in which the performance of actual tasks are directly evaluated.

TALen (Mayhew and Roth, 2018) is a tool specialized for creating Named Entity Recognition (NER) datasets in “low-resource” languages, which in some cases the annotators are not aware of. To assist annotators in this task, it includes “entity-propagation” where tagging an entity spills over to others similar to it, and there exists a mechanism to display the vocabulary of known languages. To evaluate these contributions, we adopted a method of comparing the NER task performance of the “low-resource” language with that of the baseline tool. In FAST, as in TALen, we employ metrics such as the performance in the assumed real task and the time spent for annotation work as quantitative evaluation indicators.

## 3 System Description

### 3.1 Supported Annotation Methods

FAST supports Card UI annotations and Multi-label UI annotations. Figure 1 shows the screenshots of each UI.

As can be observed from the figure 1, FAST is highly customizable. By flexibly designing questions and answers, issuers can create annotations

for a variety of tasks, ranging from simple binary classification to pairwise comparisons and element selection.

For example, if a set of generated sentences have to be ranked, we simply need to create pairwise sentences and then annotate them. Based on the results, using methods such as TrueSkill (Herbrich et al., 2007), the score and ranking of the sentences can be obtained. In addition, since HTML/CSS can be set as the evaluation target, it is possible to annotate the UI and the multimodal support close to the actual application, such as for evaluation combining images and text.

**Card UI** In FAST, we adopted the Card UI, which has been adopted by several mobile applications such as Tinder<sup>3</sup> and Grabble<sup>4</sup>, as a UI suitable for mobile environments. In the Card UI, the user is presented with a card containing text and a question. The annotator responds with two choices: whether the content of the card matches the question. The annotator can answer by swiping the card or tapping the button at the bottom of the screen. One feature of the Card UI is that the actions to perform are few. Although ordinary tools require at least two actions, selection and decision, the card system allows these actions to be executed with a single action. In addition, because swiping a card in either the left or right direction is a very familiar action in mobile devices, it is intuitive and requires little time for the user to get used to; it is therefore expected to provide fast and comfortable annotation while ensuring quality.

**Multi-label UI** The UI of the multi-label is depicted on the right side of Figure 1. In the Multi-label UI, multiple buttons are presented to a question. Annotators tap one or more buttons to answer this question. The multi-label method is an anno-

<sup>3</sup><https://tinder.com/>

<sup>4</sup><https://www.grabble.com/>

tation method that assumes multiple choices, and several evaluation tools implement it. Compared to the Card UI, the Multi-label UI can handle a wider range of annotations; however, the annotation efficiency is expected to decrease owing to the difference in the number of actions.

### 3.2 Architecture and Features

As shown in Figure 2, FAST adopts an architecture comprising Google Cloud Platform and Firebase. We adopted such a serverless design instead of hosting it on our own servers to reduce the management cost. Once FAST is deployed, there is no requirement to augment the DB or update the OS subsequently and can be therefore be conveniently operated. Additionally, since Firebase is a pay-per-use system, server costs can be kept very low for low-frequency access applications such as annotation tools. For example, 100,000 annotations cost only approximately \$0.4. which is significantly cheaper than purchasing a new machine.

**Detailed logs** It is important to keep accurate and detailed logs during an annotation. For example, when we want to estimate the difficulty of an annotation, it is useful to know how long it took the annotator to complete each question, how many times the annotator pressed the back button to revise the answer, which device was used to annotate, and so on. For this reason, FAST collects detailed logs, for example, the timestamp of the user's action (view, select, submit), the user agent, and the size of the screen.

In addition, because FAST can be linked to Google Analytics<sup>5</sup>, it is possible to know the location, device, and event information of annotators in real time.

**Device dependency** As FAST is a web application, it can be run as long as there is an accessible Internet environment and a browser. In other words, there is no device dependency, such as being limited to PCs or mobile devices, and it is possible to work with a wide range of annotators with attributes closer to those of real applications.

**Data communication via API** We expect that the annotation issuers will have a certain level of developer skills; for example, they could be researchers or machine learning (ML) model developers.

Data communication via APIs allows such developers to perform the entire process from issuing annotations to analyzing the results at a lower cost

compared to that in the case of a file format. Therefore, it can reduce the burden on the issuer in the use case, where the annotation is performed several times.

### 3.3 Overall Flow

The overall flow of the system can be described as follows:

**Step 1. Deploying the app and sharing the URL** As FAST is a web application, it has to be deployed by the annotation issuer. Then, the URL of the application has to be shared with the annotator.

