Designing Effective Interview Chatbots: Automatic Chatbot Profiling and Design Suggestion Generation for Chatbot Debugging

Xu Han
University of Colorado Boulder
Boulder, CO, USA
xuha2442@colorado.edu

Michelle Zhou
Juji, Inc.
San Jose, CA, USA
mzhou@acm.org

Matthew J. Turner
University of Colorado Boulder
Boulder, CO, USA
matthew.turner@colorado.edu

Tom Yeh
University of Colorado Boulder
Boulder, CO, USA
tom.yeh@colorado.edu

ABSTRACT
Recent studies show the effectiveness of interview chatbots for information elicitation. However, designing an effective interview chatbot is non-trivial. Few tools exist to help designers design, evaluate, and improve an interview chatbot iteratively. Based on a formative study and literature reviews, we propose a computational framework for quantifying the performance of interview chatbots. Incorporating the framework, we have developed iChatProfile, an assistive chatbot design tool that can automatically generate a profile of an interview chatbot with quantified performance metrics and offer design suggestions for improving the chatbot based on such metrics. To validate the effectiveness of iChatProfile, we designed and conducted a between-subject study that compared the performance of 10 interview chatbots designed with or without using iChatProfile. Based on the live chats between the 10 chatbots and 1349 users, our results show that iChatProfile helped the designers build significantly more effective interview chatbots, improving both interview quality and user experience.

CCS CONCEPTS
• Human-centered computing → Human computer interaction; • Computing Methodologies → Intelligent agents.
KEYWORDS
Conversational AI Agents; Interview Chatbot; Chatbot Debugging; Chatbot Evaluation Framework; Chatbot Design Suggestion; Automatic Chatbot Profiling; Automatic Chatbot Evaluation

ACM Reference Format:
Xu Han, Michelle Zhou, Matthew J. Turner, and Tom Yeh. 2021. Designing Effective Interview Chatbots: Automatic Chatbot Profiling and Design Suggestion Generation for Chatbot Debugging. In CHI Conference on Human Factors in Computing Systems (CHI ’21), May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3411764.3445569

1 INTRODUCTION
During the past few years, chatbots have been used to conduct interviews by engaging users in one-on-one text-based conversations [67]. Recent studies show that interview chatbots are more effective at engaging users and eliciting quality information from the users, compared to traditional online surveys [31, 66, 67].

Despite their promises, it is challenging and time-consuming to build effective interview chatbots due to the limitations in today’s technologies and the complexity involved in interview conversations [18, 65]. Like building any complex interactive systems [42], one potential approach is to design and improve an interview chatbot iteratively. Specifically, the iterative design of an interview chatbot is to fulfill two main goals. First, designing any user interviews or surveys [7, 42], designers of an interview chatbot need to ensure the effective design of an interview task (e.g., proper and clear wording of questions). Second, like building any conversational agents [29], designers of an interview chatbot need to make sure that the chatbot can successfully carry out such an interview task [65].

To achieve above goals, designers often conduct pilot studies prior to a formal study [49]. However, interview chatbot designers face two challenges in detecting let alone fixing the potential issues revealed by the pilot studies. First, designers must examine chat transcripts to discover potential issues and it is laborious and time-consuming to do so manually. For example, Fig 2(a) shows that an interview question poses a challenge for a user due to a lack of clarity, while Fig 2(b) shows a chatbot-recognized user input during an interview, which could result in poor user experience or even abandoned interviews. To detect such issues in practice, chatbot designers must examine chat transcripts (Fig 2(c)) to discover them. It is laborious and time-consuming to do so manually especially if the designers need to detect such issues from a large number of chat transcripts. Second, even if the designers have discovered such issues from reading chat transcripts, they might not know how to fix the issues and improve the chatbot due to a lack of relevant experience (e.g., designing effective interview interactions).

To address the above two challenges, we have been developing a tool, called iChatProfile, which can aid chatbot designers in building, evaluating, and improving interview chatbots iteratively. In particular, iChatProfile automatically analyzes chat transcripts and computes a set of chatbot performance metrics to present designers with a chatbot profile (Fig 1(a)). Based on the chatbot profile, iChatProfile also automatically generates a set of design suggestions to guide designers to improve the chatbot (Fig 1(b)).

