Ga11y: An Automated GIF Annotation System for Visually Impaired Users

Mingrui "Ray" Zhang
The Information School, University of Washington
Seattle, USA
mingrui@uw.edu

Mingyuan Zhong
Paul G. Allen School of Computer Science, University of Washington
Seattle, USA
mingyuan@cs.washington.edu

Jacob O. Wobbrock
The Information School, University of Washington
Seattle, USA
wobbrock@uw.edu

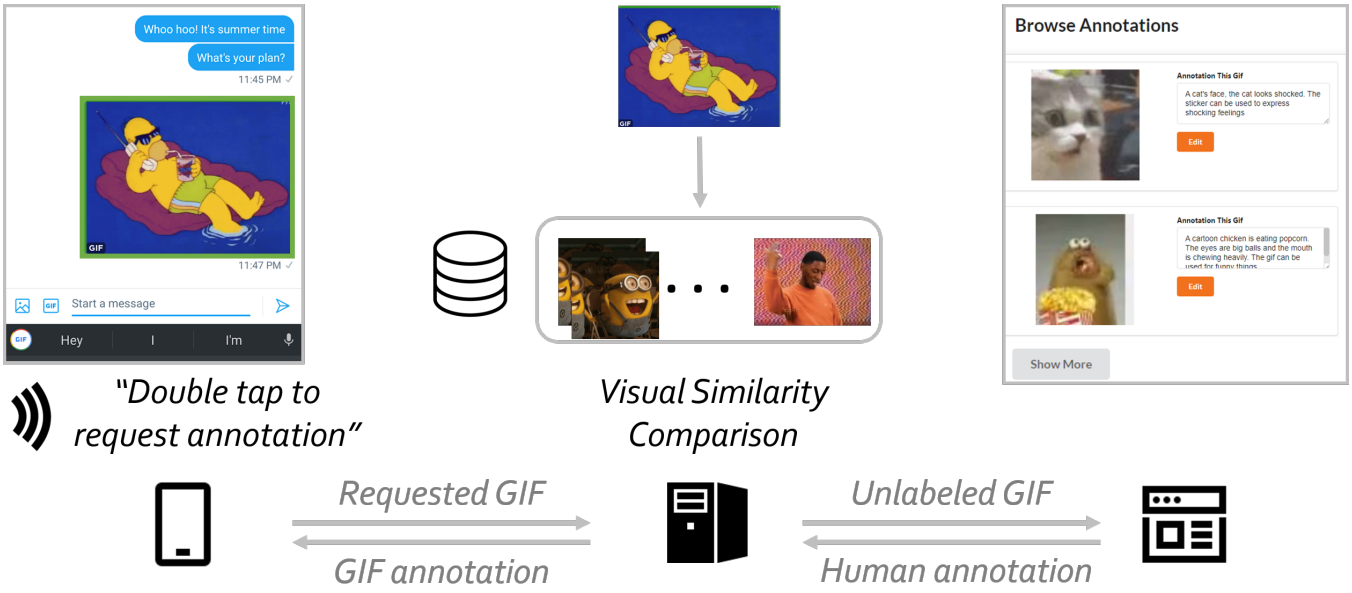

Figure 1: The components of Ga11y. The user requests the GIF annotation on the mobile client via the screen reader, and the requested GIF is searched for in the human annotation database on the server. If there is a visually similar GIF with a human annotation, that annotation will be returned; otherwise, a machine-generated annotation is returned, and the unlabeled GIF is then displayed in the Web-based human annotation interface. Once the GIF is annotated by volunteers on the website, the annotation is updated in the server's database for future retrieval.

ABSTRACT

Animated GIF images have become prevalent in internet culture, often used to express richer and more nuanced meanings than static images. But animated GIFs often lack adequate alternative text descriptions, and it is challenging to generate such descriptions automatically, resulting in inaccessible GIFs for blind or low-vision (BLV) users. To improve the accessibility of animated GIFs for BLV users, we provide a system called *Ga11y* (pronounced "galley"), for creating GIF annotations. Ga11y combines the power of machine intelligence and crowdsourcing and has three components: an Android client for submitting annotation requests, a backend server and database, and a web interface where volunteers can respond to annotation requests. We evaluated three human annotation interfaces and employ the one that yielded the best annotation quality. We also conducted a multi-stage evaluation with 12 BLV participants from the United States and China, receiving positive feedback.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3502092

CCS CONCEPTS

• **Human-centered computing** → **Accessibility**; **Interactive systems and tools**.

KEYWORDS

GIF, images, blind, low vision, text description, human annotation, crowdsourcing, accessibility.

ACM Reference Format:
Mingrui "Ray" Zhang, Mingyuan Zhong, and Jacob O. Wobbrock. 2022. Ga11y: An Automated GIF Annotation System for Visually Impaired Users. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3491102.3502092

1 INTRODUCTION

Animated *Graphics Interchange Format* images, or "GIFs," are looped animations comprising a sequence of images, and are popular forms of content on the Web, messaging, and social media. Many GIFs are made from clips of videos (e.g., movies, TV shows, etc.) or animated cartoons (such as "stickers"), and because of their dynamic nature, GIFs can be used to express richer and more nuanced meanings than static images or text. People use GIFs on social media platforms (such as Twitter), in online messaging apps (such as Facebook Messenger), and to make "memes." As of July 1, 2021, GIFs were used on 21.3% of all websites [40].

The prevalence of GIFs on the internet means that blind or low vision (BLV) users encounter them often. As we report below, in our study with 12 BLV users, all had encountered GIFs on social media platforms or received them in online massaging apps. However, most GIFs have no text annotations that would make them accessible to screen readers. For example, only 0.04% of GIFs on Twitter were annotated in February 2020 [13], and those that *were* annotated typically had short unhelpful descriptions of just one or two words.[1] Unlike static images, GIFs often contain a sequence of images with holistic meaning, which is challenging for current computer vision technologies to recognize and describe [25]. The fact that GIFs are inaccessible to screen readers creates a barrier for BLV users to fully participate in internet culture, causing social exclusion and a reduced richness of online experience.

Further exacerbating the challenge of making animated GIFs accessible to BLV users is that even when a GIF's visual content can be accurately recognized using machine intelligence, having only a GIF's visual information is usually insufficient for understanding it. Unlike emojis, whose descriptions are created and standardized by the Unicode Consortium[2], GIFs are mostly created by individual users, and their meanings can largely depend on context, such as the background of a movie character, the sarcasm of a meme, or the emotion contained in a facial expression. Therefore, it is necessary to have a person who understands this context to supply the meaning of an animated GIF.

To address these challenges, we present *Ga11y* (pronounced "galley"), a GIF annotation system that combines machine intelligence and crowdsourcing to supply annotations for animated GIFs on the internet (see Figure 1). Ga11y contains three components: (1) an Android client in which users can trigger annotation requests, (2) a server for GIF matching and annotation storage, and (3) a web interface for human annotation. The Android client runs an accessibility service on the phone, which detects animated GIF and "sticker" elements on the screen. When the focus of the screen reader is on such an element, the user can press a "request for annotation" button to record the GIF and send it to the server. The server compares the similarity of the requested GIF with existing GIFs in the database. If the GIF is not in the database or has not yet been manually annotated, the server generates an automated description using computer vision; otherwise, a human-annotated description is returned. In the meantime, Ga11y's web interface enables human annotation for all requested GIFs, and once an annotation is manually updated by a volunteer, it is supplied to the server database for future retrieval. In this way, users get timely annotations even when GIFs are new to the server, and over time, the number of human-annotated GIFs increases. To enable others to contribute to and extend Ga11y, we open-source its entire implementation as part of this work.[3]

To increase our chances of receiving useful human annotations from Ga11y's web interface, we explored three annotation interface styles for volunteers annotating GIFs. These interface styles were: (1)*freeform*, where the volunteer was only asked to "provide a description" without any guidance; (2) *semi-structured*, where the volunteer was asked to "provide a description of the GIF," with specific guidance on important aspects of a GIF to mention; and (3) *structured*, where the volunteer was asked to answer a set of structured questions regarding a GIF's content. We collected and evaluated these GIF annotations from the three styles using the Amazon Mechanical Turk platform, and gathered feedback from BLV users ($N = 11$). Our results showed that both sighted and BLV users preferred the *semi-structured* style for providing GIF annotations. We therefore implemented this style as the web annotation interface for Ga11y.

We then conducted a multi-stage user study with 12 BLV participants to evaluate Ga11y. Specifically, we were interested in how the annotation system affected participants' online communication experiences on social media and messaging platforms. We first conducted a one-hour remote usability test, where participants used Ga11y to request annotations of five GIFs that were already manually annotated on the server, and five new GIFs not in the server. Then, after the usability test, participants were encouraged to use Ga11y for two days whenever they encountered GIFs or "stickers" on their phones and provide feedback. Our results showed that users perceived Ga11y as a helpful tool for their online communication. They also rated Ga11y as having "high usability," with an average System Usability Scale (SUS) [8] score of 89.1 out of 100. Each participant also used the system 13.5 times per day on average "in the wild."

We make three primary contributions in this work:

(1) We explored three interface styles for providing human annotation for GIFs, evaluated them with both sighted and BLV users, and identified the best style for generating high-quality GIF annotations;
(2) We developed *Ga11y*, an end-to-end system providing annotations for GIFs and "stickers" on mobile devices. The contributions within the system include the interaction design for requesting a GIF description, animated GIF re-construction and similarity matching algorithms based on computer vision, and the successful combination of machine and human intelligence. Additionally, we provide the source code of Ga11y's implementation;

[1] https://blog.twitter.com/en_us/a/2016/introducing-gif-search-on-twitter
[2] https://unicode.org/consortium/consort.html
[3] https://github.com/DrustZ/Ga11y

(3) Through a multi-stage user study, we evaluated the usability and performance of Ga11y. Our results show that Ga11y received positive feedback from users, is highly usable, and our human annotations were perceived as the most helpful feature within the system.

2 RELATED WORK

In this section, we review work related to Ga11y, including research on "GIF culture" and attempts to improve GIF accessibility. We also review technical work, including crowdsourcing solutions for accessibility and interaction techniques for mobile app accessibility.