**Step 2. Creating accounts** The annotation issuer and the annotator must sign up for a Google account on the web application in order to create an account.

**Step 3. Creating tasks and assignment** The annotation issuer is then required to create a task using the annotation API. The data to be sent here includes not only the ones to be evaluated but also meta-information, such as the title and format of the task. After the task is created, the issuer allocates the task to the registered annotators in the application.

**Step 4. Conducting annotations** The annotator confirms that the assigned task has been added to the home screen and executes the annotation.

**Step 5. Checking progress and data retrieval** Annotation issuers can check the progress of each annotator on the application and receive completion notifications via Slack. After the annotation is completed, the issuer retrieves the data via the API and performs the aggregation process.

## 4 Experiment

### 4.1 Metrics

We define some metrics that should be considered when measuring the effectiveness of annotation tools and explain what numbers should be tracked for each metric.

**Annotation Efficiency** One of the most important metrics in annotation tools is annotation efficiency. Additional data could be collected using a high-efficiency tool within a short period. To track efficiency, we measured the annotation time for the application using a fixed number of annotation questions. In practice, annotators may leave during the annotation process; therefore, in our experiments, logs that took more than 60 seconds were considered dropped annotations and were excluded.

<sup>5</sup><https://analytics.google.com>



**Annotation Quality** The annotated data should be of high quality. The difficulty of the task and the ease of use of the tool are considered to contribute to the quality. We evaluated the correctness rate of the annotation results and the inter-annotator agreement rate to examine the influence of the UI/UX of the tool on the performance.

**Qualitative Usability** In addition to the aforementioned two metrics, usability as perceived by the annotators is another key metric. To measure this, we requested each annotator to perform the following six annotation tasks and rank them in terms of usability. We also collected qualitative impressions of each task through a user survey in the form of free descriptions. We calculated the average rankings for each task and evaluated them based on the annotators' comments in the experimental results.

## 4.2 Experiment Setting

We conducted the following three comparative experiments of the evaluation metrics described in Section 4.1.

**PC vs. Mobile** As mobile devices are more portable and convenient to use than PC devices, people often use them in their spare time. Thus, we assume that mobile device applications are more customary and easier to use than that of PC devices. We conducted a comparison experiment with each device to verify this assumption.

**FAST vs. Existing Tool** We compare our proposed tool, FAST, with doccano, which is widely used in existing annotation tasks, as described in Section 2.

**Card UI vs. Multi-label UI** In the proposed tool, FAST, we compare the Multi-label UI, which is commonly used in evaluation annotation, with the Card UI adopted in this study.

Based on the aforementioned scheme, we conducted six annotation tasks through crowd-sourcing via Lancers, Inc.<sup>6</sup> for a total of about 40,000 annotations were worked by 18 annotators in five days. To avoid device mismatch during annotation, the app acquires the UserAgent and only accepts mobile annotations for mobile devices and PC annotations for PC devices. The annotation fee was set at \$0.045 per annotation, taking into account the pre-measured work speed.

In the experiment, we used the product review data crawled from the e-commerce site, which consisted of text with 50 or less Japanese characters

and a five-point rating score. As mentioned in 3.1, FAST can be used for a variety of tasks, but in the current study, we simplified the problem for the sake of evaluation and experiment using a task in which whether the review was satisfactory or unsatisfactory has to be selected.

In this study, we removed the data with an intermediate evaluation score of 3 and assigned scores 1 and 2 to "dissatisfied" and scores 4 and 5 to "satisfied." In the actual annotation, the annotator reads the content of the review and selects whether the content is satisfactory or unsatisfactory.

We used a task with binary labels for simplicity, but the Card UI can be applied to tasks with more labels than binary labels depending on the design. Hu (Hu et al., 2020) reduced the multiclass problem to binary labels using pseudo-labels based on a classification model.

## 4.3 Experiment Flow

The experiment consisted of five phases: two annotation phases and a user survey phase before and after each annotation phase. Each phase can proceed to the next phase only when it is completed.

**Phase 1. Tutorial and Pre-Survey** Before starting the annotations, we provided each annotator with a tutorial on the entire process and how each tool needs to be used. A user survey questionnaire was also given to the annotators, to collect information on their attributes, their level of skill with PC and mobile devices, and the amount of time they spend out of the office per day. Based on the collected information, we assigned the tasks to the annotators.

**Phase 2. General Annotation** We requested all the annotators to perform all the annotations described in Section 4.2 on a small set of 50 data points. This was done to allow each annotator to know and experience all the annotation methods to facilitate a fair comparison.

