ProtoChat: Supporting the Conversation Design Process with Crowd Feedback

YOONSEO CHOI, School of Computing, KAIST
TONI-JAN KEITH MONSERRAT, Institute of Computer Science, University of the Philippines Los Baños
JEONGEON PARK, School of Computing, KAIST
HYUNGYU SHIN, School of Computing, KAIST
NYOUNGWOO LEE, School of Computing, KAIST
JUHO KIM, School of Computing, KAIST

Similar to a design process for designing graphical user interfaces, conversation designers often apply an iterative design process by defining a conversation flow, testing with users, reviewing user data, and improving the design. While it is possible to iterate on conversation design with existing chatbot prototyping tools, there still remain challenges in recruiting participants on-demand and collecting structured feedback on specific conversational components. These limitations hinder designers from running rapid iterations and making informed design decisions. We posit that involving a crowd in the conversation design process can address these challenges, and introduce *ProtoChat*, a crowd-powered chatbot design tool built to support the iterative process of conversation design. ProtoChat makes it easy to recruit crowd workers to test the current conversation within the design tool. ProtoChat's crowd-testing tool allows crowd workers to provide concrete and practical feedback and suggest improvements on specific parts of the conversation. With the data collected from crowd-testing, ProtoChat provides multiple types of visualizations to help designers analyze and revise their design. Through a three-day study with eight designers, we found that ProtoChat enabled an iterative design process for designing a chatbot. Designers improved their design by not only modifying the conversation design itself, but also adjusting the persona and getting UI design implications beyond the conversation design itself. The crowd responses were helpful for designers to explore user needs, contexts, and diverse response formats. With ProtoChat, designers can successfully collect concrete evidence from the crowd and make decisions to iteratively improve their conversation design.

CCS Concepts: • **Human-centered computing → Systems and tools for interaction design**; **Empirical studies in interaction design**; **User interface design**; **Interface design prototyping**.

Additional Key Words and Phrases: Conversation design, Crowdsourcing, Crowd testing, Crowd Feedback, Design process, Conversational User Interface, Chatbot Design

ACM Reference Format:
Yoonseo Choi, Toni-Jan Keith Monserrat, Jeongeon Park, Hyungyu Shin, Nyoungwoo Lee, and Juho Kim. 2020. ProtoChat: Supporting the Conversation Design Process with Crowd Feedback. 1, 1 (September 2020), 27 pages. https://doi.org/10.1145/1122445.1122456

Authors' addresses: Yoonseo Choi, yoonseo.choi@kaist.ac.kr, School of Computing, KAIST; Toni-Jan Keith Monserrat, tpmonserrat@up.edu.ph, Institute of Computer Science, University of the Philippines Los Baños; Jeongeon Park, jeongeon.park@kaist.ac.kr, School of Computing, KAIST; Hyungyu Shin, hyungyu.sh@kaist.ac.kr, School of Computing, KAIST; Nyoungwoo Lee, leenw2@kaist.ac.kr, School of Computing, KAIST; Juho Kim, juhokim@kaist.ac.kr, School of Computing, KAIST.

XXXX-XXXX/2020/9-ART $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Conversation is an essential design component of a chatbot. As a conversational user interface (CUI), in a chatbot the user and the agent interact with a series of chat bubbles in a conversational manner. When designing conversations for chatbots, designers often employ an iterative design process: designing a conversation flow, testing with users, reviewing user data, and improving the design. Many designers use existing chatbot prototyping tools such as Landbot [1], Botmock [2], and Chatfuel [3]. Designers rely on visual aids like flow diagrams offered by the tools to create a conversation scenario and generate a working prototype. Normally, an interactive prototype is exported as a web link, which gets distributed to potential users for their testing and feedback. After the testing, designers analyze the collected data manually and revise their conversation design.

Although iterative conversation design is possible with existing prototyping tools, we observed several challenges around the design process in our formative interview with conversation designers. When designers try to verify their design, it is difficult to recruit participants quickly whenever they want to in order to get feedback on the design. Even though the tools mentioned above support testing chatbots, for testers, there is no way to provide feedback on specific components (e.g., whether the sequence of the conversation is natural, whether a specific utterance is awkward, whether a branch is needed, whether additional topics should be included) due to limited ways to express detailed suggestions. This results in user feedback that is often abstract and not actionable, which in turn presents challenges to designers in making informed design decisions. There have been several methods around collecting granular feedback on design via crowdsourcing in domains such as UI [27] and poster design [23, 32, 35]. However, those approaches mainly support visual design tasks, which might not directly apply to conversational user interfaces. The design of CUI involves 'conversation' that mainly uses free-form responses whereas in GUI, the user interaction is gathered through button clicks, menu selections, etc. As follows, CUI designers cannot easily predict and limit the range of user interactions. Here, we try to collect granular feedback on the unique and specific domain of 'chatbot conversation design', and explore design considerations for getting granular feedback on conversational user interfaces.

In this paper, we explore the idea of engaging an online crowd in the design process to support conversation design. First, we increase the availability of test participants by making it possible for designers to recruit crowd workers on demand within a chatbot design tool. Second, we guide the crowd to provide concrete and clear feedback on specific components during a testing session. Finally, we provide multiple types of interactive visualizations to help designers effectively interpret the collected data and make design revisions.

To investigate the feasibility of the three directions we suggest, we introduce *ProtoChat*, a crowd-powered system built to support the iterative process of conversation design. Designers can create a conversation flow with branching to support conditional flows. After crowd-testing, designers can review and inspect crowd data with interactive visualizations, such as overview of conversation flows and an utterance-level review of crowd conversations. As a tester, a crowd worker can perform three kinds of tasks within the crowd-testing interface—conversing with the chatbot to follow the conversation flow, adding an appropriate utterance on the chatbot's side, and adding a branch in the conversation.

To evaluate how crowd workers and designers use ProtoChat in a conversation design scenario, we conducted a three-day study with eight designers. They went through a design iteration each day and performed four main design tasks (Design, Crowd-test, Review, and Interview) with ProtoChat.

[1] https://landbot.io/
[2] https://botmock.com/
[3] https://chatfuel.com/

Participants chose different domains for their conversation design, which varied from ice cream order to YouTube channel recommendation to talking behind significant other's back. Each day, we recruited a new batch of crowd workers whose number was determined by the designer. We found that ProtoChat could provide an agile design experience to create, test, analyze, and improve the conversation. Designers were able to improve their design with evidence collected from the crowd, by modifying the overall structure of the conversation or fixing a specific part of the conversation. Designers also diversified the options provided to the user, modified the response format (e.g., natural input, button choice) of topics, or gathered insights of UI design implications for the final version of chatbot. Beyond the conversation itself, some designers set a persona (e.g., proactive, good listener) for the chatbot by editing chatbot utterances with crowd input as hints. The conversation design increased in complexity over time through iterations by 33% after the first iteration, and 11% after the second iteration.

The contributions of this work include:

- Insights from the formative interview that identify challenges in conversation design and the required support for a more agile iterative design process;
- *ProtoChat*, an interactive chatbot design tool that supports designers to make informed decisions by collecting design feedback from crowd workers and visualizing the crowdsourced data;
- Empirical findings from a user study that shows how our system could help designers to utilize the crowd feedback and provide the crowd workers the methods to suggest concrete feedback.

2 BACKGROUND AND RELATED WORK

We review previous work related to conversation design and crowdsourcing applications. We first investigate what kind of methods are currently being used for conversation design of chatbots. Then, as we propose a system empowered by the crowd, we discuss how crowdsourcing is utilized in chatbot design and how the crowd is invited to work on usability testing in general.

2.1 Conversation design methods for chatbot

Prior work has been done to investigate possible ways to design conversations that can be used for chatbots. Existing approaches collect conversation data from humans by Wizard-of-Oz prototyping [13, 14, 28] and workshops [15]. The conversation of Dara [28], a chatbot that helped Indian artists to discover international opportunities, was designed with Wizard-of-Oz at the beginning. Ko et al. [13] also utilized the Wizard-of-Oz method to notice the user scenarios of searching business cards, which could result in the multi-dimensional search flow in CardBot. Moreover, Wizard-of-Oz was used to personalize the reflection questions for the agent Robota [14]. Reflection Companion [15] leverages a 12-user workshop to generate the system's mini-dialogue flows. Wizard-of-Oz and workshop methods enable designers to collect quality conversation data in a controlled setting. However, these methods make designers overwhelmed due to time and participant management. Plus, the human-human conversation needs to be verified again to apply in human-agent conversation.

Other approaches formulate the conversation by analyzing existing data sources such as Twitter conversation data [7, 33], mail threads of DBpedia [1], existing chatbot logs [36] and extracted data from apps [21]. Hu et al. [7] and Xu et al. [33] collected and utilized twitter conversation into a training dataset to generate tone-aware and emotional responses. Athreya et al. [1] used the official mailing lists of DBpedia, which includes discussion and conversational threads of mailing lists so that they could be used for creating conversational scenarios. XiaoIce [36], an empathetic

social chatbot used two data sources to generate conversation; one is conversational data from the internet, the other is the previous chat log between XiaoIce herself and her users. Kite [21] automatically created chatbot templates from existing mobile apps which share the logic of user tasks. Although, human conversation data from the internet such as twitter and mailing lists can be easily crawled at scale, it is hard to be directly applied to design human-agent conversation.

Different from the previous work, we aim to quickly and easily collect large amounts of granular feedback on the conversation design by crowdsourcing. The crowd can contribute to improving the conversation design by (1) inserting new chatbot-side utterances that match the current context, and (2) suggesting new branches in a conversation that have not been supported by the designer. Furthermore, the crowd responses could be provided as the data for designing a concrete and high-coverage conversation, which can cover as many as possible scenarios that crowd workers want to proceed around that domain. These interaction data from the crowd gives more concrete insights into how to elaborate the conversation design even before implementing a chatbot, which can foster fast iterations on the conversation design.

2.2 Crowdsourcing in chatbot design

Crowdsourcing has been applied in diverse design domains such as collecting design examples [30], real-time prototyping [17, 19], and getting design critique or feedback [23, 27, 35]. Likewise, crowdsourcing has been utilized in the chatbot domain. There has been work to utilize the crowd to collect and produce dialogue data for the social chat system. Fantom [10, 11] uses a graph-based dialog model for context-maintenance to find suitable responses. The graph gradually evolves with actual chat interactions and system responses by the crowd. InstructableCrowd [8, 9] is an agent that can crowdsource "trigger-action" rules for IF-THEN constructs to automate the management of sensors and tasks.