2.1 The Use of GIFs on the Internet

Images in *Graphics Interchange Format*, called "GIFs," are in a file format that can contain multiple images played in an animation loop. Such files are usually extracted from clips in videos [12] and from animations and cartoons created by artists as "stickers" [42]. Compared to emojis, which are controlled by the Unicode Consortium (see footnote 2), GIFs are more "democratic," where in theory, everyone can create or modify GIF content, enabling personalized communications [44, 48]. GIFs are commonly used on social media platforms such as Facebook, Tumblr, Twitter, and Reddit [13], and in online messaging apps, such as Facebook Messenger, WhatsApp, and Wechat [48]. By 2018, the GIF database and search engine service *Tenor* reportedly had 12 billion searches every month [38], while another service, *Giphy*, reached 700 million monthly active users in 2019 [36]. Clearly, the popularity of GIFs online is immense.

Researchers have found that GIFs make online interaction more engaging than static images or text [19] because of their animation, storytelling capabilities, and utility in expressing emotions [5]. Jiang et al. [20] summarized users' motivations for sending GIFs online, including to convey emotion, to express nuanced meanings that were hard to convey with text, to make humorous and eye-catching posts, and to start engaging conversations. GIFs are "a visual language unto themselves, and an emotive vocabulary made out of culture" [28]. Indeed, many GIFs are blended with pop culture and memes, where contextual information is vital to understand their meanings [20, 21, 43]. For example, the source of a GIF, the meaning of the text meme on the GIF, and the usage of a GIF are all deeply embedded in one's cultural background [17, 28]. In this work, we take the first step to make "GIF culture" accessible to blind and low vision (BLV) users.

2.2 GIF Accessibility for Blind and Low Vision Users

Visual content such as images, animated GIFs, stickers, badges, memes, and emojis can enhance online communication and social interaction. Unfortunately, much of this visual content remains inaccessible to BLV users. Although there are many efforts to improve the accessibility of *static* images, including emojis [14, 27, 32, 45], most *animated* content such as GIFs and "stickers" are inaccessible to BLV users. According to Gleason et al. [13], only 0.04% of GIF content on Twitter contained alternative text in February 2020. According to our multiple studies with BLV participants ($N = 19$), all had encountered GIFs online, but most of the time the contents were unlabelled, causing participants to either ignore GIFs or ask for help from sighted people. Although computer vision techniques are able to generate reasonable descriptions for many static images, correctly describing the contents of animated GIFs is still a challenging research problem [25], let alone providing the contextual and cultural information needed to understand them.

The common solution for making images accessible is to use alternative text, which is a method of adding text descriptions to images so that a screen reader can read it for BLV users. Researchers have investigated the usability of alternative text extensively, including how framing affects users' trust [27], what granularity descriptions should have in different usage scenarios [37], auto-generated captions using the surrounding text of an image [15] or using user-generated comments [41], alternate designs such as multi-modal and interactive alt-text for rich visual content [30]. For example, Gleason et al. [13] proposed using audio descriptions to supply emotive qualities to the descriptions of animated GIFs.

Beyond prior academic research, there has been little recognition in industry that the inaccessibility of GIFs is a problem. Twitter has launched the alt-text function for GIFs [39], and many GIF services such as *Giphy* and *Gboard* offer a one-word description of GIFs, such as "laugh" or "dance" when exploring GIFs; however, providing such short descriptions does not aid users in understanding nuanced expressions contained in animated GIFs. Neither does it help them to decide which GIFs to use. Furthermore, once a GIF is selected and sent from a keyboard like *Gboard*, it become unlabelled at its destination (e.g., for its viewer or recipient).

In this work, we used text descriptions for GIF annotations as all screen readers are compatible with text, and it is easy for volunteers to generate text annotations. However, as GIFs can be polysemic and rely on contextual and cultural knowledge, it is not adequate to only have the description for visual content; otherwise, the description can cause misunderstandings [20]. We therefore explored three annotation interface designs and evaluated them with both sighted and BLV users to discover the most effective one.

2.3 Crowdsourcing for Accessibility

Crowdsourcing is an effective solution for solving problems that are challenging for computers, since it distributes quick tasks to a potentially large pool of workers. Previous research has applied crowdsourcing to help people do document editing [6], answer questions on social networks [31], and conduct end-user elicitation studies [3, 4]. It is also widely applied in the field of accessibility. For example, *VizWiz* [7] utilized crowd-workers to answer questions with photos posted by BLV users. Taking this idea a step further, the mobile app *Be My Eyes* [4] allows crowd-workers to answer calls from BLV users in real-time.

Because of the complex contextual and cultural information required to understand animated GIFs, we applied crowdsourcing to generate alt-text for GIFs. However, the quality of the annotations generated by crowd-workers can vary a lot. Researchers have investigated various ways to improve annotation quality: (1) by improving the task designs, such as by asking a series of questions related to annotations instead of having the worker freely compose annotations [29]; (2) by letting crowd-workers self-assess their own answers [9] or assess each other's answers [6]; or, (3) by showing

[4] https://www.bemyeyes.com/

examples of high-quality annotations [22, 35]. Based on previous investigations, we generated three task designs for human GIF annotations and evaluated them to determine Ga11y's web annotation interface.

2.4 Interface Augmentation for Improving Mobile Accessibility

To improve the accessibility of mobile apps, existing user interfaces often need to be augmented, usually in the form of an accessibility service [2, 24, 33, 46]. One general approach of such augmentation is through the creation of an "interaction proxy," as proposed by Zhang et al. [46]. An interaction proxy runs as an accessibility service and creates a mid-layer handler inserted between an app's original interface and the manifest interface (i.e., the interface exposed to the end user, such as Android's TalkBack), in order to fix accessibility issues in the original interface. Interaction proxies have been demonstrated to repair accessibility issues in 26 apps in previous studies [47]. Interaction proxies have also been adopted to enable custom interactivity that is otherwise not supported by the operating system. For example, APPINITE [24] utilizes an interaction proxy to intercept touch events and provide task-related visualizations.

Although interaction proxies focus on augmenting individual apps, system-wide user interface augmentation with personalizable static overlays has been proposed by Rodrigues et al. [33], where a customizable overlay layer is consistently available across apps. A similar technique has also been proposed to support people with upper extremity motor impairments: RePlay [2] allows users to create mappings from triggering actions to interaction events for games that do not use standard Android interface elements.

In Ga11y, we developed an interaction proxy to listen to navigational accessibility events, insert action buttons for requesting GIF annotations, and play the annotations returned by our server.

3 USER INTERACTION IN GA11Y

In this section, we describe user interaction scenarios when using Ga11y to request GIF annotations from a smartphone device, and provide human annotations for GIFs on Ga11y's web interface. We then present how each part of the system is designed and implemented.

3.1 Requesting a GIF annotation

Ga11y runs as a background service on an Android device. Blind or low vision (BLV) people can use TalkBack[5] to navigate through screen elements on an Android phone, and they can move the focus of TalkBack by swiping left or right. (They also can read screen elements by keeping a finger persistently on the screen.) When a BLV user moves the TalkBack focus onto a GIF element (Fig. 2a), the Ga11y service identifies that the focused element is a GIF based on properties supplied by system *AccessibilityEvents* and app-specific heuristics (details explained in section 5.1). If the element is recognized as a GIF, then the user will hear "double tap

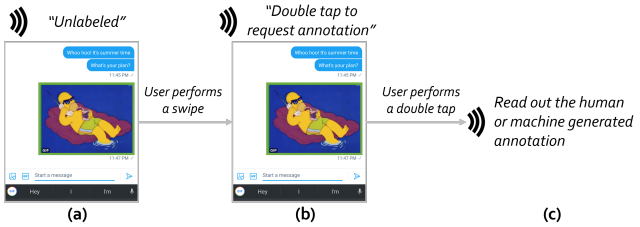

Figure 2: Requesting a GIF annotation on a smartphone: (a) an unlabeled GIF is focused by the screen reader; (b) on the next swipe that moves the screen reader focus, the user can double-tap to request the annotation; (c) after the request is processed by the server, the annotation will be returned and read out loud. A human annotation is returned if the requested GIF is already annotated in the database; otherwise, a machine-generated annotation is returned.

to request annotation" on their next swipe (Fig. 2b) [6]. The user can either continue swiping to ignore the action, or double-tap to request the annotation of the GIF, which triggers the Ga11y service to record the GIF for five seconds [7] and send a request to the server. After the GIF is recorded, the client will make beep sounds until the annotation is successfully fetched to indicate the requesting status, and the annotation will then be read aloud; the user is free to move the focus during the fetching process. If the client fails to fetch the annotation due to connection issues, a double beep will be made to indicate the failure.

On the server side, the requested GIF will be compared to existing GIFs in the annotation database based on their visual similarity. If there is a similar GIF in the database, its human annotation will be sent back to the user's smartphone and read out by the screen reader; if not, the request will be added as an unlabelled GIF in the database and, in the meantime, a computer-generated annotation derived using computer vision will be sent to the user (Fig. 2c). In this way, the user always receives an annotation of the GIF quickly, although the machine-generated annotation will generally be less informative than the human-generated ones. Over time, human annotations will accrue as will the number of annotated GIFs, giving users the best annotations possible.

3.2 Annotating a GIF

The web interface of Ga11y is available to the public, allowing volunteers to annotate GIFs through the website. The website contains two pages as shown in Figure 3. One page displays all of the GIFs whose annotations are requested from users' smartphones but which are not yet annotated; website users can click a GIF to add an annotation. The other page displays GIFs that are already annotated by volunteers, and website users can click "edit" to revise

[5]TalkBack is a screen reader service included in the Android operating system.

[6]We designed an extra swipe for annotation request as a double tap on the original GIF content usually has its own functionality (such as zoom in or open a GIF page view) which might conflict with the annotation function.

[7]An average GIF is about 3 seconds [25], we added 2 more seconds for redundancy. If the user moves the focus during the recording, the request is cancelled by default.