**Phase 3. Interim Survey** The annotator was asked to sort the six annotation methods in the order of their perceived ease of use through the General Annotation of Phase 2. They were also asked to describe the reasons why the methods were good or bad.

**Phase 4. Specific Annotation** In the General Annotation of Phase 2, the annotators were familiarized with all the annotation methods. Each annotator was asked to perform a large number of 2,000 annotations using one method in this phase. Ac-

<sup>6</sup><https://www.lancers.jp>

Table 1: Experiment results consist of Annotation Efficiency, Annotation Quality, and Qualitative Usability.

Tool	Device	UI	Annotation Efficiency		Annotation Quality		Qualitative Usability
			Total Time (m) (↓ better)	Average Time (s) (↓ better)	Accuracy (↑ better)	$\alpha$ (↑ better)	Average Rankings (↓ better)
<b>doccano</b>	PC		5.9	7.3	0.95	0.87	4.91
	Mobile		6.7	8.3	0.98	0.96	5.82
<b>FAST</b>	PC	Multi-label	4.8	5.4	0.97	0.93	2.91
		Card	<b>4.0</b>	<b>4.4</b>	0.97	0.94	2.12
	Mobile	Multi-label	4.9	5.2	0.97	0.95	3.15
		Card	4.6	4.7	0.97	0.96	<b>2.03</b>

ording to the interim user survey results in Phase 1, we assigned three annotators to each annotation method to be equally distributed in terms of attributes and skill level.

**Phase 5. Post-Survey** To check whether there was any change of opinion regarding a Specific Annotation, we asked the annotators to rank the ease of use of the six annotation methods again, as in the interim user survey of Phase 3. In addition, each annotator was asked to comment on what was good or bad about the Specific Annotation he or she was in charge of.

## 5 Results and Discussion

The experiment results are presented in Table 1. Annotation Efficiency is the total time required to complete all the tasks (Total Time) and average time per task (Average Time). Annotation Quality that refers to the accuracy and ratings on the inter-annotator agreement is derived from Krippendorff’s  $\alpha$ . Qualitative Usability is calculated from the average rankings of the Post-Survey ratings.

Notably, we excluded two annotators while calculating the result because they had a markedly low accuracy compared to others as shown in Figure 3. In addition, the annotation speeds varied significantly among the annotators in the experiment. For the Specific Annotation, the annotators were divided into groups. To reduce user bias between the groups, we aggregated the annotation efficiencies from the General Annotation results.

**PC vs. Mobile** Table 1 shows that the performance of PC and mobile devices is almost the same in terms of the annotation speed and quality. This suggests that the same level of annotation can be performed on a mobile device as on a PC. On the other hand, we could not confirm the superiority of mobile devices in terms of efficiency because the experiment was conducted for only five days. For a more appropriate verification, a comparison

based on the measurement of user’s working speed and fatigue over a longer period and with regular annotations is necessary. In terms of usability, the mobile device with the Card UI received the most first-place votes in both the interim and post-user surveys, and its average ranking was 0.23 higher than that of the PC with the Card UI. The superiority of the mobile devices in terms of usability was therefore confirmed. Additionally, three out of nine annotators assigned to mobile devices in the Specific Annotation performed their annotations during their spare time, such as in trains, cars, and cafes, confirming the superiority of mobile devices that allow work to be performed in any location.

**FAST vs. Existing tool** Table 1 shows that the average annotation time for the FAST Card UI is 4.7 seconds on mobile devices, compared to 8.3 seconds for existing tools; therefore, approximately 43% of the annotation time can be reduced using FAST. One of the reasons for this is the number of actions required to select and decide. Doccano requires at least three actions for each annotation, that is, display of the options, selection, and decision, but the Card UI of FAST requires only one action for selection and decision. It is thought that this difference affects the speed and usability of the system. Another reason is whether the UI is designed for mobile devices or not. In the user survey, we confirmed the following opinions about the mobile devices of the existing tools: “I could not operate it with one hand” and “I felt stressed because I had to scroll because the screen size did not fit the device.” As for the annotation quality, the agreement rate of the existing tool on PC was 0.87, which was lower than the other patterns. In the annotator’s opinion, “errors occurred” and “sometimes the tool does not respond to button presses” were confirmed, which is considered to be due to a problem in the application.