Other kinds of work leveraged the crowd to respond to the end-user in real-time while maintaining contexts. Chorus [18] is operated with a group of crowd workers who propose responses, vote each other for the best answer, and share collected chat history to maintain the consistency of the conversation. Chorus demonstrated how the crowd could come up with not only a diverse set of responses but also a diverse set of variations of descriptions on a given topic, where they expected crowdsourcing as a potential approach to explore diverse conversations in the chat domain. CI-Bot [22] is a hybrid system that works with crowd experts so that if the user asks an unknown question, it collects the answers from crowd experts and responds. If the answer is satisfying, the answer is appended to the response list of CI-Bot.

Otherwise, crowdsourcing has been used as a method of evaluating the chatbot. ChatEval [29] conducted automatic and human evaluation of chatbots with DBDC (Dialogue Breakdown Detection) tasks and A/B testing with the crowd. Yu et al. [34] suggested a method for evolving existing dialog scenarios by requiring users to evaluate an appropriation, correcting, for answers given by chatbot as they proceed with the conversation. This study showed that the crowd's evaluation is effective in evolving the scenario. They have indeed suggested a method of systematic, accessible chatbot evaluation, but the method is only possible with an already existing chatbot with the complete design of conversation. Choi et al. [3] explored how crowd workers can evaluate a conversation design, and identified designers' needs and expectations in involving the crowd in the design process. We extend this work by introducing a fully functional system that supports designers to quickly test with the crowd workers, collect evidence, and analyze the data and by validating through a user study.

2.3 Crowd-testing user interfaces

Crowdsourcing platforms allow for the quick collection of user feedback at a low cost. Kittur et al. [12] found that, for prototype or user testing, collecting data points from a diverse crowd population was more useful than collection data from a limited pool of experts. Also, Komarov et al. [16] showed that crowdsourcing is a productive way for conducting performance evaluations of user interfaces, there have been studies about leveraging the crowd into testing. Leicht et al. [20] organized the crowd-testing types in four categories: (1) functional and verification, (2) nonfunctional, (3) validation, and (4) usability, all of which are commonly applied to software testing. Muccini [6] shared that the method has benefits such as availability, high coverage, cost-effectiveness, real scenarios, and speediness but also has the challenges of lacking standards, reward mechanisms, and coverages.

To explore and overcome the lacking features of crowd-testing in software engineering, Guaiani et al. [5] explored the way of integrating the crowd-testing into laboratory settings which could potentially complement each other. They collected surveys from crowd testers and found that the difficulties they faced are time pressure or insufficient amount of information, which can be mitigated through better test management. When applying the crowd-testing methods to evaluate web-based interfaces, Nebeling et al. proposed the system CrowdDesign [24] and the toolkit CrowdStudy [25, 26] to invite crowd workers into the process of designing and usability testing the web-based interfaces. For easy integration of crowd-testing, ZIPT [4] was proposed as a way of comparative usability testing at scale without any integration of apps. As a result, designers can easily collect, aggregate and visualize the user's interaction path between third-party apps. Chen et al. [2] introduced two techniques to increase the coverage of crowd-testers. The interactive event-flow graphs collected interactions of every tester and visualized in a single graph and GUI-level guidance could prevent the inactive exploration of paths. As Wang et al. [31] pointed out, crowd-testing often generates a high degree of test case duplication, because crowd workers tend to follow the same paths while testing in parallel.

We aim to apply crowd-testing to support an iterative design of chatbot conversations. As the domain of our interest differs from previous literature such as software engineering [5] or web interface design [2, 24, 26], design considerations for building crowd-testing interfaces, which aim to evaluate the conversation design, need to be discussed. Our system creates a chatbot prototype that embeds the conversation design. The chatbot not only presents the utterances but also asks for feedback and suggestions to improve the design during the conversation session. Crowdsourcing granular feedback and suggestions can give designers more concrete insights into future iterations on the conversation design.

3 FORMATIVE STUDY

To understand how designers iterate on their conversation design ideas and what challenges arise during the design process, we conducted semi-structured interviews with nine designers. With the interview, we were able come up with four design goals for a system that supports a conversation design process.

3.1 Interview

We conducted one-hour long interviews with nine designers, specifically two professional conversation designers with at least one year of experience and seven amateur chatbot designers with prior experiences in conversation design. The interview questions focused on the designer's current conversation design process, the preparation stage, and the design goals of their conversation design.

Furthermore, the designers were asked about the challenges and desired support in the current conversation design process. Here we summarize the main observations from the interviews.

3.1.1 Cannot recruit the desired amount of people at the right moment.

When the designers want to test their conversation design, they usually run a Wizard-of-Oz study with messenger apps or lab study with existing chatbot prototyping tools to collect quality feedback. Because recruiting testers and testing the design takes a sufficient amount of time, it is hard for the designers to run the test whenever needed. The heavy process of repeatedly recruiting testers, making a working prototype, and running the user testing makes it challenging for the designers to increase the number of testers as well.

3.1.2 Hard to collect specific and concrete feedback on the design.

Even if the designers successfully run testing sessions, often the feedback they get is abstract and high-level, such as the overall usability of the chatbot(e.g., "*This chatbot is poorly working.*") or the overall impression(e.g., "*I don't like the movie recommendation*"), and it is difficult to get feedback on each component or specific points of the conversation. In-person sessions such as lab study can partially solve this problem by directly asking the users, but the feedback is not provided during the conversation which requires another interpretation by the designer. Furthermore, since the feedback cannot be scaled up, designers usually embed a survey at the end of the conversation session. A designer shared their experience on providing the testers with the chatbot and a survey at the end, but said that the survey results were not helpful to improve the conversation design of their chatbot.

3.1.3 Hard to analyze the feedback for concrete decision-making.

As the feedback collected through user testing tends to be abstract and often not anchored to specific design components, it is difficult to organize the feedback into action items. The designers also go through a manual process of applying the feedback to the design modification, without a proper support of representation such as visualization of unusual cases. Designers reported they often do not have enough evidence to make concrete design decisions. One expert designer mentioned that it is hard for them to feel confident about their decision even after actual chatbot deployment. They said since the data is analyzed manually, even with the large-scale natural responses that comes in after the deployment, it is still difficult to see the major trend of the users as well as the possible modification that can happen to the chatbot.

3.2 Design goals

Based on the interview results, we hypothesize that leveraging a crowd in the conversation design process can address the challenges identified. We set three design goals for a crowd-powered system that supports conversation design.

G1: Easy access to the crowd for quick and frequent testing. Designers expressed the need to run user testing frequently and whenever desired, but in practice it is difficult. By utilizing the crowd as a tester, designers can obtain the number of testers they need, at any time. We expect a seamless user testing process with integrating the crowd-testing into the design tool itself.

G2: Support the crowd to provide granular feedback. As a tester, there are limited ways of providing feedback to the designers. This results in the testers often leaving high-level feedback or overall impressions on the prototype. It is important to guide the crowd to produce granular feedback on the design, which would be helpful in improving specific parts of the conversation design. By replacing the end-of-conversation survey to a more detailed, frequent suggestion mechanism, testers can support data-supported decision-making for designers.

G3: Provide an efficient visualization for concrete decision-making. As the number of testers increases, manually reviewing and organizing the feedback becomes tedious. To support designers to make informed designs, providing multiple levels of visualization can be effective. Both a collective view of the feedback at the conversation level and a list of individual suggestions at the utterance or branch level can be effective.

4 SYSTEM: PROTOCHAT

ProtoChat supports designers to rapidly iterate on the conversation design by allowing designers to create conversation sequences, quickly test the designed conversation with the crowd, analyze the crowd-tested conversation data, and revise the conversation design. These features are manifested in two main interfaces: the designer interface and the crowd-testing interface.

Our core contribution is in enabling collecting granular crowd feedback (G2). G1 and G3 have been investigated by previous research, and we use these techniques to build our novel system. The three goals combined are incorporated into a single system ProtoChat. To support *G1* (Easy access to the crowd for quick and frequent testing), we provide the Design page (in the designer interface) including the low-fi prototyping tool and feature for simulation and crowd deployment at hand, which is similar to Chorus [18]. To support *G2* (Support the crowd to provide granular feedback), we provide the *Crowd-testing interface* which enables collecting feedback from the crowd by proceeding a conversation. Three possible crowd interactions are presented in Section 4.2.1. To support *G3* (Provide an efficient visualization for concrete decision-making), we provide the *Review page* (in the designer interface), which presents the collective view of crowd conversations (*Topic node graph*) and the micro view of each conversation (*Crowd-based review*). Plus, the frequency of crowd responses below each topic (*Topic-based review*) is provided as well.

4.1 Designer Interface

4.1.1 Design Page.

In the design page (Fig. 1), designers can draft, test, and deploy their conversation. The page consists of the conversation design on the left as a node graph, and the buttons for testing and deploying the conversation design with the crowd-testing interface.

The *design node*, a basic building block of a conversation, represents a chatbot's utterance and constructs a node graph. A design node consists of a 'topic', 'message', and 'sub-message(s)' (Fig. 1-a). The topic in the design node is used to show crowd workers the flow of the conversation and to collectively visualize the crowdsourced conversation. The sub-message feature allows the designer to create several messages coming from the bot before expecting a reply from the user, allowing an alternative for designers when they want several messages uttered by the bot instead of one long message. For example, suppose a designer is working on a *Pizza Order* scenario. One design node could have 'Customizing options' as a topic, *"Do you want to add or remove some topping options from your pizza?"* as a message, *"We provide extra cheese for free."* and *"All the other toppings are charged one dollar each."* as its sub-messages. Designers can create a new design node by clicking on the '+' button on the top-right corner, drag the node any where in the interface, and link the design nodes with each other (Fig. 1-b).