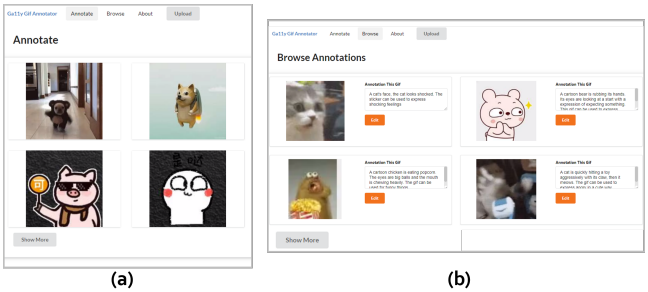

Figure 3: The annotation web interface: (a) the unlabelled GIF browsing page, and (b) the annotation browsing page.

and potentially improve [8] the annotation. Once an annotation is updated, the data is synced in the database.

4 DESIGNING THE GIF ANNOTATION TASK

As anyone can contribute annotations through the Ga11y web interface, we need to ensure that our task design leads to high quality annotations. For one thing, crowdsourcing tasks always face challenges of quality control, as the crowd is composed of people with diverse backgrounds and abilities [9]; for another, BLV users might have different needs than sighted users when trying to understand a GIF's content (e.g., sighted users might place more emphasis on visual attributes of the content). In order to design an annotation task that is easy for annotators to understand and that yields useful information for BLV users, we conducted an evaluation with three different annotation interface styles. We then ran a study on Amazon Mechanical Turk to obtain annotations from the three interface styles, and evaluated the results based on both sighted and BLV users' feedback.

4.1 Annotation Task Styles

Based on previous research on improving image annotation quality [9, 10, 22, 26, 29] and understanding BLV users' needs for visual content descriptions [13, 19, 30, 35, 37], we designed three GIF annotation interface styles: *freeform*, *semi-structured* and *structured*.

Freeform. In the *freeform* style, there is a GIF to be annotated and an instruction to "*Annotate the GIF in English. How would you describe this GIF and its context to someone, so that they can understand its content even without seeing the GIF?*" The only requirement is "*use a minimum of 8 words.*" The volunteer is then free to write the annotation.

Semi-structured. In the *semi-structured* style, beyond only providing the same instruction as the *freeform* style, we also include several tips for guidance. The tips are generated from previous research on describing GIFs and images generally [13, 25, 27, 37]:

- Please describe the visual content and the related information that is helpful for understanding the GIF;
- Please describe any actors (people, animals, etc.), their actions and expressions, the activities underway, and the environment in which those activities are taking place;
- If this GIF contains clips or actors from movies, television, or similar sources, please describe that information;
- If there is text in the GIF, please describe it;
- Please keep your description concise overall;
- Please use a minimum of 8 words.

Structured. In the *structured* style, we divided the tips from the *semi-structured* style into multiple questions. Instead of writing the annotation in one text field, the annotator provides answers to each question in a form (and they could write *N/A* for "no answer"). The answers were then concatenated into one complete annotation. The separate text fields are prompted with:

- What are the main actors (people, animals, etc.) in this GIF? (If none, write N/A.)
- What are the main actors (people, animals, etc.), if any, doing?
- What are the main actors (people, animals, etc.), if any, expressing (e.g., their emotions or expressions)?
- What is happening in the GIF (e.g., activities, events, actions)?
- If this GIF contains clips or actors from movies, television, or similar sources, provide their names, the source (if you know it), and any other relevant information.
- Is there text in this GIF? If so, what is the text?
- Is there any other information you would like to provide for describing this GIF? For example, if it is a meme, you can describe it.

This set of prompts was inspired by previous work [29] in which template-based tasks were easier for annotators and included more information compared to unstructured annotations.

In all three of the interface styles, we offered an example GIF and its annotation, as previous research suggested that providing high-quality examples could improve annotation quality [22, 35]. We also asked the annotator to "*describe when and how people might use the annotated GIF with an example*" to inform BLV users about the context of use.

4.2 Collecting Annotations

To evaluate the three interface styles, we collected annotations generated from each style on Amazon Mechanical Turk. We chose 34 GIFs that exhibited a range of features, including whether cartoon or live action, memes, clips from movies, TV programs, or illustrations, whether characters were present, whether a storyline was conveyed, and whether text was present. Example GIFs are shown in Fig. 4.[9]

We crowdsourced three annotations for each interface style for each GIF, meaning each of 34 GIFs had nine total annotations, for $34 \times 9 = 306$ annotations in all. To recruit a diverse participant pool, we limited the number of annotations one Turker could provide to two. GIF orders were randomized to avoid learning. We paid participants $1 USD for each annotation they provided.

During data collection, we removed answers that were obviously unrelated to the annotation task (e.g., pasting unrelated text from other sources), and finally collected 306 annotations from 246 Turkers. Example annotations are provided in the appendixA.

[8] For now, we do not have quality control mechanism to validate that the revised version is better than the original, and leave it as a future work.

[9] We will provide the full list of GIFs in the code repository URL upon publication.

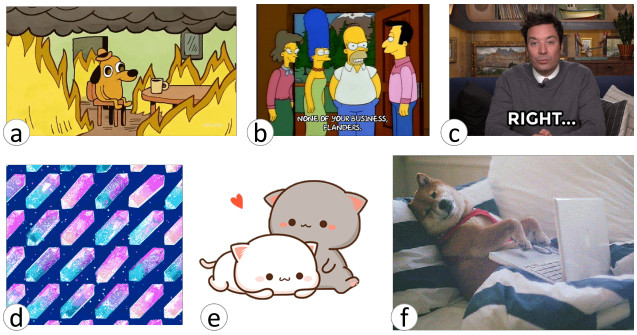

Figure 4: Example GIFs collected for the annotation tasks. (a) *This is fine* meme, including a cartoon dog drinking coffee in a room on fire, saying, "this is fine" (words not shown); (b) a clip from the cartoon *The Simpsons*; (c) a clip from *The Tonight Show* with Jimmy Fallon that contains the text "*right...*"; (d) patterened illustration of blue and purple diamonds; (e) a "sticker" with two cartoon cats; and (f) a video clip showing a dog typing rapidly on a laptop computer.

Table 1: Means (and standard deviations) for annotation length and completion time (for one annotation) for the three interface styles.

Task Design	Annotation Length (words)	Task Completion Time (seconds)
Freeform	56.8 (25.7)	1358.9 (885.3)
Semi-structured	60.1 (24.6)	998.9 (853.7)
Structured	57.2 (25.1)	1280.7 (921.5)

The descriptive results for each task design are shown in Table 1. We performed a one-way ANOVA on the *annotation length* and the *task completion time*, both of which were log-transformed to comply with the assumption of conditional normality [11]. We found that *Interface Style* had a significant effect on both annotation length ($F(2, 304) = 8.04, p < .001$) and task completion time ($F(2, 304) = 53.5, p < .001$). *Post hoc* pairwise comparisons, corrected with Holm's sequential Bonferroni procedure [18], indicated that the *semi-structured* style yielded significantly more words than both the *freeform* style ($t(34) = 3.79, p < .001$) and the *structured* ($t(34) = 3.03, p < .005$) style, with no significant difference between the *freeform* and *structured* styles ($t(34) = 0.73$, *n.s.*). As for task completion time, the *semi-structured* style took significantly less time than both the *freeform* ($t(34) = 10.13, p < .001$) and the *structured* ($t(34) = 6.83, p < .001$) styles, and the *structured* task took significantly less time than the *freeform* style ($t(34) = 3.24, p < .005$). On the whole then, it seemed the *semi-structured* interface style produced the most content in the least amount of time.

4.3 Evaluating Annotations with Sighted Users

We evaluated the collected GIF annotations by having both sighted users and BLV users rate the annotations and provide feedback. Our evaluation with sighted users was conducted on the Amazon Mechanical Turk platform. The task description was, "*Rate each description of the same GIF, as we wanted to use one of them to make people understand the content without seeing the GIF.*" For each GIF, we displayed one annotation from each interface style in a single page, and let the user rate each annotation in four respects:

- Informative (*does the annotation contain enough information and detail?*)
- Clear (*is the language style clear?*)
- Accurate (*does the annotation accurately describe the GIF content?*)
- Understandable (*is the annotation easy to understand?*)

For each GIF, there were $3 \times 3 \times 3 = 27$ annotation combinations, and we collected 9 ratings for each annotation. Each Turker was allowed to only rate at most five tasks to increase the number of distinct participants.

To evaluate the annotations for one GIF, participants first read its three annotations. Then, the associated GIF appeared after one minute. We delayed the unveiling of GIFs in this way to ensure that GIFs' visual depictions would not influence participants' judgments of their text annotations. Participants then provided their ratings for each annotation on a scale from 1 to 10, with "1" being "extremely negative" to "10" being "extremely positive." Participants could also write down their reasons for their ratings in an optional text box. The user interface for rating annotations is shown in Fig. 5.

Table 2: Means and standard deviations for ratings from sighted users of GIF annotations. The scale is 1-10, with "10" being the most positive.

	Freeform	Semi-structured	Structured
Informative	7.1 (2.4)	7.5 (2.3)	6.8 (2.5)
Clear	7.2 (2.3)	7.6 (2.2)	6.6 (2.5)
Accurate	7.4 (2.3)	7.7 (2.2)	7.1 (2.4)
Understandable	7.4 (2.4)	7.8 (2.2)	6.8 (2.6)

Overall, we collected $9 \times 9 \times 34 = 2754$ ratings for each annotation, the results for which are presented in Table 2. (Detailed distributions of results for each interface style are provided in Appendix D.) We performed analyses of variance based on mixed ordinal logistic regression [1, 16], treating each GIF id as a random factor to account for repeated measures. We found that *Interface Style* (freeform, semi-structured, structured) had a significant effect on how informative annotations were ($\chi^2(2, N = 2754) = 17.08, p < .001$), how clear annotations were ($\chi^2(2, N = 2754) = 30.95, p < .001$), how accurate annotations were ($\chi^2(2, N = 2754) = 17.34, p < .001$), and how understandable annotations were ($\chi^2(2, N = 2754) = 34.76, p < .001$).