**Card UI vs. Multi-label UI** Table 1 shows that

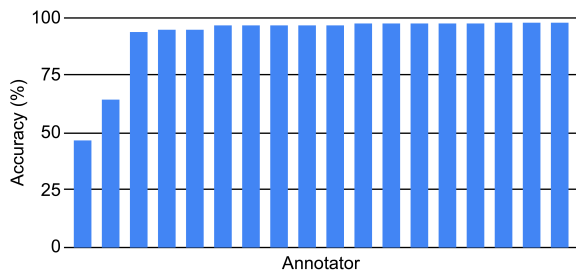


Figure 3: Sorted annotators' accuracy in the Phase 4 Specific Annotation.

the Card UI is 18% faster on mobile devices and 25% faster on PCs than the Multi-label UI. The Card UI is also superior in terms of usability.

For the same reason as in the comparison with the existing tools, the difference in the number of actions is considered to have affected the speed and ease of use.

## 6 Conclusion

In this study, we proposed and developed FAST, an annotation tool for application tasks that emphasize the impact of UI/UX on mobile devices, which is a crucial topic, although not investigated thus far.

FAST is a web application designed for use on any device, including mobiles and PCs. This web application is highly customizable in that the issuers can create views that are optimized for their tasks utilizing two types of UI: Card UI and Multi-label UI. We compared the devices and UIs, as well as an existing tool with FAST, in an experiment involving a sentiment analysis task; we also evaluated their efficiency, quality, and usability.

The results showed that the mobile operation of FAST provides annotators with a more user-friendly experience while maintaining the efficiency and quality of the PC. Furthermore, in comparison with the existing tools, FAST was able to reduce the annotation time by 43% and an improvement in work efficiency was also confirmed.

In the future, we plan to conduct quantitative evaluations using additional detailed indicators such as the trajectory of user operations, and task load metrics measured using NASA TLX. Moreover, we aim to increase the number of supported task types, enhance the management functions, and support on-premise environments to strengthen its usefulness as a general-purpose annotation tool.

## References

- Wei-Te Chen and Will Styler. 2013. [Anafora: A web-based general purpose annotation tool](#). In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 14–19, Atlanta, Georgia. Association for Computational Linguistics.
- Economic Research Office. 2020. [Information and Communications in Japan 2020](#). *White Paper 2020 by The Ministry of Internal Affairs and Communications of Japan*.
- Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. [TrueSkill™ : A Bayesian Skill Rating System](#). In *Advances in Neural Information Processing Systems 19*, pages 569–576. MIT Press.
- Hengtong Hu, Lingxi Xie, Zewei Du, Richang Hong, and Qi Tian. 2020. [One-bit supervision for image classification](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Jan-Christoph Klie, Michael Bugert, Beto Boulosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. [The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.
- Jonathan K. Kummerfeld. 2019. [SLATE: A super-lightweight annotation tool for experts](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Florence, Italy. Association for Computational Linguistics.
- Stephen Mayhew and Dan Roth. 2018. [TALEN: Tool for annotation of low-resource ENTities](#). In *Proceedings of ACL 2018, System Demonstrations*, pages 80–86, Melbourne, Australia. Association for Computational Linguistics.
- Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. [doccano: Text Annotation Tool for Human](#).
- Kyeongmin Rim. 2016. [MAE2: Portable Annotation Tool for General Natural Language Use](#). In *Proc 12th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 75–80.
- 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, Avignon, France. Association for Computational Linguistics.

- Michael Stewart, Wei Liu, and Rachel Cardell-Oliver. 2019. [Redcoat: A collaborative annotation tool for hierarchical entity typing](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 193–198, Hong Kong, China. Association for Computational Linguistics.
- Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. [Best practices for the human evaluation of automatically generated text](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.