Once a series of design nodes have been linked, we refer this conversation flow as a *node graph* (Fig. 1-c). Designers can create a *branch* in their conversation by connecting two or more nodes to a single node. Branching allows designers to create conditional flows so that the conversation can react differently to different user responses. In the *Pizza Order* scenario, a design node with 'Payment Option [*"How would you like to pay?"*]' can have multiple branches, 'Credit Card', 'Cash', and 'Others', each proceeding to nodes 'Credit Card [*'Please proceed with the following link to pay online."*]', 'Cash [*"Please pay when the pizza arrives."*]', and 'Others [*"Then, how would you like*

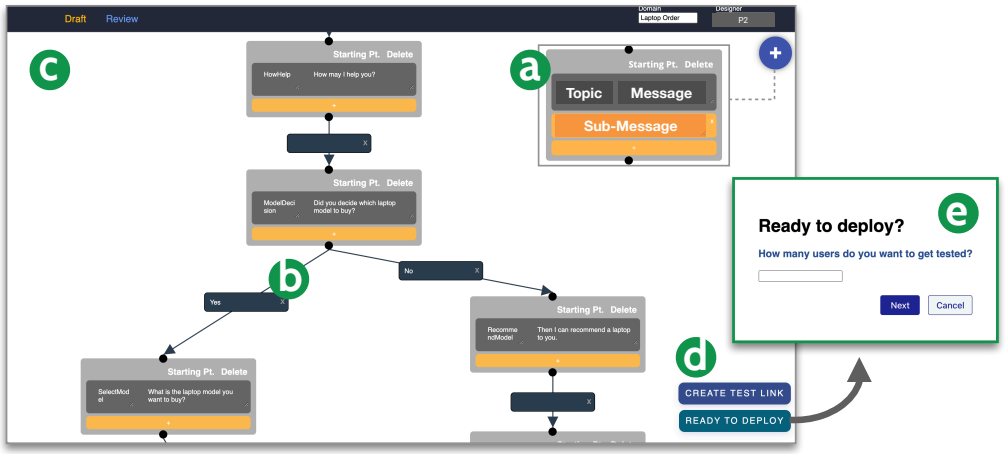

Fig. 1. Designer interface – *Design page* of ProtoChat. On the top right, there is a '+' button for creating a new node (a). The node can be moved and connected to another node with an edge (b). A node graph (c) is constructed with nodes and edges. On the bottom right (d), the 'Create Test Link' button gives a URL to an interactive prototype of the current chatbot, and the 'Ready to Deploy' button allows the designer to test and deploy the current design. Upon clicking on the 'Ready to Deploy' button, a pop-up (e) asks "How many users do you want to get tested?" to determine the number of crowd workers to recruit on MTurk.

to pay?"].' If designers choose to use branches after a specific node, the response format in the crowd-testing interface is given as a button choice instead of a natural response.

Designers can simulate their conversation through the *'Create Test Link' button* (Fig. 1-d) before the actual deployment. Clicking on the button creates a link where the designer can test their own chatbot before deploying to the crowd.

Once the designer finishes the design, they can now deploy it to gather feedback from crowd workers using the *'Ready to Deploy' button*. Clicking the button opens a pop-up window asking the number of crowd workers to recruit. After filling in the number, the pop-up window shows the total amount that will be spent on the testing, and generates a unique URL for the crowd-testing interface (Fig. 1-d, e). Then a HIT on Amazon Mechanical Turk [4] is automatically created with the system-generated, uniform instructions along with the current design. As a default compensation, determined through empirical cases of how long crowd workers normally spend with the pilot study while considering a minimum wage, each crowd worker is paid 1.5 USD for testing a conversation.

4.1.2 *Review Page.*

Once all assigned crowd workers finish testing, the conversation from the crowd is shown on the *Review page* (Fig. 2) to help designers browse and analyze the data. As more crowd workers complete the conversation session, the page gets updated with up-to-date data. This allows the designer to track the crowd's progress through the page. We provide the version record of reviews so that designers can keep track of their design iterations and review history within the review page.

Based on the topics of design node and their links created by designers in the design phase, a *topic node graph* is shown to represent a topical flow and sequences of the chatbot (Fig. 2-a). The topic node graph consists of two elements: a node and a directional edge. When clicking the top-right

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

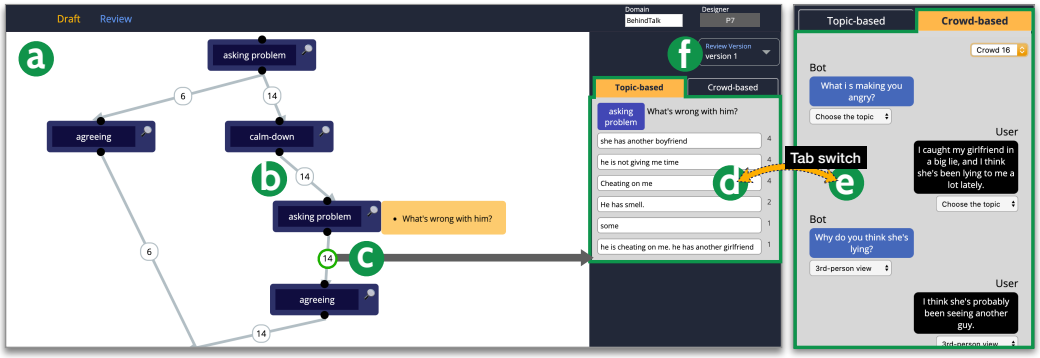

Fig. 2. Designer interface – *Review page* of ProtoChat. On the left side, a topic node graph (a) is displayed, which is composed of chatbot-side topics with messages as nodes (b) and the number of crowd workers that went through a specific flow as edges (c). The right side shows the topic-based review (d) and crowd-based review (e) tabs to see the collected conversation samples in detail. Clicking on (c) displays sorted utterances of the crowd, and different deployment versions can be accessed through the version button (f).

magnifier icon of each topic node, the main message and its sub-messages on the chatbot's side are shown (Fig. 2-b). A directional edge between topic nodes represents user-side responses and shows the number of crowd workers who followed that particular flow in the conversation. This helps designers get a sense of how many users followed the specific path, which is specifically useful when comparing different branch options (Fig. 2-c).

To support the designer's efficient exploration of user responses, we provide a *topic-based review*. If the designer clicks on the number on the edge, the system presents a 'Topic-based' tab on the right panel and shows a response set that comes after the starting topic of the directed edge. The responses are sorted by frequency so that designers can understand what responses are submitted for each topic and which user responses are popular (Fig. 2-d).

In the other tab of the right side panel, we provide a *crowd-based review*. The crowd-based review has two roles: (1) to support a micro-level review of crowd conversations and (2) to provide automatic updates on the topic node graph. Designers can browse through each end-to-end crowd conversation with a dropdown, and analyze each conversation in depth (Fig. 2-e). When the crowd suggests new conversation flows, those pieces of conversation are not yet assigned a topic. The designer can label these conversations by either grouping them with existing topics or creating a new topic label, to complete the topic node graph. When labeling a new topic, the topic node graph automatically updates itself to show the updated version of the conversation sequence based on the crowd's suggestions.

4.2 Crowd-testing Interface

Once the designer deploys a designed conversation, crowd workers can test that conversation through the *Crowd-testing interface* (Fig. 3). Each worker is assigned to one session of conversation through MTurk.

The left section of the crowd-testing interface shows the *Topic sequence graph* of the designer's conversation (Fig. 3-a). The topic sequence graph is provided as a hint for upcoming topics in the overall conversation, which allows the crowd to understand where they are in the full conversation tree as well as explore different paths that are available. The crowd worker's current position is displayed as yellow, and the topics already covered are marked as blue. When there is a branch

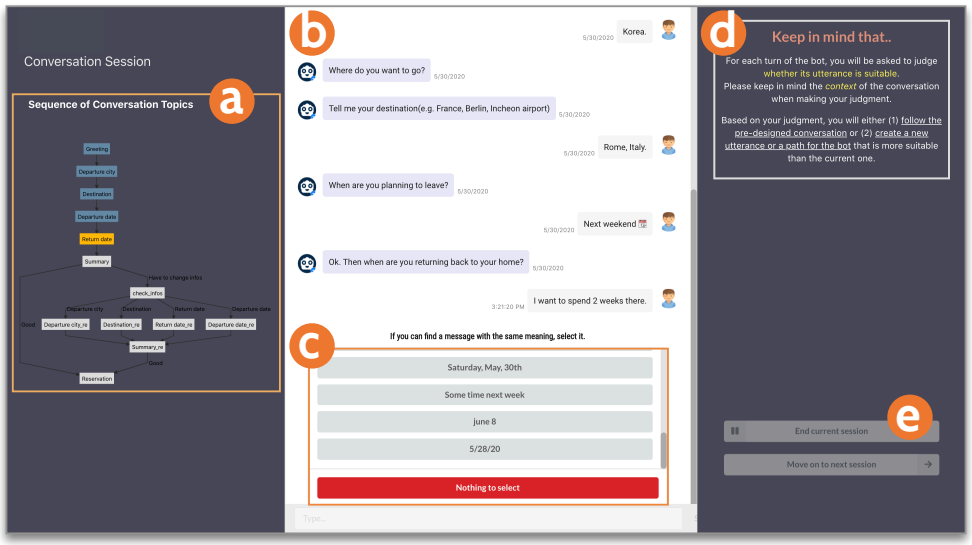

Fig. 3. Crowd-testing interface of ProtoChat. On the left section, a topic sequence graph (a) shows the worker's current position. The middle section (b) is where the chat takes place. Other crowd workers' responses (c) are shown after the worker enters a natural language utterance, for them to identify a similar response. The top-right corner shows instructions (d) for the overall system. After getting to the end of the designed conversation, the crowd worker can end the session by clicking on the "End current session" button (e).

in the conversation, the button choices for selecting each branch are shown as text next to the directional edge.

The middle section is where the chat takes place (Fig. 3-b). It looks like an online chat interface, and the crowd worker's goal is to go through a possible end-to-end path in the conversation of their choice, while checking the utterances and topics they encounter along the way. They enter a user-side response as they converse with the bot. After entering a user-side utterance, a list of other crowd workers' utterances are presented (Fig. 3-c). If the crowd worker thinks their response is similar to one of the existing utterances, they can click on the utterance to merge their response with it. By asking crowd workers to find similarities among other workers' natural language responses, similar responses can be aggregated. This allows designers to easily identify which responses are either common or unique. Every time the crowd finishes responding to the chatbot, the next chatbot-side utterance and the question asking "Do you think the above message(s) suits the current context?" to move onto next topic. The crowd proceeds with the next topic only if they answer "Yes, it's suitable" to the question, and is asked to add a chatbot-side utterance if they answer "No, it's not" (Fig. 5-b).

When the current topic has branches, the crowd worker is first asked to provide a natural language response and then asked if the response matches any of the existing branches with a list of buttons (Fig. 5-c). Detailed explanations about crowd interactions are explained in the next section *Crowd Interactions*.

On the right-top corner, a *short instruction box* is shown (Fig. 3-d). Buttons to end a session are at the bottom-right corner, and when the crowd worker successfully gets to the bottom of the topic sequence graph, the buttons are enabled and the worker can end the session (Fig. 3-e).