Post hoc pairwise comparisons corrected with Holm's sequential Bonferroni procedure [18] indicated that for the informative rating, semi-structured annotations were significantly more informative than both *structured* ones ($Z = 4.15, p < .001$) and *freeform* ones ($Z = 2.54, p < .05$). For the clarity rating, *semi-structured* annotations were significantly clearer than both *structured* ones ($Z = 5.69, p < .001$) and *freeform* ones ($Z = 2.64, p < .05$), and *freeform* ones were significantly clearer than *structured* ones

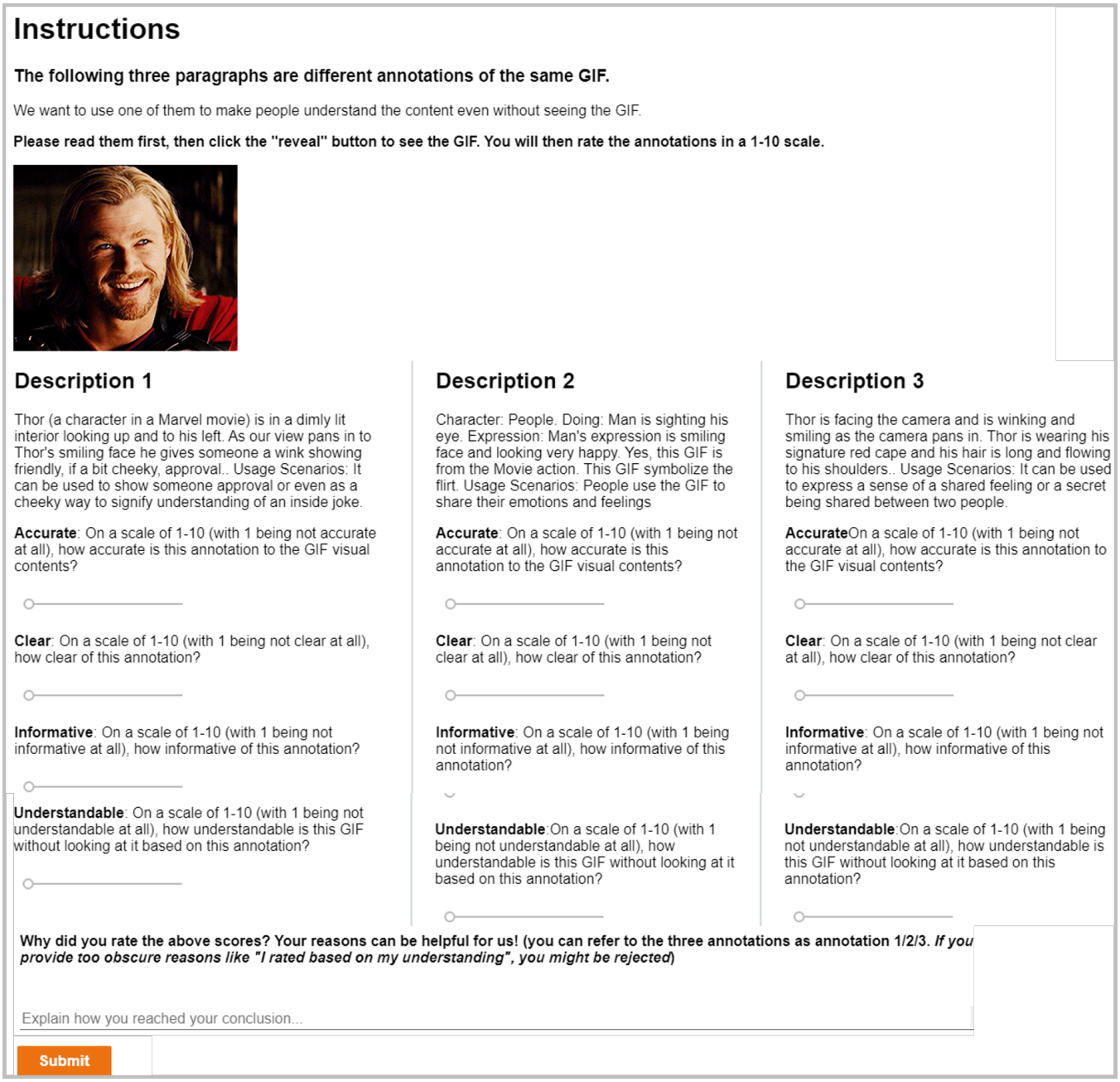

Figure 5: The annotation rating interface on Amazon Mechanical Turk. The GIF is revealed after one minute to encourage the rater to read the three descriptions first without being influenced by the GIF's appearance. Annotations from the three annotation interface styles are shown left-to-right as *semi-structured*, *structured*, and *freeform*.

($Z = 3.08, p < .01$). For the accuracy rating, *semi-structured* annotations were significantly more accurate than both *structured* ones ($Z = 4.30, p < .001$) and *freeform* ones ($Z = 2.44, p < .05$). For the understandability rating, *semi-structured* annotations were significantly more understandable than both *structured* ones ($Z = 6.04, p < .001$) and *freeform* ones ($Z = 2.82, p < .05$), and *freeform* annotations were significantly more understandable than *structured* ones ($Z = 3.26, p < .005$). Thus, annotations from the *semi-structured* interface style seemed to outperform other annotations in terms of their informativeness, clarity, accuracy, and understandability for sighted users.

4.4 Evaluating Annotations with Blind and Low-Vision Users

We also conducted a study with BLV users to evaluate our annotations from the three different interface styles. Specifically, we wanted to discover whether there were particular needs or preferences of BLV users when listening to GIF annotations, and whether their perceptions differed from those of sighted users.

We recruited 11 BLV users (4 women, 7 men, mean age = 32.7) via social media platforms and word-of-mouth. Participants' demographic information is shown in Table 3. Nine participants self-identified as fully blind and two identified as having low vision. All participants owned at least one smartphone and were familiar with screen readers. All participants had encountered animated GIFs on their smartphones, and understood what a GIF was. The study was conducted remotely via Zoom, and each participant was compensated $15 USD for the study, which took less than one hour to complete.

Table 3: Demographic information of BLV participants.

ID	Age	Gender	Visual Impairment	Phone Platform(s)
P1	25	Man	Fully blind	iOS
P2	35	Man	Fully blind	iOS + Android
P3	28	Woman	Fully blind	iOS
P4	28	Man	Fully blind	iOS + Android
P5	32	Man	Fully blind	iOS
P6	24	Woman	Fully blind	iOS + Android
P7	68	Woman	Low vision (central vision loss)	iOS
P8	32	Woman	Fully blind	iOS
P9	23	Man	Low vision (glaucoma)	iOS + Android
P10	36	Man	Fully blind	Android
P11	24	Man	Fully blind	iOS

As it was infeasible for each participant to read all 2754 annotations for the 34 GIFs, we manually selected five GIFs comprising 45 annotations for rating. [10] We sent a Google Sheet to the participant ahead of the study session, which contained all annotations, with their orders for each GIF randomized to avoid order effects. During the study, the researcher asked the participants to first listen to the nine annotations of a GIF, and then rate them one-by-one in four respects (they could re-listen to the annotation when rating it): *informative*, *clear*, *understandable* and *overall preference*. (We removed *accurate*, as participants could not see the GIFs to verify accuracy.) We explained the meaning of each rating category, and the rating was from 1-10, as described above for sighted users. After listening to all nine annotations of a GIF, the participants then typed their ratings into the Google Sheet.

We collected 45×11 = 495 ratings over all annotations, the results from which are shown in Table 4. (The distribution of ratings is provided in Appendix E.). We performed analyses of variance based on mixed ordinal logistic regression [1, 16], treating the GIF ID as a random factor to account for repeated measures. We found that *Interface Style* had a significant effect on the informative ($\chi^2(2, N = 495) = 25.66, p < .001$), clear ($\chi^2(2, N = 495) = 41.27, p < .001$), understandable ($\chi^2(2, N = 495) = 28.95, p < .001$) and preference ($\chi^2(2, N = 495) = 30.39, p < .001$) ratings.

[10] We selected various GIFs with different features as described in Section 4.2. The five GIFs are provided in Appendix B.

Table 4: Means and standard deviations for annotation ratings by BLV participants. The scale is 1-10, with "10" meaning the most positive.

	Freeform	Semi-structured	Structured
Informative	6.9 (2.5)	7.5 (2.3)	6.4 (2.5)
Clear	6.9 (2.6)	7.7 (2.2)	6.1 (2.5)
Understandable	7.0 (2.5)	7.6 (2.2)	6.3 (2.5)
Overall Preference	7.0 (2.5)	7.6 (2.2)	6.2 (2.5)

Post hoc pairwise comparisons corrected with Holm's sequential Bonferroni procedure [18] indicated that for the informative rating, *semi-structured* annotations were significantly more informative than both *structured* ones ($Z = 5.01, p < .001$) and *freeform* ones ($Z = 2.79, p < .05$). For the clarity rating, *semi-structured* annotations were significantly clearer than both *structured* ones ($Z = 6.31, p < .001$) and *freeform* ones ($Z = 2.87, p < .05$), and *freeform* ones were significantly clearer than *structured* ones ($Z = 3.56, p < .005$). For the understandability rating, *semi-structured* annotations were significantly more understandable than *structured* ones ($Z = 5.29, p < .001$), and the *freeform* ones were significantly more understandable than *structured* ones ($Z = 3.09, p < .01$). For participants' overall preference, *semi-structured* annotations were significantly preferred to both *structured* ones ($Z = 2.59, p < .05$) and *freeform* ones ($Z = 5.45, p < .001$), and *freeform* ones were significantly preferred to *structured* ones ($Z = 2.93, p < .01$). Thus, again we see the superiority of annotations coming from the semi-structured interface style, this time for BLV users.

During participant debriefing, we asked BLV participants for their rationale behind their ratings. Most participants did not enjoy the structured annotations. Although the structured descriptions contained important information listed as bullet points, the language style felt "too robotic" (P1) and "just like a collection of keywords" (P2). By contrast, the *semi-structured* annotations had a more natural style, and because there were guidelines provided, the quality of these annotations was perceived as higher than the *freeform* annotations. Participants generally appreciated that contextual or cultural information (e.g., the background of a movie character) was provided in certain annotations, and they also liked the description of usage scenarios for the GIFs, commenting that while the usage scenarios could be "subjective" (P8), knowing how GIFs could be used was helpful for knowing when to send it to others or post it online. These findings were generally consistent with findings from prior work on image annotation [13].