In this paper, we present the key steps taken to build iChatProfile. First, we present a formative study that was conducted to understand the difficulties that designers face when building an interview chatbot and identify their desired design assistance. Second, we describe a computational framework that quantitatively measures the effectiveness of an interview chatbot from multiple dimensions, including elicitation ability, user experience, and ethics. Third, we present iChatProfile that was built based on our formative study and the computational framework. To validate the effectiveness of iChatProfile, we designed and conducted a between-subject user study that compared the performance of chatbots designed with or without using iChatProfile. A total of 10 chatbots were created and evaluated live by 1349 participants from Amazon Mechanical Turk. We compared the performance of these chatbots. The results show that the chatbots designed with the help of iChatProfile performed significantly better along many dimensions, including improved user response quality and user experience.

To the best of our knowledge, our work is the first on building an assistive design tool for creating interview chatbots. As a result, our work reported here provides three unique contributions:

(1) A computational framework for quantifying the effectiveness of interview chatbots. This framework comprehensively evaluates the effectiveness of an interview chatbot by computing an extensive set of performance metrics covering multiple dimensions: elicitation ability, user experience, and ethics. Other chatbot researchers and practitioners can easily adopt or extend this framework to build their own chatbot evaluation tools.

(2) Practical approaches to assisting iterative design of interview chatbots. iChatProfile presents practical implementations of an assistive chatbot design tool. Because we have demonstrated the effectiveness of our implementations, others could replicate or extend our approaches to create more tools aiding chatbot design.

(3) Design implications for building assistive chatbot design tools beyond building interview chatbots. Although our current work focuses on aiding the design of interview chatbots, it presents design considerations for assisting the design of other types of chatbots, such as counseling or training chatbots, which share similar design requirements (e.g., ethical considerations).

2 RELATED WORK

2.1 Chatbots for Information Elicitation
AI-powered conversational user interfaces, also known as AI chatbots or chatbots for short, allow users to communicate with computers in natural language, providing more flexible [5] and personalized user experience [69]. Such benefits have encouraged the creation of a wide array of chatbot applications, such as virtual assistants [38], social companions [56], and interview chatbots [37]. Our work is most relevant to the use of chatbots for information elicitation [31, 66, 67].

Researchers have developed various chatbots to elicit information from users through text-based conversations. For example, Bohus and Rudnicky introduce dialog systems that gather required
Designing Effective Interview Chatbots: Automatic Chatbot Profiling and Design Suggestion Generation for Chatbot Debugging

CHI ’21, May 8–13, 2021, Yokohama, Japan

Figure 2: Examples of poorly handled conversations by an interview chatbot. (a) vague question and (b) unhandled user input. (c) chat transcripts set in practice with poorly handled conversations

information for performing specific tasks (e.g., making travel reservations) [4]. More recently, a number of interview chatbots have been developed to elicit information from a target audience. For example, a chatbot is built to interview students for effective teaming [66] and another chatbot to interview gamers for eliciting their game opinions [67]. Williams et al. have developed a chatbot to interview employees for workplace productivity [64]. Compared to traditional, static online surveys, these interview chatbots enhance information elicitation [31, 66] by providing interactive feedback [8] and asking follow-up questions [46].

Our work is directly related to the efforts of creating interview chatbots. However, existing work focuses on developing interview chatbots for specific information elicitation tasks (e.g., [64, 66, 67]) or powering interview chatbots with specific skills (e.g., giving them a personality [71] and active listening skills [65]). While we learn from these efforts, our work reported here has a very different focus: we want to build a tool that can automatically evaluate the performance of an interview chatbot and provide design suggestions for improving the chatbot.

2.2 Chatbot Platforms

There are a number of chatbot platforms and these platforms can be broadly divided into three categories. First, chatbot platforms like Chatfuel [60] and ManyChat [2] allow non-IT professionals to build a chatbot without coding. Since these platforms provide limited AI/NLP capabilities, it would be difficult to create interview chatbots that can understand users especially when open-ended interview questions are involved. The second type includes platforms like Google Dialogflow [13] and IBM Watson [25]. These platforms provide designers with more flexibility to customize a chatbot’s AI/NLP capabilities but designers must have basic AI/NLP knowledge to use the tools. The third category includes platforms like Juji, which provides a rich set of pre-built AI capabilities to enable non-IT designers to build chatbots without any expertise of AI/NLP [59]. While chatbot designers can choose to use any of the chatbot platforms, none of the platforms provides a tool like ours reported here: a tool that helps designers evaluate a chatbot’s performance and provides design suggestions to improve the chatbot.