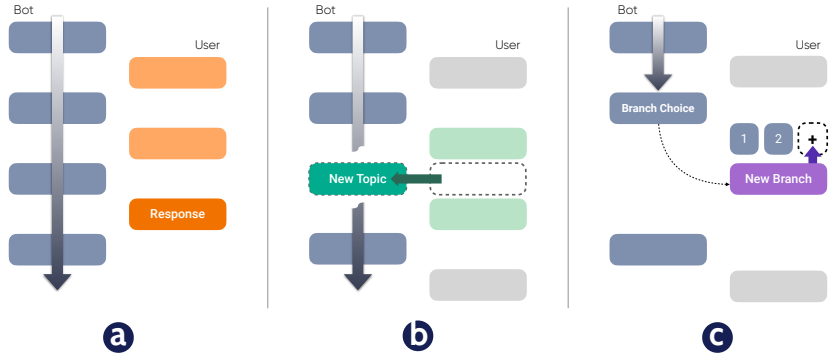

Fig. 4. Three kinds of interactions that a crowd worker can do within the crowd-testing interface. The crowd worker can (a) follow the conversation flow and add a response, (b) add an utterance on the chatbot's side, and (c) add a branch on the user's side.

4.2.1 *Crowd Interactions.*

When using the crowd-testing interface, crowd workers can perform three kinds of interaction in the designed conversation: (a) follow the conversation flow and add a user-side response, (b) add a bot-side utterance, or (c) add a branch on the user's side (shown in Fig. 4). In other words, the crowd worker can both follow and make suggestions on both the user's and the chatbot's sides.

With the sequence of topic-utterance sets that a designer has deployed, the crowd-testing proceeds. To proceed with the conversation, the crowd worker enters a natural language response as a user-side utterance (Fig. 4-a). Next, the crowd worker is asked to choose a similar response among other crowd responses which matches their input utterance (Fig. 5-a).

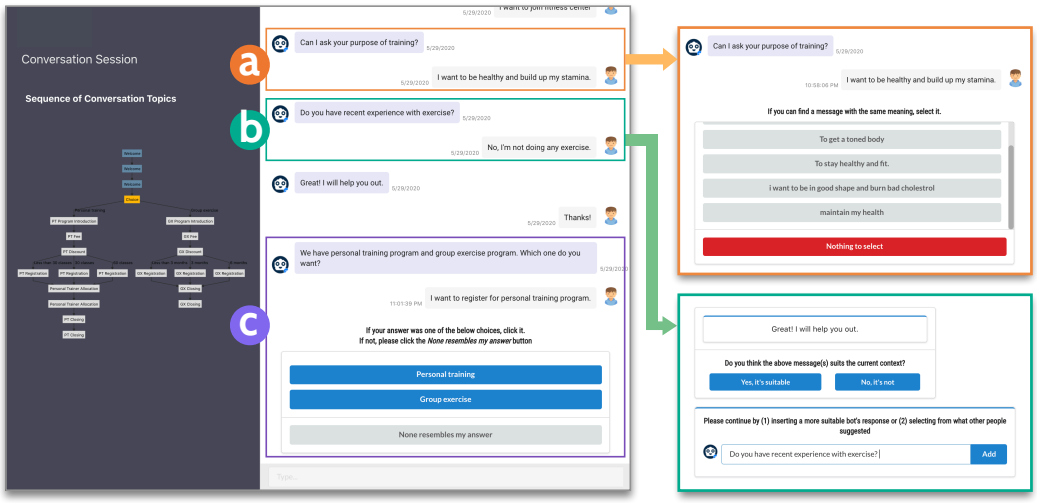

Fig. 5. Interactions that are supported in ProtoChat's crowd-testing interface. Note that (a), (b), and (c) match with (a), (b), and (c) in Fig. 4.

In each turn of the conversation, the crowd workers are asked whether the upcoming chatbot-side utterances match the previous context of the conversation. If the worker answers 'no', they can add a new chatbot-side utterance in-between the conversation (Fig. 4-b). The crowd can either add a new message or choose an existing message from other crowd workers' responses (Fig. 5-b).

At the moment of branching, the crowd worker first needs to enter a user-side natural language response on a topic (Fig. 4-c). Instead of choosing a similar response, they are asked if the input response matches existing branches, which are given as a list of buttons (e.g., 'Personal training', 'Group exercise' in Fig. 5-c). If their input does not match the current branch list, they can add their own response as a new branch by clicking the 'None resembles my answer' button (Fig. 5-c).

4.3 Implementation

The designer interface was implemented with JavaScript and Lit-element. The crowd-testing interface was implemented with React and Dagre library [5] (similar to Chen et al. [2]) for the chat sequence graph for crowd workers. For the database, we used Firebase [6] to connect two separate interfaces.

5 EVALUATION

We ran a study with conversation designers to evaluate ProtoChat. We sought to answer the following research questions: **(1) Can the crowd produce high-quality work with ProtoChat?** and **(2) How does the designer utilize the crowd outcome in their design process with ProtoChat?** To examine the system's role during the overall design process, we conducted a three-day long study.

5.1 Participants

We recruited eight designers (two female, six male) who have prior experience in conversation design and conversational agent development. The participants were recruited through several universities' online board and Facebook group, and had the following eligibility constraint – they have either worked on research topics related to conversation design or had prior experience in chatbot conversation design. The age of the participants ranged from 22 to 34. Participants received 60 USD for their four hours of participation over three days.

5.2 Procedure

The overall procedure of the experiment is shown in Fig. 6. Due to the COVID-19 situation, we conducted the study in a remote setting with Zoom [7]. We showed slides with the system tutorial and the interview questions, and asked the participants to share their screen during the design and review sessions. All sessions were recorded with the participant's consent.

Design phase. On Day 1, participants were asked to create a conversation flow that successfully guides users to finish a task in the domain of their choice. Participants were allowed to do a web search if they needed accurate information about the domain, but searching for and trying any chatbot was prohibited. From Day 2, we asked participants to update their previous version of conversation design after reviewing and analyzing crowd data in the *Review phase*. Participants were asked to think aloud their design decisions and rationale.

Crowd-testing phase. After designing their own conversation flow, we asked participants to decide on the number of crowd workers to test their prototype and ran the test through MTurk. To ensure

[5] https://github.com/dagrejs/dagre
[6] https://firebase.google.com
[7] https://zoom.us

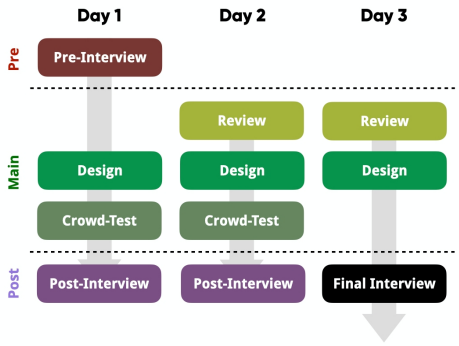

Fig. 6. The procedure of the three-day study. The main part of the study include 'Review', 'Design', and 'Crowd-Test' phase, with no Review phase on Day 1 and no Crowd-test phase on Day 3. Interview sessions were included at the beginning and the end of each day.

the quality responses from crowd workers, we only recruited crowd workers who had a 97 percent of minimum HIT approval rate and more than 50 completed tasks. Plus, we only used the data of crowd workers who completed the crowd-testing session from end-to-end.

During the study, due to a request of the participant to customize the MTurk Survey instructions, we did not use the automatic deployment function and instead manually ran the test with customized domain name and task description. Participants could get the crowd results on the next session of the study.

Review phase. After testing the design with the crowd, participants were asked to browse and analyze the collected data with the crowd-testing interface. Participants used the topic node graph, topic-based review, and crowd-based review to investigate the crowd-tested conversation. During the review phase, participants were asked to think aloud what they have learned and felt.

In addition to the main design activities, the study had multiple interview sessions. The pre-interview (Day 1) mainly focused on understanding their prior experiences, needs, and challenges in the conversation design process. The post-interview (Day 1 & 2) asked about participants' experience and design process with ProtoChat, and the final interview (Day 3) additionally asked about the overall usability and feature suggestions for ProtoChat.

5.3 Measures

To answer the first research question (*"Can the crowd produce high-quality work with our system?"*), we analyzed the crowdsourced conversations by work performance and overall quality of the crowd contributions. The crowd work performance was measured with the length of conversation, the activeness of the crowd workers and the overall quality of crowd conversation. To measure the turn length of conversation, we counted the number of turns in versions deployed to crowd workers and the actual number of turns crowd workers completed. To measure the type and activeness of crowd contributions, we measured the quantity of new utterances added on the chatbot's side, and new branches and responses added on the user's side. Furthermore, we asked participants to rate each crowd worker's conversation quality based on whether it met their expectations (in a 5-point scale).

To answer the second research question (*"How does the designer utilize the crowd outcome into their design with our system?"*), we relied on qualitative responses from participants during the study. We analyzed both what participants said as they were thinking aloud during the design

process, as well as what they said in the pre/post/final interviews. To analyze qualitative data from think-aloud session and interview, we first scripted all the audio recordings into text and two researchers conducted a content analysis on it.

5.4 Result

Participants chose different domains for their conversation design, which varied from ice cream order (P4) to YouTube channel recommendation (P3) and talking behind when having a conflict with a significant other (P7). We describe participants' demographics, background information, and the domain they picked for the study in Fig. 7.

Participant (Gender, Age)	Background information	The domain of conversation design
P1 (M, 29)	UX designer in a company customer service chatbot	Airplane ticket reservation
P2 (M, 22)	Designed conversation and developed a counseling chatbot	Laptop order
P3 (F, 31)	Designing conversation for a chatbot in research	YouTube channel recommendation
P4 (M, 34)	Designing conversation for a chatbot in research	Ice cream order
P5 (F, 29)	Designing UX for VUI and home agents	Jewelry order
P6 (M, 24)	Designing user interaction methods for smart home device automation	Home repairing service
P7 (M, 33)	Designed airport concierge based on humanoid robot	Talking behind significant other's back
P8 (M, 26)	Designed a chatbot for depression therapy chatbot for graduate students	Registration at a fitness center

Fig. 7. Participants' prior experience in conversation design and the domain they chose for the experiment.

Most participants chose domains they are familiar with for their design. P4 (Jewelry order) chose a domain that was the closest, as they were previously a jewelry designer. P7 (Talking behind when having a conflict with a significant other) chose the most interesting domain they could think of, as the domain was more emotional than other task-oriented domains. Nonetheless, the designer had a specific goal in mind, which was to help resolve a couple's current conflict.

5.5 Crowd work performance and designer's perception

Here we summarize the results of crowd-testing as well as whether and how they matched designers' expectations.

5.5.1 The details of crowd-testing for each day.

On average, designers tested their conversation with 28 crowd workers (min: 10, max: 50) on Day 1 and 32 crowd workers (min: 20, max: 50) on Day 2 (Fig. 8). In total, designers recruited 474 crowd workers across 16 batches of crowd-testing (eight designers, two times of deployment). The average crowd-testing completion time was 3 minutes and 50 seconds for Day 1 and was increased to 5 minutes 25 seconds on Day 2. Day 2 completion time was comparatively longer than that of Day 1, which might be natural due to the increased complexity and completeness of the design with the iterative process. It is directly related to increased turn length of the tested designs, which we explain in the next subsection. On average, each crowd took 4 minutes and 44 seconds to complete their given task.