Both evaluations with sighted and BLV users showed similar results: the annotations from the *semi-structured* interface style were better than the *freeform* and *structured* ones. We therefore incorporated the *semi-structured* interface style into Ga11y as its web-based human annotation interface for annotating animated GIFs.

5 IMPLEMENTATION OF GA11Y

In this section, we present the implementations of the three major components of the Ga11y system: the Android client, the annotation web interface, and the backend server. We provide these details for completeness and reproducibility, and because realizing Ga11y required solving certain technical problems that constitute their own contribution.

5.1 The Android Client

To enable end users request GIF annotations, we developed an Android client that could (1) detect on-screen GIF elements, and (2) record animated GIFs and communicate with the Ga11y server.

In order to monitor on-screen contents for GIFs and insert buttons for requesting annotations, we adopted an "interaction proxy" [46] built on the Android Accessibility API. Specifically, we implemented an accessibility service that listens to screen updates signified by system *AccessibilityEvents* and captures the view hierarchy of the current app. By comparing the elements within the view hierarchy to a predetermined set of heuristics, we were able to identify GIF elements contained within supported apps[11] because of the app-specific UI structures. Our heuristics mainly drew upon the *ClassName*, *ViewIdResourceName*, and *ContentDescription* attributes of an element and its child elements, if any. These heuristics were determined by monitoring unique properties of GIF elements within each app using the Android Accessibility API. For example, in Facebook Messenger, an element is considered to be a GIF when its bounding box contains one *ViewGroup* with the *ContentDescription* of "Sent photo message" and an *ImageView* with the *ContentDescription* of "Forward button."

Once a GIF element is identified, we insert a virtual request button immediately after it in the focus order. The inserted button would normally be announced by Android's TalkBack feature as a regular button (i.e., "Get GIF annotation, button"), but the button is controlled and monitored by our Android client instead of the current app. As a result, the user is prompted to "double-tap to request annotation." The user either double-taps to initiate a request for the corresponding GIF element, or continues left or right swiping to ignore the prompt, which would switch the focus back to the foreground app so that TalkBack can properly return to the app's element tree.

When the request button is triggered via double-tap, Ga11y initiates continuous screen capture using the Android MediaProjection API. This API captures the entire screen and sends it to our client; we then crop out only the GIF within the bounds of the identified element so as to preserve user privacy and reduce memory usage. We wait a fixed amount of time before ending the capture, determined to be five seconds in our studies. The captured image sequence is then compressed and sent to Ga11y's Annotation Server for further processing and analysis, as described below in Section 5.3. To prevent accidental actions that might move or occlude the GIF during our recording, the client produces an intermittent beep sound at twice a second to indicate that capture and analysis is in progress. If the user interacts with the device during this period of time, we consider the annotation request to be canceled and discard the captured images. Otherwise, the client will announce when the server returns its annotation to the user, which takes around 5 - 20 seconds depending on the network connection.

To speed up processing for previously annotated GIFs, we implemented a local cache in the client. After a successful annotation is returned by the server, we store the image hash values for each GIF image sequence, calculated using an average perceptual hash function [23].[12] We also store the returned annotation along with the hash values. Every time the user initiates an annotation request and the image capture starts, each captured frame is hashed using the same hashing function and the result is compared against all encountered image hashes. Once we find at least three successful matches, we consider the two sequences to be from the same GIF. The client then stops the screen capture and announces the cached annotation. For a previously encountered GIF, this reduces the annotation time to around one second.

The Ga11y client utilizes two special permissions: the accessibility service permission and screen capture permission. The user is asked to grant these two permissions after the Ga11y app is launched. If the user chooses not to grant either permission, the app will stay on the permission request screen and will not be able to provide GIF annotations. To support both English and Chinese annotations as used in our studies, the Ga11y client adapts to the system language based on the device's language and locale settings.

5.2 The Annotation Website

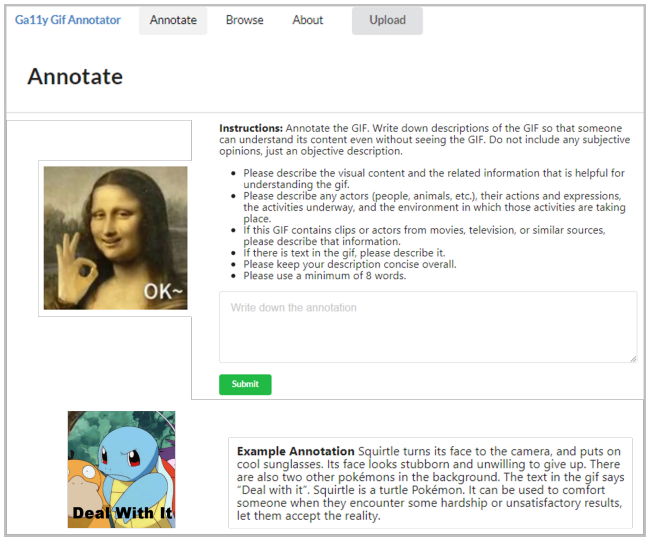

Figure 6: Ga11y's web-based annotation interface, which applies the *semi-structured* interface style based on our evaluation study described in Section 4.

We implemented the annotation website using the React framework.[13] The annotation website contains two main pages (Fig. 3): the *annotate* page, for browsing the GIFs that are requested by

[11] We currently support Facebook Messenger, Twitter, WeChat, Telegram, and Discord.

[12] A perceptual hash function generates the same hash values for images that look similar, which is suitable for comparing the visual similarity among a set of images. We used the average hash function because it was fast and was not sensitive to pixel noise generated during screen recordings.

[13] React: https://reactjs.org/

users but have not been annotated yet, and the *browse* page, for browsing and editing existing GIF annotations. One can click a GIF on the *annotate* page to go to the annotation interface, which displays the *semi-structured* interface style with an open text box accompanied by annotation instructions, together with an example annotation (Fig. 6). The user can also upload their own GIFs with accompanying annotations via the website by clicking the "upload" button.

5.3 The Annotation Server

Ga11y's backend server was implemented using the Tornado framework.[14] The server is responsible for two main tasks: (1) matching the requested GIF in the annotation database, and (2) requesting automated annotations for unmatched GIFs. We describe the various functions that the server performs.

GIF Reconstruction. To handle an annotation request from an Android client, Ga11y's server first assembles captured screenshots into an animated video. As the GIF is captured directly from the screen, there is no indication about timing (e.g., the start and ending frame of the GIF). We therefore designed Algorithm 1 to identify the loop based on a sequence of GIF snapshots. For each snapshot of the GIF, we calculate its hash value by using the average perceptual hash function. We then put frames with the same hash value into the same bin, and derive the loop interval of the GIF based on the bins. Identifying the loop is important for the crowdworkers to annotate the reconstructed GIFs in the web client.

Algorithm 1 Identifying the loop from a sequence of GIF snapshots

```
frame_hash_dict ← {}
for frame in GIF.Snapshots do
    hash ← PerceptionHash(frame)
    if hash not in frame_hash_dict then
        frame_hash_dict[hash] ← [index_of_frame]
    else
        frame_hash_dict[hash].append(index_of_frame)
    end if
end for
duplicate_frames ← all items that have more than two values in
frame_hash_dict
if duplicate_frames is empty then
    no loop in the GIF
else
    cnt ← the most frequent count of the item in duplicate_frames

    loop_frames ← items whose count equals cnt in dupli-
    cate_frames
    intervals ← time difference between consecutive pairs in each
    item of loop_frames
    loop_interval ← max(intervals)
    (loop_start_frame, loop_end_frame) ← the frame pair whose
    time difference is loop_interval
end if
```

[14] Tornado Web Server: https://www.tornadoweb.org/en/stable/

After the loop is identified, we further interpolate the GIF with snapshots that are outside the loop timespan to minimize any effects of lag from screen capture on the mobile client.

GIF Comparison. Because of the loop detection algorithm, the same GIF content might yield two loops starting at different frames. We thus used multiple frames within a GIF loop for comparison to increase robustness. We extract several keyframes of the GIF after reconstructing it by splitting the GIF loop into equal-lengthed subclips and taking the first frame of each clip as the keyframe. To compare whether two GIFs are equal, we compare the perceptual hashes of the two GIFs' keyframes: if any pair of the frame hash are identical, we treat the two GIFs as visually similar. We decided to have four keyframes for each GIF, as this number was empirically sufficient to identify similar GIFs and distinguish different ones.

GIF Annotation Database. After extracting the keyframes from the requested GIF, the server tries to match the keyframes with existing GIFs in the database. For each requested GIF, we store the perceptual hashes of its keyframes in a local Elasticsearch server.[15] If any of the keyframe's hash match an existing item in the database, the server will request the corresponding annotation; if not, the server will first store the keyframe hashes in the database, and request automated annotations for the GIF. The reconstructed GIFs are stored as videos in an AWS S3 database; all annotations are stored on the AWS DynamoDB database, which can be directly updated via the web annotation interface.

Requesting Automated Annotation. If a requested GIF has not yet been manually annotated, the server will request automated annotations by sending one of the keyframes to the Google Vision service[16] and the Microsoft Azure Computer Vision service. [17] The former service provides the objects and text in the image, while the latter service generates a caption of the image. The server then combines the two recognition results into a single description as Ga11y's automated annotation. We used the Google Translation API [18] to translate the annotation to other languages, namely Chinese in our user study.

6 GA11Y EVALUATION

We conducted a user study to evaluate Ga11y. Specifically, we were interested in three questions: (1) How do BLV users experience Ga11y's usability? (2) How do BLV users perceive the quality of human-labelled and machine-generated annotations? (3) How might Ga11y affect BLV users' online communication experiences? We describe our study to answer these questions below.