2.3 Evaluating Conversational AI Systems

Researchers have developed a number of approaches to evaluating conversational AI systems. These approaches can be roughly organized into two categories: measuring objective system performance (e.g., task completion rate for task-oriented chatbots [29]) and assessing subjective human experience (e.g., measuring users’ trust in a chatbot [71]). Incorporating multiple metrics, evaluation frameworks have also been proposed to systematically measure the performance of conversational AI systems. For example, PARADISE has been used to evaluate the performance of task-oriented, spoken dialog systems [62], typically developed by AI/NLP experts. Unlike these works that evaluate conversational AI systems in general, our work presented here focuses on evaluating the performance of interview chatbots, typically designed by non-IT professionals. While we borrow some of the existing objective and subjective metrics, we have developed a computational framework specifically for quantifying the performance of interview chatbots with actionable insights—design suggestions that can help designers improve an interview chatbot.

2.4 Design Suggestion Generation for Effective Interaction

Our work on generating design suggestions is also related to various efforts on guiding the design of human-computer interfaces, such as chatbot systems [21] and graphical user interfaces (GUI) [36, 68]. For example, Han et al. combine domain-specific knowledge together with observational studies to generate rule-based design suggestions for task-oriented chatbots [21]. One of the drawbacks of this approach lies in its inflexibility of adapting design suggestions to changing design goals or dynamic design issues occurring in real time. On the other hand, Lee et al. use autoencoder and k-nearest neighbor algorithms to recommend GUI design examples that help designers in real time [36]. Moreover, Xu et al. have developed a system that incorporates crowdsourcing to generate design suggestions for GUI designers [68]. While we learn from these approaches, we are unaware of any approach to automatic generation of design suggestions based on computed chatbot performance as our approach does.
3 STUDY PLATFORM - JUJI

As mentioned in Section 2.2, there are three types of chatbot platforms. Although we could build iChatProfile on top of any chatbot platform, we decided to build it on Juji for three main reasons.

3.1 Supporting Interview Chatbots

First, recent studies show that other researchers have used Juji to build various interview chatbots, which matches our focus on aiding the design of effective interview chatbots [37, 61, 65, 66]. Building and deploying an interview chatbot on Juji is very similar to creating a survey on a popular survey platform like SurveyMonkey or Qualtrics. Designers use Juji’s GUI to enter a set of interview questions and Juji will automatically generate a publicly accessible interview chatbot with a set of default conversational skills [67]. Juji also automatically handles side talking and keeps a conversation on track to ensure the completion of an interview [65].

3.2 Supporting Non-IT Designers

For non-IT chatbot designers, Juji relieves them from implementing many needed AI skills of interview chatbots while providing them with much freedom to customize a conversational experience. Specifically, Juji offers a graphical user interface (GUI) for chatbot designers to create, deploy, and manage their custom chatbots without coding [59]. Below are some of the common chatbot customizations supported by Juji GUI.

**Customize chatbot questions/messages.** Not only can designers easily add/edit/delete text-based chatbot questions, but can also customize questions or messages by adding paraphrasing, inserting URLs or images, and using functions in such text (Fig 3(a)). For example, one can insert a function to retrieve an interviewee’s name to personalize a conversation.

**Customize chatbot responses and persona.** Designers can customize chatbot responses to user input by either directly reusing Juji pre-built conversations [28] or defining their own if-then statements. For example, Fig 3(b) shows such a customization. It states that if interviewees’ responses contain positive sentiment, the chatbot would then acknowledge such input accordingly. Additionally, one can customize chatbot persona or the pace of a conversation.

**Access interviewee responses.** To help designers monitor interview progress and make design adjustments, Juji provides designers with an interactive report dashboard that displays interviewee responses visually Fig 3(c). These responses are automatically extracted from the interview conversations. Designers can also download all interviewee responses into a CSV file that contains all the question-response pairs collected. Because non-IT designers can use Juji to create working interview chatbots, Juji platform is suitable for testing whether our approach can assist any chatbot designers to improve their chatbot iteratively.

3.3 Supporting Easy Integration and Public Access

Lastly, the extensibility of Juji makes it easy for us to build and integrate iChatProfile. Specifically, Juji provides APIs for developers to access chatbot services and extend Juji chatbots with third-party functions [27]. Moreover, Juji is publicly available and provides an easy access for our study participants and also for others who wish to replicate or extend our work.

4 FORMATIVE STUDY AND DERIVED ICHATPROFILE DESIGN GUIDANCE

To guide the development of iChatProfile, we conducted a study to first identify the types of design assistance desired by chatbot designers.