For crowd-testing on Day 1, designers found it difficult to decide on the number, and made the decisions without much evidence. Factors they took into consideration were the purpose of testing and the reviewing workload on the next day. After one iteration, designers were able to adjust the number of crowd workers based on the crowdsourced data. The variance in the number of

Participant	Day 1 Crowd-testing				Day 2 Crowd-testing			
	Crowd	New topics	New branches	Responses	Crowd	New topics	New branches	Responses
P1	50	20	3	101	30	3	1	75
P2	50	33	5	73	30	20	2	95
P3	30	38	2	51	25	11	1	32
P4	30	23	11	39	50	47	11	82
P5	20	5	3	31	35	22	3	40
P6	14	1	1	28	40	6	0	57
P7	10	12	3	28	20	16	1	46
P8	20	4	0	21	20	6	0	57

Fig. 8. The quantitative outcome of crowd-testing with ProtoChat. For each day, the first column shows the number of crowd workers designers chose to recruit. The remaining columns show the number of new topics, the number of new branches, and the unique number of user-side utterances after merging.

crowd was lower on Day 2, as most participants found 30 as a good number in both verifying and exploring the conversation domain as well as analyzing each conversation. Some participants (P1, P2, P3) decreased the number of crowd workers as they aimed to verify the current design and did not want additional expansion of the conversation paths. Others (P4, P5, P6, P7) increased the number for their own reasons. P4 wanted to look for design mistakes and room for improvement with a larger group of users. P6 anticipated to get more diverse responses and inputs from the crowd as they changed parts of the design from buttons to natural responses. P8 decided not to change the number (20 testers) as they observed that most workers followed the deployed design rather than suggesting new paths in Day 1.

5.5.2 The work performance of the crowd.

With the two iterations, designers increased the average length of their conversation design from 6.83 (Day 1), to 9.09 (Day 2), to 10.07 (Day 3) (see Fig. 10). For the first design iteration (Day 1 → Day 2), designers made significant changes in their design with the crowd-tested conversation. Compared to the first iteration, the second iteration (Day 2 → Day 3) was more focused on completing the design as a final product, and the length of the design decreased (P4) or stayed the same (P1) for some designers. To show each designer's design improvement in terms of conversation length, the range and average number of turns are shown in Fig. 9.

Designers were able to iteratively refine and improve the conversation flow with new inputs from the crowd. With the crowd-testing interface, the crowd proceeded with a varying number of turns (see Fig. 10) depending on the length of the specific branch they chose. The length of the conversation does not indicate the quality of work, but rather shows that the crowd performed the tasks with different flows.

There were three types of crowd suggestion allowed in the crowd-testing interface. For addition of new topics, there were total 267 suggestions made by the crowd. In Day 1, P3 (YouTube channel recommendation) got the largest number of crowd suggestions (38 topics) on the chatbot's side. In Day 2, P4 (Ice cream order) received 47 topics. P6 and P8 did not have many new topics introduced by the crowd, and it may be due to the fact that their design was already stable on Day 1. Designers received crowd suggestions for new topics to include in the conversation and applied them to their next designs. Example topic additions from the crowd and designer's corresponding design improvements are shown in Fig. 11. For example, P2 (Laptop order) observed that several crowds

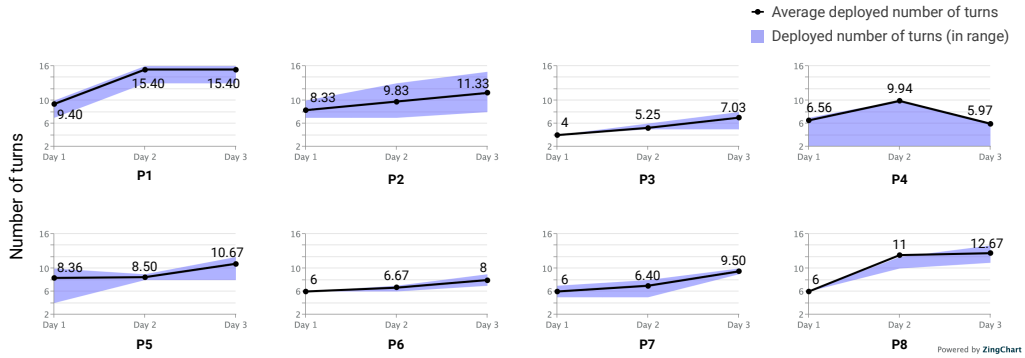

Fig. 9. Visualization of the conversation length of each participant for the three-day study. The purple area shows the range of the deployed number of turns, and the black dot shows the average. The more detailed result on the length of conversation appears in Fig. 10.

Participant	Day 1				Day 2				Day 3	
	Deployed Design		After Crowd-testing		Deployed Design		After Crowd-testing		Design	
	Mean	Range	Mean	Range	Mean	Range	Mean	Range	Mean	Range
P1	9.40	7-10	7.34	6.5-12	15.40	13-16	13.20	8-16	15.40	13-16
P2	8.33	7-10	8.56	4-13	9.83	7-13	11.40	7-18	11.33	8-15
P3	4.00	4	5.18	4-10.5	5.25	5-6	5.52	5-9	7.03	5-8
P4	6.56	2-7	6.88	4-10	9.94	2-10	9.39	2.5-12	5.97	2-6
P5	8.36	4-10	7.05	3-9	8.50	8-9	8.31	6.5-11	10.67	8-12
P6	6.00	6	5.79	3-6	6.67	6-7	6.41	6-8	8.00	7-9
P7	6.00	5-7	7.20	5-16	6.40	5-8	7.83	5-13	9.50	9-10
P8	6.00	6	6.18	6-7	11.00	10-12	11.65	10-13.5	12.67	11-14
Total Mean	6.83		6.77		9.09		9.21		10.07	

Fig. 10. The length of conversation measured with the number of turns. The deployed and the crowdsourced number of turns are shown below the each day column to show how the crowd proceeded and added on the deployed conversation. Both the mean and the range of the turn numbers are shown in the table, and the mean is calculated by averaging all possible end-to-end conversation flows. Because the designers did not run crowd-testing on Day 3, the 'After Crowd-testing' column does not exist.

suggested how long does it take to deliver a laptop. They added the chatbot utterances like "*The cost will be 399.99 and will be shipped in 5 to 7 days.*", "*It will be shipped today and you should have it by Wednesday.*" just before the conversation ends. With these collected data, P2 could add the topic 'Delivery information' with the utterance "*It will be arrive at the destination in 3 days.*"

Compared to topic suggestions, much fewer branch suggestions (47 total) appeared in the crowd-testing sessions, except for P4 with 11 new branches in both iterations. Since P4 designed most of his conversation with branches, there were relatively a large number of new branches with suggested options (e.g., 'Strawberry' in Ice cream flavor, or 'Hazelnut' in Syrup choice) added, but most of them were redundant in the design modification.

For user responses, we counted the unique number of utterances on the user side. As many designers used branches in the conversation design, the number of responses are related to the number of chatbot-side utterances that had natural language responses. There were mainly three patterns of how the participants analyzed the *topic-based review* and applied them to their design – by exploring user needs, user context, and diverse response format. P1 observed different types of response formats about the topic 'Return date'. Due to the responses like "*Sometime in next week*" and "*I will be back by Friday*", they decided to log the local time of the users in the final chatbot so that they support the diverse responses regarding the date. Examples of other patterns observed from the *review page* are shown in Fig. 12.

Participant (Domain)	Bot utterances created by the crowd	How did the designer apply them?	
P2 (Laptop order)	• It will be shipped today and you should have it by Wednesday. • You can book time of delivery for your product. • The cost will be 399.99 and will be shipped in 5 to 7 days. • You will received your product in 3 days time.	***Delivery information***	*It will arrive at the destination in 3 days.*
P3 (YouTube channel recommendation)	• Ok. Here's what I recommend: Cinemassacre. • There are channels for water color painting. • I can recommend food channels like Tasty, Jamie Oliver, Bon Appétit, New York Times Cooking. • Yeah sure. You can watch The Ellen Show, Dude Perfect, Kids Diana Show.	***Art*** ***Entertainment***	*Do you enjoy drawing art? I can recommend how to videos!* *Do you enjoy watching short entertaining clips? Like Ellen show or Dude Perfect?*
P4 (Ice cream order)	• Payment methods? • Yes you can pay through credit card. • How will you pay?	(Sub-message of) ***Syrup choice***	*How do you want to pay? You can pay with a) credit card, b) cash.*
P5 (Jewelry order)	• If you confirm the order and design, we will process your order. • Okay, please wait while you send a confirmed text message with the order details and then your order will be completed.	***Confirm***	*Then you are looking for a fancy/simple jewelry for your {recipient}?*
P7 (Talking behind significant other's back)	• Do you think you can do that now? • Can you talk now? • What can you do about it? • Can you work it out?	***Willingness to solve the problem*** ***Solution question***	*Do you want to move forward? or you want to still talk behind him?* *What do you want him to do?*
P8 (Registration on a fitness center)	• Do you prefer a male or female trainer? Or does it matter? • Can you tell me about your preference?	***Personal trainer allocation***	*Any preference for your personal trainer?* *Please tell me your preference! You can tell me regarding to gender, training experience and etc.*

Fig. 11. Examples of chatbot-side utterance suggestions by the crowd and how the suggestions were applied to the next design iteration.