6.1 Participants

We recruited 12 BLV participants, with four from the United States (P2, P4, P9, and P10 from the first study in Table 3) and eight from China, whose demographic information is listed in Table 5. Participants' average age was 29.1 years (SD = 5.6). All participants were familiar with GIF images generally and with using a mobile screen reader. We recruited Chinese participants to understand how Ga11y performs beyond the English language, including for pictographic

[15] https://www.elastic.co/
[16] https://cloud.google.com/vision
[17] https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/
[18] https://cloud.google.com/translate

languages. Participants received $25 USD or 150 CNY for the one hour study.

Table 5: Demographic information for our study participants. All participants owned an Android device.

ID	Age	Gender	Visual Impairment
CN_P1	23	Man	Fully blind
CN_P2	24	Man	Low vision (cataract)
CN_P3	30	Man	Low vision (traumatic visual loss)
CN_P4	26	Man	Low vision (cataract)
CN_P5	29	Man	Low vision (amaurosis)
CN_P6	40	Man	Fully blind
CN_P7	31	Woman	Fully blind
CN_P8	24	Man	Low vision (optic atrophy)
US_P1	35	Man	Fully blind
US_P2	28	Man	Fully blind
US_P3	23	Man	Low vision (glaucoma)
US_P4	36	Man	Fully blind

6.2 Apparatus

We hosted our Ga11y service on a server from a research institute, which had a public IP address to which participants' clients could send requests. All participants used the Ga11y service on their own Android devices. To send GIF content, for American participants, we used Twitter as the communication platform; for Chinese participants, we used WeChat.

6.3 Procedure

Study sessions were conducted remotely via Zoom, and were audio-recorded for further analysis. We asked participants to turn up their screen reader volume so that the experimenter could hear the output. We first sent participants the Android client application package (APK) and instructed them how to install and configure it. We then asked participants several questions regarding their existing experiences and difficulties with online GIFs. After set-up was complete, we sent a test GIF to participants and let them try making an annotation request. After participants confirmed that they were familiar with the Android client, we began the formal study session.

During the formal part of the study, we sent participants two groups of GIFs, five that were already annotated by Turkers from the first study, and five that were not manually described and would therefore elicit machine-generated annotations. For context, the machine-generated annotations provided for items (f) - (j) in Figure 7 always began with "automatically generated description"; their detailed annotation is provided in Appendix C. All GIFs in Figure 7 were chosen by the experimenters so that they exhibited different features as specified in Section 4.2. The order of the GIFs were randomized for each participant. For each GIF sent to a participant, the participant was told to operate their screen reader to perform an annotation request. After all the annotations were requested, participants rated Ga11y on the System Usability Scale (SUS) [8]. Participants were also interviewed for their feedback on using the Ga11y system. Participants were encouraged to continue using the system after the study, and to provide their feedback when using it, although this was optional.

Figure 7: Ten GIFs used for the evaluation task. GIFs (a) – (e) had human annotations already on the server, whereas (f) – (j) did not. (a) Cartoon cat sticker with text; (b) the infinite dog; (c) Jimmy Fallon with text; (d) "This is fine" meme; (e) rapidly typing dog; (f) vibing cat; (g) telescope sticker with text; (h) smiling *SpongeBob SquarePants*; (i) plants growing; (j) cartoon rabbit sticker with text.

7 RESULTS

In this section, we present our study results. In general, all participants were positive about Ga11y and said they would like to have it on their phone. We calculated the SUS score on a 0 – 100 scale, where higher scores indicate "more usable."[19] The average SUS score was 89.1 (SD = 9.0), indicating high usability of the system. We present the detailed results in this section.

7.1 Current Experience with GIFs

All participants said that they had encountered GIFs online, and most GIFs were not understandable because of the lack of annotations. US_P2 mentioned that he had encountered many GIFs on Twitter, and although some of them had descriptions, most of the descriptions were automatically generated and only contained one or two keywords, which was not helpful. US_P3 mentioned that with keyboards like Google's *Gboard*, he could select and send GIFs. However, the problem was that while the GIFs had keyword annotations within the keyboard, their annotations were lost once the GIFs were sent to a chat. All participants from China mentioned that they encountered GIFs and stickers every day as people in group chats liked to send funny GIFs and compete with each other, which has been referred to as "sticker competitions" [48]. However, because of the lack of annotations, our participants have not been able to participate in such activities. CN_P2 mentioned that he only used certain GIFs for simple expressions that he was familiar with, such as "thank you" and "okay," and CN_P1 mentioned that he only used emojis and not GIFs, as emojis are better annotated.

Participants tried to ask a sighted friend for help when they encountered unfamiliar GIF contents. As CN_P4 said, "I will just ask others the meaning of the GIF they sent if I don't understand." However, most of the time participants reported just ignoring the GIFs and stickers they encountered.

[19] https://measuringu.com/sus/

7.2 Ga11y Usage "In the Wild"

To see how Ga11y might help participants in their daily internet usage, we logged each participant's usage of Ga11y for three days after the formal study, with their knowledge and permission. As noted above, this part of the study was optional, but we encouraged participants to continue to use Ga11y after the study session. In total, our Ga11y server received 548 annotation requests, with 458 requests from Chinese participants and 90 from American participants. Among them, 203 requests included new GIFs that were not annotated in the server, and 345 requests included GIFs that were already in the database. For the newly requested GIFs, after the machine-provided annotations were sent, we, the researchers, provided the subsequent human annotations through Ga11y's web interface. We also assigned a unique ID for each participant in the request log, finding that each user sent about 13.5 requests per day on average (SD = 16.6), indicating that Ga11y was frequently used and could potentially play an important role in participants' online communications.

7.3 Qualitative Feedback

We collected participants' feedback after the formal study and during the three-day "in the wild" usage period, and coded the feedback using affinity diagramming [34]. Three main topics emerged: *Perceptions of human- and machine-generated annotations*, *usability suggestions*, and *the effects of Ga11y on online communication*. We take each of these in turn.

7.3.1 Perceptions of Human- and Machine-Generated Annotations. All participants felt that the human-generated annotations were helpful in understanding animated GIFs, especially when context, such as the background of a movie or cartoon character, was provided. Participants also appreciated the provision of machine-generated annotations, although these descriptions were thought to lack a "human touch" (US_P1). Two participants also appreciated that there was always a prefix "automatically generated description" for the machine-generated annotations, allowing them to adjust their expectations of the annotation quality.

However, while the four participants from the U.S. were all comfortable with the detailed level of the annotations, three Chinese participants commented that the translated annotations were too long and contained too much detail. For example, CN_P8 said that some of the detail could be omitted, and the language style of the transcribed annotations was not very natural. On the other hand, all participants appreciated having human volunteers providing annotations. They also appreciated the explicit usage scenarios offered in the annotations, even though the usage scenarios were clearly "subjective" (CN_P8).

7.3.2 Usability Suggestions. All participants found the interaction with Ga11y's Android client interface for requesting annotations to be intuitive, where they could easily choose whether to request an annotation or not. CN_P1 said, "The interaction [of pressing a button alongside a GIF] is far easier than other accessibility apps," referring to other image recognition apps in which he had to "take a screenshot and switch to the app for recognition results." Two participants (CN_P3, CN_P7) also suggested that the Ga11y service could automatically request the annotations, and attach annotated labels to the GIFs without the user having to request them interactively. We leave implementing this option for future work.

As two participants mentioned that the annotations were too lengthy, they suggested adding functions to adjust the detail level of the annotations. CN_P6 suggested that the system could first read out a brief version of the annotation, including only the most important content such as the main characters and the meaning of the GIF. Subsequently, the user could click a button to listen to the full annotation if they wanted. CN_P8 suggested a similar idea, where the user could adjust the level of annotation detail in the app settings.

7.3.3 The Effects of Ga11y on Online Communication. All participants appreciated Ga11y for enabling them understand unlabelled GIFs. As US_P3 mentioned, "Although sometimes the annotations are automatically generated, it offers information which is not accessible at all before." US_P1 said that he was a user of many social media platforms, and "it is important to speak the language others are speaking." CN_P4 also talked about the helpfulness of being able to understand GIFs: "I was very careful about using and sending new GIFs, as I am worried that I might send something inappropriate. And each time I find others are sending stickers and GIFs in the group chat, I feel left behind. Having the annotations can definitely help me understand what they are saying, and give me the sense of belonging." Taken together, our results indicate that although Ga11y could certainly be improved, it had a positive impact on the accessibility of online GIFs and participants' online communication generally.

8 DISCUSSION

In this work, we presented Ga11y, a GIF annotation system combining the power of crowdsouring with machine intelligence. With the three components of the Ga11y system, we were able to not only provide on-demand GIF annotations to BLV users on their mobile devices, but also to provide a *semi-structured* annotation interface style that yields high quality GIF annotations. On the client side, we utilized the accessibility framework on the Android platform, and created an interaction proxy that allowed users to request on-screen contents without switching the application; on the server side, we designed a comparison algorithm for handling screen-recorded GIFs. By conducting a study with both English and Chinese language speakers, we validated the usefulness and usability of Ga11y for enabling GIF accessibility.

From the study results, we found that the current computer vision based techniques usually only generated high-level annotations without enough details, and many of the characters/objects were misrecognized [20], which confused participants. This also indicated that although for static images, auto-generated descriptions could be of high quality and were already used on commercial platforms (such as iOS and Chrome), they were not capable to handle the GIF content yet. On the other hand, participants appreciated the timeliness of the machine-generated annotation, and found it was especially useful for GIFs containing text, as the text could convey important information even if the visual content was misrecognized.

[20] Annotation examples are shown in Appendix C

As for the human-generated annotations, we found it interesting that two participants complained that the annotations were too lengthy and detailed, as they were more interested in what the GIF tried to convey, rather than the content itself. Two participants also commented that they needed to listen to the same annotation multiple times, as the text was too long to remember. As a future step, it is worthwhile to investigate how users' annotation preferences vary for different usage scenarios (e.g. online messaging, social media post, blog/article, etc).

