ProtoChat: Supporting the Conversation Design Process with Crowd Feedback

Yoonseo Choi
yoonseo.choi@kaist.ac.kr
School of Computing
KAIST
Daejeon, South Korea

Jeongeon Park
jeongeon.park@kaist.ac.kr
School of Computing
KAIST
Daejeon, South Korea

Nyoungwoo Lee
leenw2@kaist.ac.kr
School of Computing
KAIST
Daejeon, South Korea

Toni-Jan Keith Monserrat
tpmonserrat@up.edu.ph
Institute of Computer Science
University of the Philippines Los Baños
Los Baños, Laguna, Philippines

Hyungyu Shin
hyungyu.sh@kaist.ac.kr
School of Computing
KAIST
Daejeon, South Korea

Juho Kim
juhokim@kaist.ac.kr
School of Computing
KAIST
Daejeon, South Korea

CSCW'20 Companion, October 17–21, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8059-1/20/10.
https://doi.org/10.1145/3406865.3418568

ABSTRACT

Conversation designers use iterative design to create, test, and improve conversation flows. While it is possible to iterate conversation design with existing chatbot prototyping tools, challenges remain such as recruiting participants and collecting structured feedback on specific conversational components, hindering rapid iterations, and making informed design decisions. To address these limitations, we introduce *ProtoChat*, a crowd-powered design tool built to support the iterative process of conversation design for a chatbot. ProtoChat enables rapid testing with the crowd and guiding the crowd workers to provide granular feedback on specific points of conversation.

CCS CONCEPTS

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

KEYWORDS

Conversation design, Crowdsourcing, Iterative Design, 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. In *Companion Publication of the 2020 Conference on Computer Supported Cooperative Work and Social Computing (CSCW'20 Companion), October 17–21, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3406865.3418568

BACKGROUND

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 chatbot scenario. 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 inviting the 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. We provide an overview of the CONVIT system that is built upon our previous work [1].

SYSTEM

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

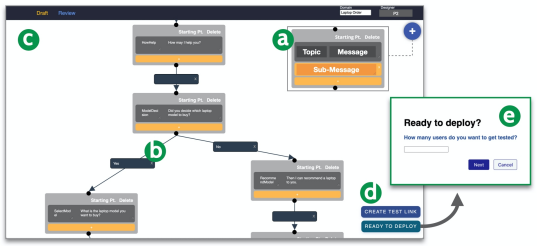

Figure 1: Designer interface – Design page of ProtoChat.

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

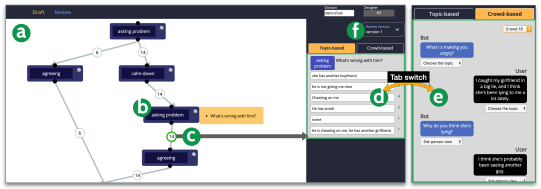

Figure 2: Designer interface – Review page of ProtoChat.

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.

Designer Interface

Design Page.
In the design page (Fig. 1), designers can draft, test, and deploy their conversation. The *design node*, a basic building block of a conversation, consists of a 'topic', 'message', and 'sub-message(s)' (Fig. 1-a). 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. 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* 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. 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 before the actual deployment (Fig. 1-d) with 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 [1] is automatically created with the system-generated, uniform instructions along with the current design.

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 (Fig. 2-b) and a directional edge . A directional edge

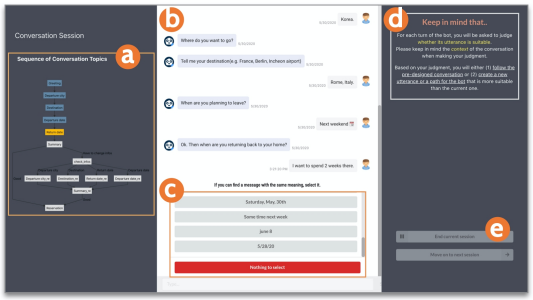

Figure 3: Crowd-testing interface of ProtoChat

between topic nodes represents user-side responses and shows the number of crowd workers who followed that particular flow in the conversation (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.

Crowd-testing Interface

When the designer deploys a designed conversation, crowd workers can test that conversation through the *Crowd-testing interface* (Fig. 3) through MTurk. The left section of the crowd-testing interface shows the *Topic sequence graph* of the designer's conversation (Fig. 3-a). This allows crowd workers to check where in the conversation they are at the moment topic-wise. The crowd worker's current position is displayed as yellow, and the topics already covered are marked as blue. When there is a branch 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. 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?". 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. 4-b).

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. When the current topic has branches, a list of buttons are presented instead for the worker to choose a path.

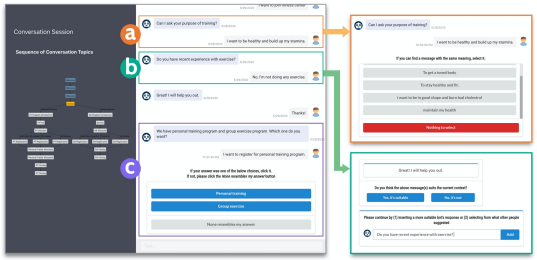

Figure 4: Interactions that are supported in ProtoChat's crowd-testing interface.

Other than entering a user-side response (Fig. 4-a), there are two more interactions: adding a bot-side utterance (Fig. 4-b) and adding a branch on the user's side (Fig. 4-c). In other words, the crowd worker can both follow and make suggestions on both the user's and the chatbot's sides. For (b), the crowd can either add a new message or choose from existing messages from other crowd workers' responses. For the branch situation, the crowd worker can choose from the button options to follow an existing path or add a new branch.

CONTRIBUTION

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. The crowd feedback was helpful for designers to explore user needs, contexts, and diverse response formats. Overall, ProtoChat supports rapid prototyping and evaluation with the crowd and guiding the crowd workers to provide granular feedback on specific points of conversation. With ProtoChat, designers can successfully collect concrete evidence from the crowd and make informed decisions to iteratively improve their conversation design.

ACKNOWLEDGMENTS

This work was supported by Samsung Research, Samsung Electronics Co.,Ltd.(IO180410-05205-01).

REFERENCES

[1] 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