5.5.3 The overall quality of crowd contributions met the expectation of designers.

In the beginning of the review session on Day 2 and 3, designers were asked about their expectations of crowd-testing outcomes before they looked at the crowd-tested data. Designers had different expectations for the crowd in each iteration, from successfully achieving the goal (P1-Day 1, P4, P7-Day 2), to actively suggesting topics and utterances within a conversation (P2-Day 1, P3-Day 1, P5, P6, P8-Day 1), to experiencing a natural conversation (P7-Day 1), to getting confirmation about the current design from the crowd (Day 2 of P1, P2, P3, P8). The expectations evolved to be more detailed in the second iteration based on their Day 1 testing experience. Designers sometimes edited their conversation flow by removing the branch and making the utterance more open-ended. This enabled designers to explore diverse response sets. During the review session, designers rated each conversation on whether each crowd could proceed and complete the conversation to

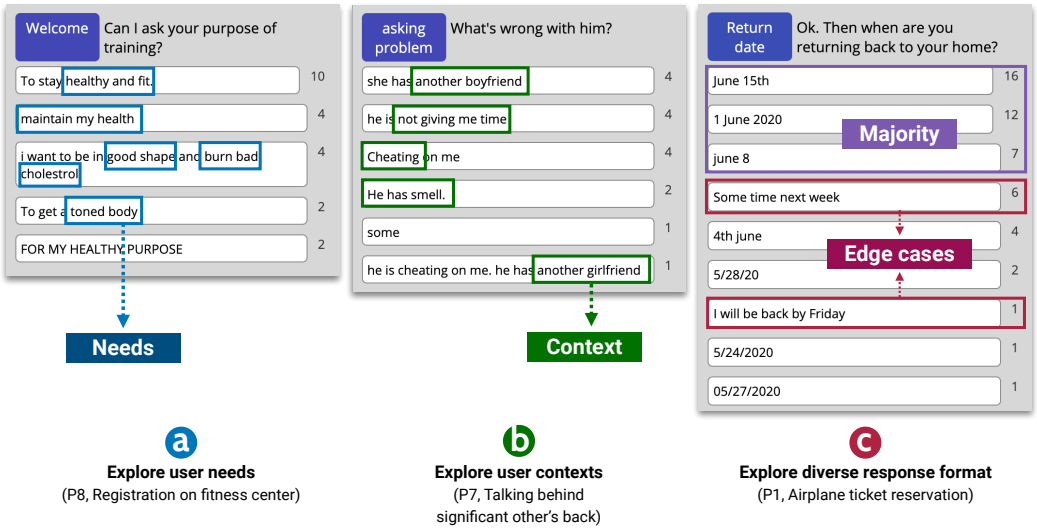

Fig. 12. Examples of responses by the crowd organized by three main patterns.

their expectations in the testing (in a 5-point scale). The average rating was 3.877 across the eight designers.

5.6 How does the designer utilize the crowd outcome into their design?

In this section, we describe how designers used crowd-testing to iteratively develop their conversation design. Designers tested and collected data from the crowd for different purposes, and improved their design from minor details (e.g., the tone of an utterance) to major flow modification (e.g., adding chatbot-side utterances or changing the order of topics). After using ProtoChat for three days, designers commented that the conversation design process was quick and fun, and emphasized that they would use it later in their chatbot design practice.

5.6.1 Overall usage patterns of ProtoChat.

During the design process, designers drafted the conversation from top to bottom. They usually filled out the topic and the message first, and added sub-messages if the message became too long, or if options were provided. Occasionally designers tried to simulate their current design in a generated crowd-testing interface. The 'Create Test Link' button was one of the most beloved features as the designers could experience their conversation in the perspective of the crowd and make detailed revisions.

Based on the review features provided by the system, designers could efficiently analyze the crowdsourced data. At a glance, they checked their deployed design through the topic node graph and the number of crowd workers who followed each path in the conversation by inspecting the edge count between a pair of topics. By looking at the numbers, designers verified the user needs or identified a majority flow of the crowd-tested conversation. The topic-based review helped designers quickly figure out edge cases and the crowd's tendencies in the responses, by providing a quick preview of responses before moving on to reviewing each crowd worker's conversation in depth. Designers spent most of their time on crowd-based reviews, and analyzed each conversation within its specific context. To include the added utterances of the crowd in the topic node graph, designers were asked to add a topic label on each new utterance, that incurs real-time changes of

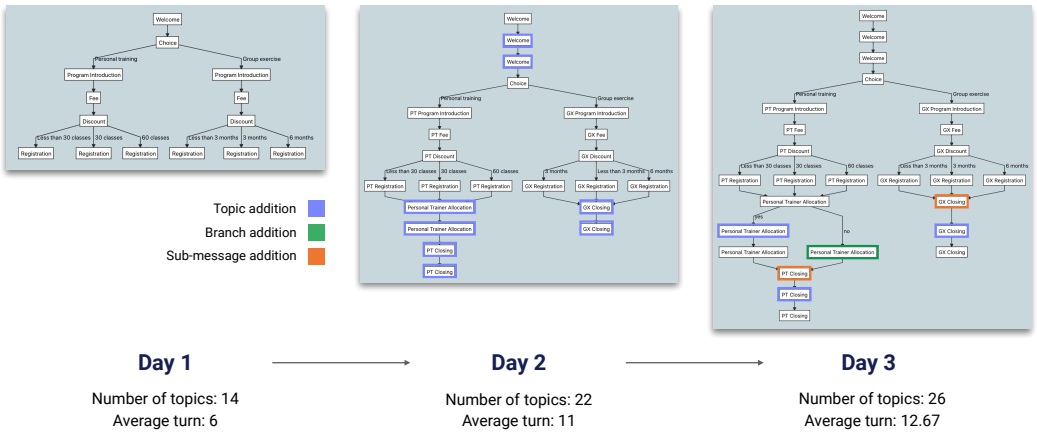

Fig. 13. Example of design iterations (P8). The conversation design become more complex and refined with multiple iterations.

the topic graph. If they did not want to include the added utterances in the topic node graph, they skipped labeling.

5.6.2 Improving designs with collected evidence from the crowd.
Designers revised their design by consulting diverse feedback and suggestions from the crowd. Revisions occurred at both high level by changing the order of the conversation or low-level by adding options or changing the tone of an utterance. We describe the five revision patterns – changing the main flow, diversifying branch options, fixing a specific flow, setting a persona, and deciding & modifying response formats. An example of design iterations (P8) is shown in Fig. 13.

Designers *changed the main flow of conversation.* P6 (House repair) designed a conversation with two major flows: (1) online diagnosis of the problem and (2) reservation for door-to-door repair. They assumed that users who directly inquired about solutions would not make a request to visit their home. However, after the second iteration, the designer found that some crowd conversations made a visit request after choosing the online diagnosis branch. P6 decided to add an extra branch to help the users who wish to make a reservation for door-to-door repair after the crowd responded with 'yes' for the question "*Do you think we need to visit your home?*" Aside from the original intention of the branching feature which was to support distinct paths, we realized that branching could be used to connect separate flows of conversation.

Designers also *diversified the options provided to users.* For instance, P3 (YouTube channel recommendation) initially designed two options (food, others) to support the users who prefer indoor activity and expected that the two options would be enough to cover the crowd preferred activities in the first deployment. With the first round of crowd-testing, the designer was able to explore additional activity preferences. In their next iteration, the designer added categories such as *cooking, art, and entertainment* from crowd suggestions for YouTube channel recommendation. Furthermore, they were able to supplement the list of channel recommendations provided to the user with the crowd's YouTube channel recommendations that came in as the user-side utterances.

Sometimes, designers found the need to *fix a specific flow of conversation.* P8 (Registration at Fitness center) provided two choices to the users, which were (1) personal training and (2) group exercise in their Day 1 design. The flow of conversation for each choice was not much different, as it was designed with the same topics 'Program introduction', 'Fee', 'Discount', and 'Registration'.

After one round of iteration, P8 found that the process of registration should be different between two registration choices as some crowd workers asked about 'Trainer information [*"Details about the personal trainer and his experience"*]' and 'Trainer allocation [*"I need a well experienced trainer with minimum of 5 years of experience."*]' As a result, they complemented the flow of personal training with utterances related to trainer allocation and preference [*"Any preference for your personal trainer?"*].

Beyond the conversation itself, some designers *set a persona of the chatbot.* During the design iteration, P5 (Jewelry order) added the chatbot's persona with crowd-testing. During the conversation, the crowd suggested confirmation of their current status of the order. P5 realized that doing so could give more condifence to potential customers and added the message, *"Then you are looking for a simple/fancy jewelry for your (recipient)?"* which confirms the user's preferred style of jewelry. Plus, a crowd worker suggested a conversation that asked the customer to order jewelry right now or later by adding to cart. After reviewing the crowd-based review, P5 inserted the utterance, *"Then do you need to order now because you need the gift you want to buy right now? Or will I just put it in a cart for later?"* at the last part of their conversation.

Designers also *decided or modified the response format of each topic of conversation* for several reasons. Sometimes, designers did not change the content of the conversation but changed the form of response between natural response and choice options. To support the conditional flow of conversation, designers included many branches within their conversation flow. P6 (House repair) made a branch from the topic 'response type [*"Okay, how would you like to receive a response?"*]' in two categories (SMS, E-mail). The designer expected more choices to be added, but there was no new suggestion as all crowd workers chose these two options. After an iteration, the designer got rid of the branch to get natural responses, which later helped them collect a broader set of responses. Moreover, P1 (Airplane ticket reservation) tried to collect user-side information in natural responses by asking 'departure city', 'destination', 'departure date', 'return date' with open-ended questions. By doing so, they wanted to get a sense of what input format needs to be supported in the final chatbot. They were able to collect a diverse set of responses in different formats to use as evidence for future design choices. One example was the responses from the question under topic 'Return date [*"Ok. Then when are you returning back to your home?"*]', where the crowd answers varied from '30th may 2020', '01 june', 'June 22', to 'The next Saturday'.

5.6.3 Overall design process with ProtoChat was light-weight and fun.

On the third and the final day of the experiment, we asked designers about the overall experience of the design process and our system. They commented that the system was intuitive to use, although there were some learning curves in the beginning. Designers became familiar with the system in three days and eventually sped up the design process (P3, P8). Designers shared their previous experience in using diagram tools (e.g., Draw.io [8], Miro [9]) or builder tools (e.g., Dialogflow, Chatfuel) in the conversation design process, but commented that they could not use them to run multiple iterations of conversation design. They were also satisfied that ProtoChat enables running multiple iterations and spending much less time in completing a chatbot conversation. Furthermore, designers stated that the iteration could happen in a light manner and at a micro-scale. Designers reported a lower psychological barrier in adding or editing the current design. For example, designers edited their conversation based on the unit of design nodes (topic, message, sub-messages), which helped them pay attention to each turn of the conversation. This allowed designers to organize their ideas during the design with the system. Designers also mentioned that not only designers but also developers who are potential collaborators can benefit from using

[8]https://app.diagrams.net
[9]https://miro.com

the system. From the developer's perspective, editing the conversation flow after deployment is inefficient and takes too much effort, and using ProtoChat decreases the burden of repeatedly editing and deploying the conversation flow. P6 mentioned that ProtoChat enables iteration before implementation, so that it can potentially lead to a better collaboration between designers and developers.

6 DISCUSSION

In this section, we first analyze the contribution pattern of the crowd. Then we discuss how to mediate the communication between the designer and the crowd, which is important to collect high-quality feedback within the testing. Furthermore, we discuss the potential direction of improving the crowd feedback mechanisms. Based on the interview, we introduce the potential way of expanding the tool usage. In retrospect, we revisit the identified challenges in conversation design and discuss how ProtoChat was able to address them. At the end, we try to bridge the design considerations for collecting granular crowd feedback to other contexts.