We also found that the semi-structured prompt for annotating GIFs was recognized as providing the highest description quality by both sighted and BLV people, compared to structured and freeform prompts. This finding contradicts the claim of previous work [29], where the authors found structured prompts yielded better annotations for scientific figures. One possible explanation is that structured annotations sound less "human", and contain many redundant information [26] in comparison to the semi-structured annotations. In addition, our participants commented that since GIFs often contained nuanced expressions, or culture-related background information, it was important to have a "human touch" in the explanation. Hence the annotation design can be deeply situated in the task context: for contents that has objective descriptions such as scientific figures or charts, structured annotation might provide ease to both annotators and readers; for contents that are subjective and require personalized explanations, the semi-structured annotation design might be the best.

The technical solution provided by this paper did not fully investigate the privacy issues, and we would expect a more sophisticated way to handle the user data in the future. For now, the screenshots are sent to the remote server and displayed in the web client. This approach might leak the users' browsing data, or the source app they are using to capture the GIFs. In the future, a local cache containing hash + descriptions of most popular GIFs might mitigate the problem, as most of the requests can be processed offline. A better GIF recognizing algorithm can also be applied to crop unrelated portions of the screenshots.

The popularity of the GIF image format has created a unique aspect of internet culture, and the ability to understand GIFs well is a key to participating in that culture. As GIFs often contain subtle emotions and rich expressions, it is necessary to generate human annotations to convey human feelings. By providing a publicly accessible annotation service, we can also raise awareness of the need for GIF accessibility. Similarly, GIF platforms such as GIPHY [21] and Tenor [22] could also provide ways for users to annotate a GIF when uploading it or selecting it for use. These platforms already contain user-generated metadata about GIFs such as tags, which could be utilized to train machine learning models that generate better annotations.

Although the annotations in Ga11y are provided as text, there are richer ways to represent a GIF, such as with audio. Cole et al. [13] suggested that including an audio description, such as any sound accompanying the source clip of a GIF, could enrich emotive understanding. Future versions of Ga11y could also employ automated methods to match GIFs with existing videos and extract the audio as part of the annotation. Of course, the use of audio also poses certain accessibility barriers, and would need to be addressed, perhaps again leveraging text like in Ga11y.

[21] https://giphy.com/
[22] https://tenor.com/

9 LIMITATIONS

As with any research project, there are several limitations of this work. First, because we used the Android accessibility framework, participants found the setup of the app to be complex, as they needed to go through multiple permission-granting steps. Furthermore, the screen capture service we used for GIF recording was not entirely stable and could be shut down by the system unexpectedly. Capturing the whole screen also raised some privacy concerns by participants. The experience of Ga11y could have been smoother if the operating system supported annotation requests natively. Second, Ga11y only supported GIF recognition in a limited set of mobile apps by recognizing their user interface structures. This limitation could be improved if there were a universal interaction that could be employed, such as a hard button or a gesture, to trigger Ga11y's screen recording. In this way, the user could perform annotation requests on any screen element, and there would be no need to recognize on-screen GIFs. Third, although we logged the annotation service usage during the "in the wild" period of three days after the study sessions, conducting a long-term field deployment would reveal more insights on how participants use Ga11y in their daily lives. Fourth, as Ga11y's annotation web interface is not yet publicized, it remains an open question as to whether people are willing to contribute annotations, and the annotation quality on a large scale remains an open question. Finally, the client was implemented on the Android system, thus we did not gather the feedback from iOS users. However, we are aware of the huge BLV population using VoiceOver as their main screen readers, and the experience might be different from the Android TalkBack. That said, the iOS system is very strict on the accessibility frameworks and third party apps, hence we chose Android as the main platform.

10 CONCLUSION

In this work, we presented *Ga11y*, a GIF annotation system that utilizes both crowdsroucing and machine intelligence to help blind and low vision (BLV) users understand the content and meaning of animated GIFs. Ga11y contains three components, including a mobile client, a data processing and storage server, and a GIF annotation website. In order to have high quality annotations, we evaluated different annotation interface styles with both sighted and BLV users, and applied the best style—one utilizing an open text field with guiding prompts—on Ga11y's annotation website. We also implemented GIF processing for our Android client, including GIF recording, reconstruction, and comparison, to support annotation requests. Our user study with both American and Chinese participants demonstrated that Ga11y was perceived as highly usable, and the combination of human- and machine-generated annotations was an effective solution for aiding in GIF understanding. We hope that by open-sourcing our implementation, Ga11y will encourage GIF platforms and system providers to consider designing annotation supports for GIFs and stickers, and maybe for other forms of dynamic content. Everyone should be able to fully participate in the online culture enabled by animated GIFs.

ACKNOWLEDGMENTS

We thank Amy X. Zhang for providing suggestions on annotation task designs, Qisheng Li for providing feedback on the early paper draft, Mengqi Li for offering participant resources. This work was funded in part by the UW CREATE Center and the National Science Foundation under award IIS-1702751. Any opinions, findings, conclusions or recommendations expressed in our work are those of the authors and do not necessarily reflect those of any supporter.

REFERENCES

[1] A. Agresti. 2010. *Analysis of Ordinal Categorical Data.* Wiley, Hoboken, NJ, USA. https://books.google.com/books?id=VVIe4BPDR7kC

[2] Dragan Ahmetovic, Daniele Riboli, Cristian Bernareggi, and Sergio Mascetti. 2021. RePlay: Touchscreen Interaction Substitution Method for Accessible Gaming. In *Proceedings of the 2021 International Conference on Mobile Human-Computer Interaction*. Association for Computing Machinery, New York, NY, USA, 1–13.

[3] Abdullah X. Ali, Meredith Ringel Morris, and Jacob O. Wobbrock. 2018. Crowdsourcing Similarity Judgments for Agreement Analysis in End-User Elicitation Studies. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 177–188.

[4] Abdullah X. Ali, Meredith Ringel Morris, and Jacob O. Wobbrock. 2019. Crowdlicit: A System for Conducting Distributed End-User Elicitation and Identification Studies. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12.

[5] Saeideh Bakhshi, David A. Shamma, Lyndon Kennedy, Yale Song, Paloma de Juan, and Joseph 'Jofish' Kaye. 2016. Fast, Cheap, and Good: Why Animated GIFs Engage Us. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 575–586.

[6] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 313–322.

[7] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010. VizWiz: Nearly Real-Time Answers to Visual Questions. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 333–342.

[8] John Brooke. 1996. SUS: a quick and dirty usability scale. *Usability evaluation in industry* 1 (1996), 189.

[9] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Comput. Surv.* 51, 1, Article 7 (Jan. 2018), 40 pages.

[10] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the Crowd Yields Better Work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (Seattle, Washington, USA) *(CSCW '12)*. Association for Computing Machinery, New York, NY, USA, 1013–1022.

[11] Limpert Eckhard, Werner A. Stahel, and Markus Abbt. 2001. Log-normal Distributions across the Sciences: Keys and Clues. *BioScience* 51, 5 (2001), 341–352. http://www.jstor.org/stable/10.1641/0006-3568(2001)051[0341:lndats]2.0.co;2

[12] Jason Eppink. 2014. A brief history of the GIF (so far). *Journal of Visual Culture* 13, 3 (2014), 298–306.

[13] Cole Gleason, Amy Pavel, Himalini Gururaj, Kris Kitani, and Jeffrey Bigham. 2020. Making GIFs Accessible. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, Greece) *(ASSETS '20)*. Association for Computing Machinery, New York, NY, USA, Article 24, 10 pages.

[14] Cole Gleason, Amy Pavel, Xingyu Liu, Patrick Carrington, Lydia B. Chilton, and Jeffrey P. Bigham. 2019. Making Memes Accessible. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA) *(ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 367–376.

[15] Darren Guinness, Edward Cutrell, and Meredith Ringel Morris. 2018. Caption Crawler: Enabling Reusable Alternative Text Descriptions Using Reverse Image Search. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–11.

[16] Donald Hedeker and Robert D. Gibbons. 1994. A Random-Effects Ordinal Regression Model for Multilevel Analysis. *Biometrics* 50, 4 (1994), 933–944.

[17] Amanda Hess and Quoctrung Bui. 2017. What Love and Sadness Look Like in 5 Countries, According to Their Top GIFs. *The New York Times* 1, 1 (Dec. 2017), 1 pages. https://www.nytimes.com/interactive/2017/12/29/upshot/gifs-emotions-by-country.html

[18] Sture Holm. 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics* 6, 2 (1979), 65–70.

[19] Jialun "Aaron" Jiang, Jed R. Brubaker, and Casey Fiesler. 2017. Understanding Diverse Interpretations of Animated GIFs. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 1726–1732.

[20] Jialun "Aaron" Jiang, Casey Fiesler, and Jed R. Brubaker. 2018. "The Perfect One": Understanding Communication Practices and Challenges with Animated GIFs. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 80 (Nov. 2018), 20 pages.

[21] Annika Kaltenhauser, Naundefineda Terzimehić, and Andreas Butz. 2021. MEMEography: Understanding Users Through Internet Memes. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 332, 7 pages.

[22] Masaki Kobayashi, H. Morita, Masaki Matsubara, N. Shimizu, and Atsuyuki Morishima. 2018. An Empirical Study on Short- and Long-Term Effects of Self-Correction in Crowdsourced Microtasks. In *HCOMP*. AAAI Press, Palo Alto, CA, USA, 79–87.

[23] Neal Krawetz. 2011. Looks Like It - The Hacker Factor Blog. https://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html

[24] Toby Jia-Jun Li, Igor Labutov, Xiaohan Nancy Li, Xiaoyi Zhang, Wenze Shi, Wanling Ding, Tom M. Mitchell, and Brad A. Myers. 2018. APPINITE: A Multi-Modal Interface for Specifying Data Descriptions in Programming by Demonstration Using Natural Language Instructions. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. Association for Computing Machinery, New York, NY, USA, 105–114.

[25] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. arXiv:1604.02748 [cs.CV]

[26] Kelly Mack, Edward Cutrell, Bongshin Lee, and Meredith Ringel Morris. 2021. Designing Tools for High-Quality Alt Text Authoring. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, USA) *(ASSETS '21)*. Association for Computing Machinery, New York, NY, USA, Article 23, 14 pages.