## A System Details

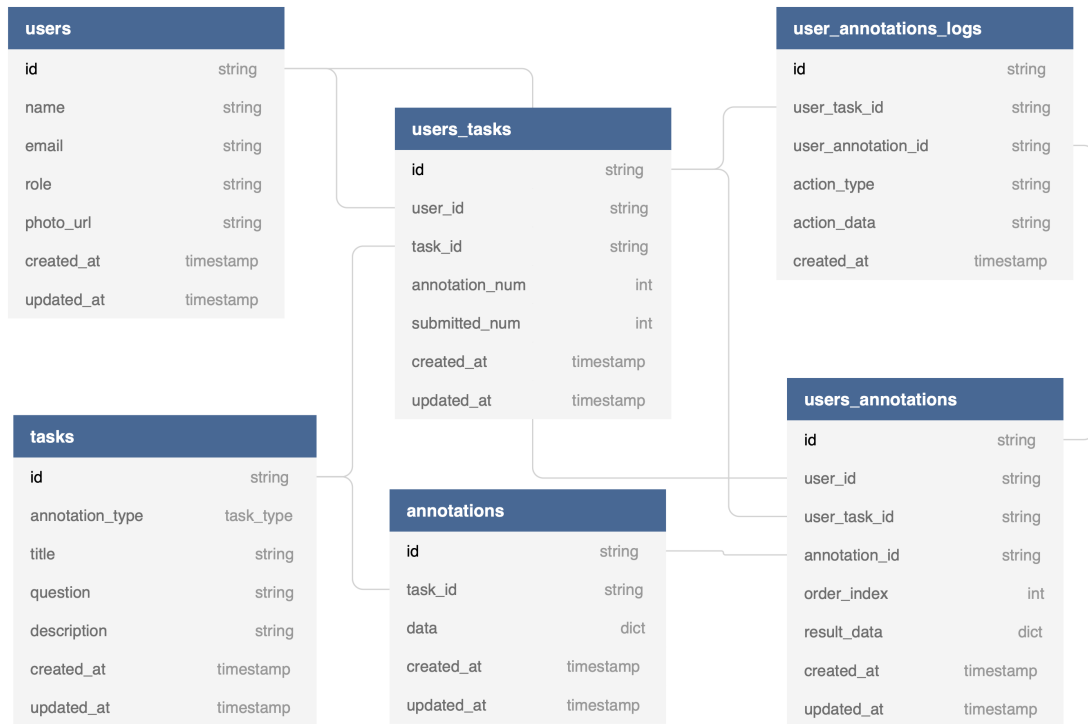


Figure 4: DB Diagram

Listing 1: Sample of annotated data to be fetched via API

```
{
  "task": {
    "id": "card-demo-20200601",
    "annotation_type": "card",
    "title": "Card Demo",
    "question": "This is card demo",
    "description": "This is a card demo, so feel free to annotate it as you wish.",
    "created_at": "2021-06-01T07:39:38.916473+00:00",
    "updated_at": "2021-06-01T07:39:38.916490+00:00"
  },
  "annotations": [
    {
      "id": "0V7KGVUs3cijjr1HG6J6",
      "name": "Shunyo Kawamoto",
      "email": "kawamoto_shunyo@cyberagent.co.jp",
      "data": {
        "text": "This is a annotation.",
        "show_ambiguous_button": true,
        "hidden_data": {
          "desc": "Data for aggregation. It can be a dictionary or a string."
        },
        "yes_button_label": null,
        "question_overwrite": null,
        "no_button_label": null,
        "baseline_text": null
      },
      "result_data": {
        "result": "No",
      },
      "order_index": 44,
      "user_id": "enTzOydWPXfYyXF9vOhuu38a5DA2",
      "user_task_id": "TG9EvlyOUQ43hBYlAM8b",
      "annotation_id": "tVl0rySSTMV2m0JEIris",
      "created_at": "2021-06-01T08:00:54.099000+00:00",
      "updated_at": "2021-06-01T08:00:54.099000+00:00"
    },
    ...
  ]
}
```

## B Survey Details

Table 2: Questions in the Pre-Survey

Question	Format	Options
Email address	Free text	
Gender	Selection	Male, Female, Other
Age	Selection	10s, 20s, ..., Over 70s
Current address (prefecture)	Free text	
Occupation	Free text	
Number of days per week that you go out for more than one hour	Selection	1day, 2day, ..., 7day
Hours away from home per day	Selection	Almost none, 1hour, 2hour, ..., Over 10hour
Model name of mobile device	Free text	
Time spent on the PC per day	Selection	Almost none, 1hour, 2hour, ..., Over 10hour
Time spent on mobile devices per day	Selection	Almost none, 1hour, 2hour, ..., Over 10hour
PC, Mobile proficiency	5-point Likert scale	

Table 3: Questions in the Interim Survey

Question	Format	Options
Email address	Free text	
Ranking of ease of use for each pattern	Selection	1st, 2nd, ..., 6th
Reasons for the patterns that were easiest to use	Free text	
Reasons for the most difficult patterns to use	Free text	
Other problems during annotation, etc.	Free text	

Table 4: Questions in the Post-Survey

Question	Format	Options
Email address	Free text	
Ranking of ease of use for each pattern	Selection	1st, 2nd, ..., 6th
Good points about the tool you were responsible for in Specific Annotation	Free text	
Bad points about the tool you were responsible for in Specific Annotation	Free text	
(Mobile annotators only) The location where you annotated	Free text	