6.1 The patterns of the crowd contribution

During and after the review stage, designers were able to understand the patterns of the crowd input, which was helpful to them. With the interview session, we identified three distinct crowd patterns by analyzing the sessions. We organized crowd patterns with analyzing the type of contribution. The most common pattern was the "follower", which refers to the participants that passively follow the pre-designed conversation flow. Designers could easily observe the "followers" during the crowd-based review. When there was a branching point in the conversation design, designers referred to the number of crowd workers following each branch. Designers were also able to confirm their design when they saw no major addition to the conversation.

The pattern that designers liked the most was the "active suggester", which represents crowd workers who actively suggested chatbot-side utterances and branches. Designers mentioned that this type of crowd workers helped discover chatbot-side utterances they had missed. P2 (Laptop order) was able to add utterances related to shipping after several suggestions from crowd workers and indicated that without shipping, users will not reach the goal of ordering a laptop. However, too detailed suggestions were sometimes ignored by designers as they focused on setting up the overall flow of the conversation. For example, P4 (Ice cream order) got many crowd suggestions on the option choices such as suggesting other ice cream flavors, syrup options, etc. One crowd added the chatbot-side utterance "What size would you like?" but P4 did not apply it to design because their main purpose of the design was to quickly accomplish the goal with chatbot assistance. Even though some designers did not incorporate some of the suggestions into their design, they mentioned that they would still help in latter design stages such as making UI level decision. This type of crowd workers helped direct modification of the conversation design, and designers were able to make confident decisions based on evidences provided within the conversation context.

Designers also gained insights from the "strayer", crowd workers who got lost while following the conversation. Sometimes, designers could notice that some crowd workers could not achieve the end-goal and respond with illogical utterances. By trying to understand why the worker was lost, designers were able to modify their conversation flow. Designers pointed out the needs of the crowd's explanation as to why they were lost, as some designers had no idea why some of them were lost.

Overall, designers thought that the crowd data helped modify their design. If the majority of crowd workers suggest a specific alternative, designers were able to incorporate it with high confidence. As crowd input covered more edge cases, designers were able to revise their designs to accommodate more of these edge cases that are rarely but likely to occur in real user scenarios.

Designers got evidence from the "follower" crowd about their design choices and applied the idea raised by "active suggester"' to enrich the conversation design. P5 quoted that "*having a crowd is like having an extra designer.*"

6.2 Communication between the designer and the crowd

One designer (P3) requested that *"I want to provide a free-form survey about the overall experience with the conversation testing."*, and asked *"Can I leave a note to the crowd so that they could be aware of testing purpose?"*. We did not allow the modification since we thought it could make a major difference in the experiment conditions setting, but designers wanted to at least customize the task name and description along with their design domain when running the crowd-testing. Thus, during the study, we decided not to use the automatic deployment function that instantly deploys the conversation design on MTurk. The researchers manually customized the task name and the explanation before deploying the conversation design. By these requests, we noticed that designers want to communicate or guide the crowd toward a specific direction of testing to receive a more focused result. Designers had a clear need to communicate with the crowd, to convey their desired way of testing.

An interesting future direction would be to allow designers to assign specific task goals to each crowd worker, or more generally, specify different goals in each crowd-testing round (e.g., explore the overall conversation, focus on a particular path, be as creative as possible, etc). An easy way of supporting the communication would be enabling designers to write down instructions or a simple message to the crowd.

6.3 Further exploration for improving feedback mechanisms

With ProtoChat, the crowd responses and suggestions are collected on top of the deployed version of a conversation flow so that those results become feedback provided to designers. Designers interpret the feedback in meaningful ways and apply them in action to improve their designs. Besides, there are potential ways to explore different kinds of feedback such as allowing designers to assign specific tasks to crowd workers for testing purposes at the beginning or adding an overall survey at the end. Collecting specific types of feedback could be possible with the three main patterns (Fig. 12) such as exploring user needs, contexts, and diverse response formats, which we have found from crowd responses. Furthermore, designers could directly ask the crowd workers to provide different levels of feedback from utterance-level comments (e.g., "I think this message is awkward. Let's first ask the customer's preference.") to overall feedback about the conversation (e.g., "We might need a branch for a first comer to introduce our service.").

Future studies could explore and compare various feedback mechanisms for improving conversation designs with the crowdsourced data. Plus, it is possible to combine the explored feedback mechanisms into the overall chatbot design process.

6.4 Expanding the uses of ProtoChat for need-finding and UI decision

With crowd-testing, designers observed user needs based on crowd responses and added utterances on the chatbot's side. Aside from the primary purpose of crowd-testing, which is getting feedback for an existing conversation, it could be used as a method of user research. Since the crowd can be interpreted as potential users in the wild, they could express their needs and expectations toward a chatbot while proceeding the conversation. P3 mentioned that analysis of crowd-testing results was rich enough to construct an affinity diagram, which is a frequently used method to analyze the workshop or interview data. It shows the potential of leveraging the power of the crowd for both need-finding and usability testing.

However, we acknowledge the limited usage of ProtoChat in an extremely early stage where designers struggle to imagine what kind of conversation they need to layout from scratch. ProtoChat assumes that designers already have initial ideas for designing a conversation. Existing qualitative methods like WOz and designer workshops could be more useful to explore and imagine an initial flow of design. ProtoChat can be used alongside those methods to complement the benefits of each method. ProtoChat has a unique position where the designed initial flow needs quick validation before moving on to the implementation stage.

With crowd responses, some designers determined an appropriate format of response to be used in their chatbot. Conversational user interfaces can accept various types of user responses such as natural language, button, gallery, datetime, etc. With crowd-testing, designers were able to determine what kind of response formats should be supported in their final chatbot.

6.5 Existing challenges in conversation design could be solved with ProtoChat

Other than the challenges introduced in the Formative study section, existing challenges remaining in conversation design were 'difficult to discover potential scenarios', 'hard to collect feedback for making design decisions', and 'easily overwhelmed by the interactive process required in conversation design'. During the interview in the study, P6 and P7 pointed out that it is hard to cover diverse use cases likely to appear in real deployments only within the designer group. Similarly, P8 said it is hard to predict user responses as a little change in the scenario has a significant influence on the user's answer. P1 pointed out that the current design process of conversation has several challenges, such as *"Having iteration on early-stage like user research is hard"*, *"Even if they could use the WOz method, it is hard to test conversation itself without a prototype."* P5 pointed out the big difference between the 'design itself' and 'experiencing the conversation within chatbot,' which emphasized the need for testing during the design process.

With ProtoChat, designers could finalize a general scenario followed by the majority of crowd workers and edge cases suggested by crowd workers. Designers even investigated user needs with open-ended questions. Furthermore, as the crowd empowers the iterative process of conversation design, ProtoChat enabled designers to get useful feedback from potential users and run multiple iterations to concretize their design. P6 mentioned that *"For a successful conversation design, conversation flow planning is critical. This system has the potential to allow designers to focus on conversation flow planning and design."*

6.6 Collecting granular crowd feedback beyond conversation design

When we invite the crowd into the design process, the crowd feedback can be collected either explicitly or implicitly. For instance, the designer could directly ask the crowd a couple of questions about the design at the end of testing or they could collect logs from user testing sessions to analyze the users' real usage patterns. ProtoChat is unique in that the crowd workers experience the overall conversation flow by testing and naturally suggesting their responses within the conversation. This organic and implicit way of collecting feedback can provide more realistic data and potentially keep the crowd more engaged. In addition, our system provides designers with the tools to analyze this granular feedback and also identify high-level points. Like our approach, researchers could investigate collecting granular feedback from organic situations—such as discussions between crowd workers around a design—and the design of tools that allow designers to inspect the data with a higher-level lens.

7 LIMITATIONS

The study results with ProtoChat were overall positive by inviting the crowd into the conversation design process, allowing an iterative design of the conversation, and supporting the designers' decision-making in a broad set of CUI application scenarios. Yet, we acknowledge some limitations.

First, as 'conversation designer' is a position that has relatively recently emerged, it was challenging to recruit enough target users of ProtoChat. Thus, we ran the interview and the study with participants who have experiences in designing chatbot scenarios or user interaction and developing conversational agents.

Second, because we ran a lab study, we limited the duration of the experiment to three days, allowing two iterations. All designers mentioned that the three-day period was enough to develop and complete the conversation iteratively, but some designers mentioned that they would need to perform more iterations with broader or more complex domains.

Third, manual merging process in crowd-testing interface did not work as expected. We asked the crowd workers to manually merge their response to other crowd's similar response to display the frequency of the user-side utterances. However, since the crowd had different interpretations on the expression 'similar response', this manual merging process sometimes resulted in uniform utterances. Sometimes, they chose other crowd's similar response even their written expression is far from that response. Using Natural Language Processing (NLP) to classify similar responses and calculate the frequency could be a way to solve the problem.

Lastly, we did not specify the crowd workers to those who are familiar with each conversation domain. During the study, designers sometimes asked if they could narrow down the pool of crowd workers to collect in-depth feedback on their conversation. In future work, we might provide the option to narrow down the background of the testers. Adding filters and qualifications to recruit crowd workers would be helpful.

8 FUTURE WORK

Based on the discussion and limitation, we address potential future directions for this research. In addition, we aspire to expand the flexibility of the system to support novice designers to easily design high-quality conversations.

ProtoChat focuses on expert designers who are already familiar with conversational user interfaces and have prior knowledge in conversation design. They intuitively knew how to use the system to design their conversation towards guiding the users to complete the task such as 'ordering a jewelry', 'reserving a plane ticket'. As chatbots are becoming increasingly widespread, many non-experts are interested in building a chatbot as well. At this point, we could expand our system to proactively guide novice designers to follow the process of an iterative design by providing the structured workflow within our system, while giving more flexibility to the experts.

In the chatbot design process, there's a gap between designers and developers as the different fidelity of conversation is designed and required in different stage. Thus, it is inefficient that developers need to adjust the conversation to apply in implementation of a chatbot. In ProtoChat, the building block for design used in this system has elements which are topic, message and sub-message. Designers mentioned that the elements are similar with the 'intent', 'utterance' and 'follow-up intent' in the machine learning based chatbot builder, thus their design seems to be directly applied in the builder (e.g., Dialogflow[10]). With this advantage, developers do not need to make another version of design adjust the design to apply in the implementation but rather they could take a look on the design before and discuss with designers which helps them to communicate in a better way.

[10]https://dialogflow.cloud.google.com/

9 CONCLUSION

The core value of our system is inviting the crowd into the conversation design process to support iterations on chatbot conversation design. ProtoChat enables rapid testing with the crowd and guiding the crowd workers to provide granular feedback on specific points of conversation. With the crowdsourced data, ProtoChat provides interactive visualizations which help designers to make informed decisions.