[27] Haley MacLeod, Cynthia L. Bennett, Meredith Ringel Morris, and Edward Cutrell. 2017. Understanding Blind People's Experiences with Computer-Generated Captions of Social Media Images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 5988–5999.

[28] Kate M. Miltner and Tim Highfield. 2017. Never Gonna GIF You Up: Analyzing the Cultural Significance of the Animated GIF. *Social Media + Society* 3, 3 (2017), 2056305117725223.

[29] Valerie S. Morash, Yue-Ting Siu, Joshua A. Miele, Lucia Hasty, and Steven Landau. 2015. Guiding Novice Web Workers in Making Image Descriptions Using Templates. *ACM Trans. Access. Comput.* 7, 4, Article 12 (Nov. 2015), 21 pages.

[30] Meredith Ringel Morris, Jazette Johnson, Cynthia L. Bennett, and Edward Cutrell. 2018. Rich Representations of Visual Content for Screen Reader Users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–11.

[31] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What Do People Ask Their Social Networks, and Why? A Survey Study of Status Message Q&A Behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1739–1748.

[32] Meredith Ringel Morris, Annuska Zolyomi, Catherine Yao, Sina Bahram, Jeffrey P. Bigham, and Shaun K. Kane. 2016. "With Most of It Being Pictures Now, I Rarely Use It": Understanding Twitter's Evolving Accessibility to Blind Users. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 5506–5516.

[33] André Rodrigues, André Santos, Kyle Montague, and Tiago Guerreiro. 2017. Improving Smartphone Accessibility with Personalizable Static Overlays. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (Baltimore, Maryland, USA) *(ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 37–41.

[34] RAYMOND SCUPIN. 1997. The KJ Method: A Technique for Analyzing Data Derived from Japanese Ethnology. *Human Organization* 56, 2 (1997), 233–237.

[35] Rachel N. Simons, Danna Gurari, and Kenneth R. Fleischmann. 2020. "I Hope This Is Helpful": Understanding Crowdworkers' Challenges and Motivations for an Image Description Task. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 105 (Oct. 2020), 26 pages.

[36] Todd Spangler. 2019. Giphy Video Launches: GIF-Sharing Platform Adds Audiovisual Clips From Universal Pictures, Other Partners. https://variety.com/2019/digital/news/giphy-video-launch-universal-pictures-1203430165/

[37] Abigale Stangl, Meredith Ringel Morris, and Danna Gurari. 2020. "Person, Shoes, Tree. Is the Person Naked?" What People with Vision Impairments Want in Image Descriptions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY,

USA, 1–13.

[38] TechCrunch. 2018. Tenor hits 12B GIF searches every month. https://social.techcrunch.com/2018/02/20/tenor-hits-12b-searches-in-its-gif-keyboard-every-month/

[39] Twitter. 2021. How to make images accessible for people. https://help.twitter.com/en/using-twitter/picture-descriptions

[40] W3Techs. 2021. Usage Statistics of GIF for Websites, August 2021. https://w3techs.com/technologies/details/im-gif

[41] Ruolin Wang, Zixuan Chen, Mingrui Ray Zhang, Zhaoheng Li, Zhixiu Liu, Zihan Dang, Chun Yu, and Xiang 'Anthony' Chen. 2021. Revamp: Enhancing Accessible Information Seeking Experience of Online Shopping for Blind or Low Vision Users. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 494, 14 pages.

[42] Yuan Wang, Yukun Li, Xinning Gui, Yubo Kou, and Fenglian Liu. 2019. Culturally-Embedded Visual Literacy: A Study of Impression Management via Emoticon, Emoji, Sticker, and Meme on Social Media in China. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 68 (Nov. 2019), 24 pages.

[43] Richard Yao. 2018. The Surging Popularity of GIFs In Digital Culture. https://medium.com/ipg-media-lab/the-enduring-popularity-of-gifs-in-digital-culture-54763d7754aa

[44] Mingrui Zhang, Alex Mariakakis, Jacob D. Burke, and Jacob O. Wobbrock. 2021. A Comparative Study of Lexical and Semantic Emoji Suggestion Systems. In *iConference*. Springer Nature, London, UK, 229–247.

[45] Mingrui Ray Zhang, Ruolin Wang, Xuhai Xu, Qisheng Li, Ather Sharif, and Jacob O. Wobbrock. 2021. Voicemoji: Emoji Entry Using Voice for Visually Impaired People. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 37, 18 pages.

[46] Xiaoyi Zhang, Anne Spencer Ross, Anat Caspi, James Fogarty, and Jacob O. Wobbrock. 2017. Interaction Proxies for Runtime Repair and Enhancement of Mobile Application Accessibility. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 6024–6037.

[47] Xiaoyi Zhang, Anne Spencer Ross, and James Fogarty. 2018. Robust Annotation of Mobile Application Interfaces in Methods for Accessibility Repair and Enhancement. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 609–621.

[48] Rui Zhou, Jasmine Hentschel, and Neha Kumar. 2017. Goodbye Text, Hello Emoji: Mobile Communication on WeChat in China. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 748–759.

A SAMPLE ANNOTATIONS COLLECTED FROM MTURK WORKERS

Annotations for Fig 8 from different task designs:

Freeform Thor is facing the camera and is winking and smiling as the camera pans in. Thor is wearing his signature red cape and his hair is long and flowing to his shoulders.. Usage Scenarios: It can be used to express a sense of a shared feeling or a secret being shared between two people.

Semi-structured Chris Hemsworth (Thor) is smiling widely with his teeth as the camera zooms in on his face, he winks.. Usage Scenarios: This can be used for a situation where someone might want to flirt and express the wink to the person that they're sending it to,

Structured Character: Thor, a Greek God. Doing: Thor is winking his eye. Expression: Joy or an inside joke. Activity: Thor is expressing happiness and smiling. Thor is played by Chris Hemsworth and this character is from the film series the Avengers by Marvel.. The gif is a happy moment of Thor smiling and winking. Usage Scenarios: People might use this gif to express that they understand an inside joke

B FIVE GIFS FOR ANNOTATION EVALUATION WITH BLV USERS

Figure 8: GIFs for the annotation evaluation study with BLV users. (a) A cartoon eye opening; (b) a clip from the movie *thor*; (c) a sticker of light bulb; (d) a clip from an old movie; (e) a cartoon clip with a speedy driving meme

C SAMPLE MACHINE-GENERATED ANNOTATIONS

The machine generated annotations are described as follows for Figure 4. There are many misrecognized objects and scenarios.

(f) "Automatically generated caption is a close-up of a person's face. Best guess is lip. The labels are Glasses and Vision care and Eyelash";

(g) "Automatically generated caption is logo, company name. Best guess is metal wheels. The labels are Font and Cylinder and Rectangle. The text in the image is LOOKING FOR IDEAS";

(h) "Automatically generated caption is a yellow and green balloon. Best guess is spongebob quarantine meme. The labels are Eye and Smile and Cartoon";

(i) "Automatically generated caption is a person standing on a dirt hill. Best guess is soil. The labels are Sky and Asphalt and Grass. The text in the image is GIF";

(j) "Automatically generated caption is a cartoon of a dog. Best guess is cartoon. The labels are Hair and Head and Cartoon. The text in the image is GOOD NIGHT".

D RATING OF SIGHTED USERS

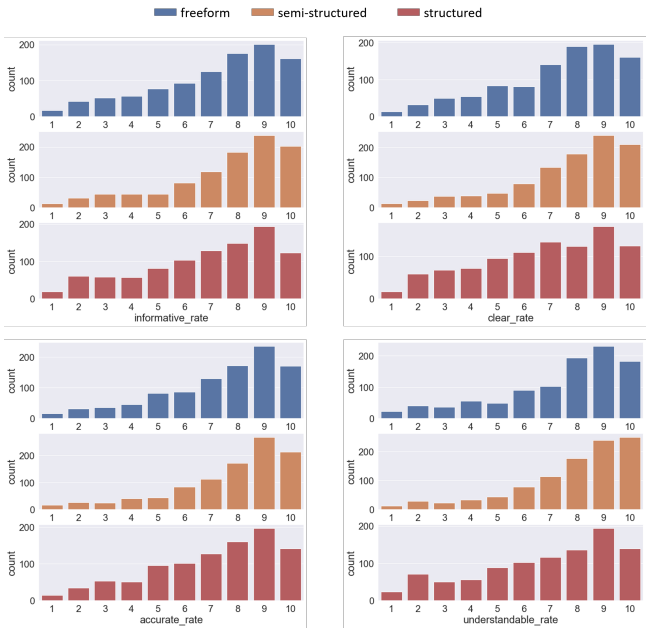

Figure 9: The rating distribution on *informative, clear, accurate* and *understandable*

E RATING OF BLV USERS

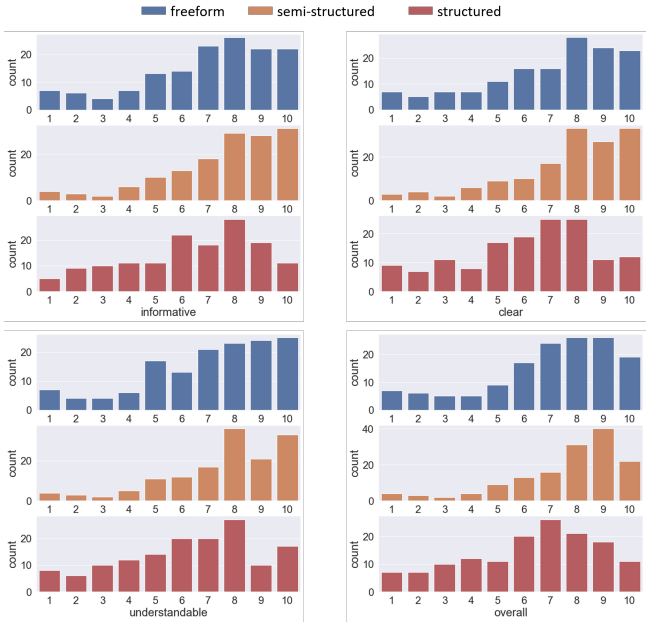

Figure 10: The rating distribution of the three task designs on *informative, clear, understandable* and *overall preference* aspects