10 ACKNOWLEDGMENTS

This work was supported by Samsung Research, Samsung Electronics Co., Ltd. (IO180410-05205-01). We all thank members of KIXLAB (KAIST Interaction Lab) for their support, feedback, and infinite cups of coffee. We thank all of our interviewees and study participants. We thank the reviewers for their detailed and tremendously helpful comments. The first author thanks all of the ProtoChat team members for their endless support and collaboration during her first first-author paper submission. The first author wants to acknowledge three children of project members born during the project period (Doah, Yongha, and Ri).

REFERENCES

[1] Ram G. Athreya, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck. 2018. Enhancing Community Interactions with Data-Driven Chatbots–The DBpedia Chatbot. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 143–146. https://doi.org/10.1145/3184558.3186964

[2] Yan Chen, Maulishree Pandey, Jean Y. Song, Walter S. Lasecki, and Steve Oney. 2020. Improving Crowd-Supported GUI Testing with Structural Guidance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376835

[3] Yoonseo Choi, Hyungyu Shin, Toni-Jan Keith Monserrat, Nyoungwoo Lee, Jeongeon Park, and Juho Kim. 2020. Supporting an Iterative Conversation Design Process. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382951

[4] Biplab Deka, Zifeng Huang, Chad Franzen, Jeffrey Nichols, Yang Li, and Ranjitha Kumar. 2017. ZIPT: Zero-Integration Performance Testing of Mobile App Designs. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 727–736. https://doi.org/10.1145/3126594.3126647

[5] Fabio Guaiani and Henry Muccini. 2015. Crowd and Laboratory Testing Can They Co-Exist? An Exploratory Study. In *Proceedings of the Second International Workshop on CrowdSourcing in Software Engineering* (Florence, Italy) *(CSI-SE '15)*. IEEE Press, 32–37.

[6] Henry Muccini. 2014. Is Crowd Testing (relevant) for Software Engineers? Keynote presentation at AST 2014, the 9th IEEE/ACM International Workshop on Automation of Software Test. http://www.slideshare.net/henry.muccini/is-crowd-testing-relevant-for-software-engineers.

[7] Tianran Hu, Anbang Xu, Zhe Liu, Quanzeng You, Yufan Guo, Vibha Sinha, Jiebo Luo, and Rama Akkiraju. 2018. Touch Your Heart: A Tone-aware Chatbot for Customer Care on Social Media. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 415, 12 pages. https://doi.org/10.1145/3173574.3173989

[8] Ting-Hao Kenneth Huang, Amos Azaria, and Jeffrey P. Bigham. 2016. InstructableCrowd: Creating IF-THEN Rules via Conversations with the Crowd. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 1555–1562. https://doi.org/10.1145/2851581.2892502

[9] Ting-Hao Kenneth Huang, Amos Azaria, Oscar J. Romero, and Jeffrey P. Bigham. 2019. InstructableCrowd: Creating IF-THEN Rules for Smartphones via Conversations with the Crowd. *Human Computation* (2019), 101–131.

[10] Patrik Jonell, Mattias Bystedt, Fethiye Irmak Dogan, Per Fallgren, Jonas Ivarsson, Marketa Slukova, Ulme Wennberg, José Lopes, Johan Boye, and Gabriel Skantze. 2018. Fantom: A crowdsourced social chatbot using an evolving dialog graph. *Proc. Alexa Prize* (2018).

[11] Patrik Jonell, Per Fallgren, Fethiye Irmak Doğan, José Lopes, Ulme Wennberg, and Gabriel Skantze. 2019. Crowdsourcing a Self-Evolving Dialog Graph. In *Proceedings of the 1st International Conference on Conversational User Interfaces* (Dublin, Ireland) *(CUI '19)*. Association for Computing Machinery, New York, NY, USA, Article 14, 8 pages. https://doi.org/10.1145/3342775.3342790

[12] Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 453–456. https://doi.org/10.1145/1357054.1357127

[13] Meng-Chieh Ko and Zih-Hong Lin. 2018. CardBot: A Chatbot for Business Card Management. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion* (Tokyo, Japan) *(IUI '18 Companion)*. ACM, New York, NY, USA, Article 5, 2 pages. https://doi.org/10.1145/3180308.3180313

[14] Rafal Kocielnik, Daniel Avrahami, Jennifer Marlow, Di Lu, and Gary Hsieh. 2018. Designing for Workplace Reflection: A Chat and Voice-Based Conversational Agent. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 881–894. https://doi.org/10.1145/3196709.3196784

[15] Rafal Kocielnik, Lillian Xiao, Daniel Avrahami, and Gary Hsieh. 2018. Reflection Companion: A Conversational System for Engaging Users in Reflection on Physical Activity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 70 (July 2018), 26 pages. https://doi.org/10.1145/3214273

[16] Steven Komarov, Katharina Reinecke, and Krzysztof Z. Gajos. 2013. Crowdsourcing Performance Evaluations of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 207–216. https://doi.org/10.1145/2470654.2470684

[17] Walter S. Lasecki, Juho Kim, Nick Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. 2015. Apparition: Crowdsourced User Interfaces That Come to Life as You Sketch Them. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1925–1934. https://doi.org/10.1145/2702123.2702565

[18] Walter S. Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F. Allen, and Jeffrey P. Bigham. 2013. Chorus: A Crowd-Powered Conversational Assistant. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 151–162. https://doi.org/10.1145/2501988.2502057

[19] Sang Won Lee, Rebecca Krosnick, Sun Young Park, Brandon Keelean, Sach Vaidya, Stephanie D. O'Keefe, and Walter S. Lasecki. 2018. Exploring Real-Time Collaboration in Crowd-Powered Systems Through a UI Design Tool. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 104 (Nov. 2018), 23 pages. https://doi.org/10.1145/3274373

[20] Niklas Leicht, Ivo Blohm, and Jan Marco Leimeister. 2017. Leveraging the Power of the Crowd for Software Testing. *IEEE Softw.* 34, 2 (March 2017), 62–69. https://doi.org/10.1109/MS.2017.37

[21] Toby Li and Oriana Riva. 2018. Kite: Building Conversational Bots from Mobile Apps. 96–109. https://doi.org/10.1145/3210240.3210339

[22] Xulei Liang, Rong Ding, Mengxiang Lin, Lei Li, Xingchi Li, and Song Lu. 2017. CI-Bot: A Hybrid Chatbot Enhanced by Crowdsourcing. In *Web and Big Data*, Shaoxu Song, Matthias Renz, and Yang-Sae Moon (Eds.). Springer International Publishing, Cham, 195–203.

[23] Kurt Luther, Jari-Lee Tolentino, Wei Wu, Amy Pavel, Brian P. Bailey, Maneesh Agrawala, Björn Hartmann, and Steven P. Dow. 2015. Structuring, Aggregating, and Evaluating Crowdsourced Design Critique. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work amp; Social Computing* (Vancouver, BC, Canada) *(CSCW '15)*. Association for Computing Machinery, New York, NY, USA, 473–485. https://doi.org/10.1145/2675133.2675283

[24] Michael Nebeling, Stefania Leone, and Moira C Norrie. 2012. Crowdsourced web engineering and design. In *International Conference on Web Engineering*. Springer, 31–45.

[25] Michael Nebeling, Maximilian Speicher, Michael Grossniklaus, and Moira C Norrie. 2012. Crowdsourced web site evaluation with crowdstudy. In *International Conference on Web Engineering*. Springer, 494–497.

[26] Michael Nebeling, Maximilian Speicher, and Moira C. Norrie. 2013. CrowdStudy: General Toolkit for Crowdsourced Evaluation of Web Interfaces. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (London, United Kingdom) *(EICS '13)*. Association for Computing Machinery, New York, NY, USA, 255–264. https://doi.org/10.1145/2494603.2480303

[27] Jonas Oppenlaender, Thanassis Tiropanis, and Simo Hosio. 2020. CrowdUI: Supporting Web Design with the Crowd. *Proc. ACM Hum.-Comput. Interact.* 4, EICS, Article 76 (June 2020), 28 pages. https://doi.org/10.1145/3394978

[28] Archana Prasad, Sean Blagsvedt, Tej Pochiraju, and Indrani Medhi Thies. 2019. Dara: A Chatbot to Help Indian Artists and Designers Discover International Opportunities. In *Proceedings of the 2019 on Creativity and Cognition* (San Diego, CA, USA) *(C&C '19)*. ACM, New York, NY, USA, 626–632. https://doi.org/10.1145/3325480.3326577

[29] Jo ao Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. 2019. ChatEval: A Tool for Chatbot Evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics (Demonstrations) (Minneapolis, Minnesota). Association for Computational Linguistics, 60–65. http://aclweb.org/anthology/N19-4011

[30] Nikita Spirin, Motahhare Eslami, Jie Ding, Pooja Jain, Brian Bailey, and Karrie Karahalios. 2014. Searching for Design Examples with Crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web* (Seoul, Korea) *(WWW '14 Companion)*. Association for Computing Machinery, New York, NY, USA, 381–382. https://doi.org/10.1145/2567948.2577371

[31] Junjie Wang, Mingyang Li, Song Wang, Tim Menzies, and Qing Wang. 2019. Images don't lie: Duplicate crowdtesting reports detection with screenshot information. *Information and Software Technology* 110 (Jun 2019), 139–155. https://doi.org/10.1016/j.infsof.2019.03.003

[32] Anbang Xu, Shih-Wen Huang, and Brian Bailey. 2014. Voyant: Generating Structured Feedback on Visual Designs Using a Crowd of Non-Experts. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work amp; Social Computing* (Baltimore, Maryland, USA) *(CSCW '14)*. Association for Computing Machinery, New York, NY, USA, 1433–1444. https://doi.org/10.1145/2531602.2531604

[33] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A New Chatbot for Customer Service on Social Media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 3506–3510. https://doi.org/10.1145/3025453.3025496

[34] Zhou Hao Yu, Ziyu Xu, Alan W. Black, and Alexander I. Rudnicky. 2016. Chatbot Evaluation and Database Expansion via Crowdsourcing.

[35] Alvin Yuan, Kurt Luther, Markus Krause, Sophie Isabel Vennix, Steven P Dow, and Bjorn Hartmann. 2016. Almost an Expert: The Effects of Rubrics and Expertise on Perceived Value of Crowdsourced Design Critiques. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work amp; Social Computing* (San Francisco, California, USA) *(CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 1005–1017. https://doi.org/10.1145/2818048.2819953

[36] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2018. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *CoRR* abs/1812.08989 (2018). arXiv:1812.08989 http://arxiv.org/abs/1812.08989