From a public university, we recruited five students (3 males and 2 females, age ranges from 18 to 31) who were interested in building chatbots. None of them reported any prior chatbot design experience. Three of them interacted with conversational agents like Siri or Amazon Alexa. Our study was a semi-structured, face-to-face online interview. Each interview lasted about an hour and each participant received $20 for their time. At the beginning of each interview, the participants were asked about their past chatbot design experience. The participants were then given a 15-minute tutorial of Juji. They were also encouraged to try different Juji features and get themselves familiar with the Juji GUI. After the tutorial, the participants were asked to use Juji to design an interview chatbot that elicits user input about the COVID-19 pandemic. They were given a list of questions on this topic (Table 1).

We selected this set of interview questions for three reasons. First, we wanted to ensure the practical value of our tool development
Table 1: Interview Questions Used to Build a Chatbot

| Q1 | How are you feeling today? |
| Q2 | Where are you located? |
| Q3 | What do you do outside work? |
| Q4 | What are the challenges you currently face? |
| Q5 | What do you think you can do to help w/ this pandemic? |

effort, which is to help designers build interview chatbots for real-world uses (e.g., practical user research). Second, we wanted the interview questions to be posed to a wide audience who would be interacting and evaluating the designed chatbots. Third, COVID-19 is a pressing topic that satisfies both criteria.

In this study, we intentionally did not set any specific design requirements because we wished to observe what the participants would do and the challenges they would face. Each participant was allotted 30 minutes to design their interview chatbot. The allotted time was determined based on the results of a pilot study where 3 participants could accomplish such a design task well within 30 minutes. After completing the task, participants were interviewed to discuss the types of design help they had hoped to receive during their design process. We transcribed the audio conversations from these discussions.

We followed qualitative analysis methods and the grounded theory [40, 68] to code the participant interview data. During the first pass, two coders individually reviewed and coded participants’ responses. They then met and discussed their respective codes to identify common themes and reconcile differences. Below we report the main findings, which influenced the design of our computational framework for evaluating the performance of interview chatbots as well as the development of iChatProfile.

4.1 Two Types of Design Assistance Wanted

During the participant interviews, all participants expressed the importance of receiving design assistance. Our analysis also revealed two main types of design assistance that the participants wanted. The first type (T1) is objective, quantitative feedback on their existing chatbot design that could help designers understand the chatbot deficiencies and point them to the right directions to improve their chatbot. Almost all participants expressed the need for receiving such feedback on their chatbot. For example, one participant mentioned that “I hope to receive some feedback telling me the exact score my chatbot will get ... Just like those user ratings on the website of Alexa skills.”

The second type (T2) is design suggestions for improving a chatbot. Almost all participants expressed that they still would not know what to do even if a quantitative evaluation was available. For example, one participant stated “I am really new to this (interview chatbot design). I am afraid even I was told this part should be improved, I still don’t know how. More specific design suggestions would be of great help.” This indicates that designers also wish to receive concrete and actionable design suggestions that could guide them to improve a chatbot.

In addition to obtaining design guidance, the participants also expressed the need of viewing relevant conversation examples in the chatbot “debugging” process. For example, one participant stated “When doing chatbot response customization, I thought a lot about the wording choice since we all know that everyone’s having a hard time during this pandemic. I hoped my chatbot can always be empathetic but I have to admit it might not be the case due to so many different real-world cases.” In such a case, providing designers with the actual conversation fragments (evidence) might help them better grasp the conversation situations and improve their chatbot. In other words, augmenting design suggestions with real conversational examples would also be helpful.

4.2 iChatProfile Design Guidance

Based on the desired design assistance, we derived three design goals of iChatProfile so it can fulfill designers’ needs:

- Evaluate the performance of an interview chatbot quantitatively and present the evaluation results in a structured way. (G1)
- Provide specific, actionable design suggestions based on the evaluation results to help a chatbot designer improve the chatbot. (G2)
- Augment design suggestions with evidential conversation examples to guide a chatbot designer to make design choices. (G3)

In addition to the three goals directly determined from the findings of our formative study (G1 from T1, G2+G3 from T2), we derived another two criteria to guide the implementation of iChatProfile for practical purposes:

- Adoption. Ensure that non-IT experts can easily utilize iChatProfile. (C1)
- Compatibility. Ensure that iChatProfile can be utilized regardless which chatbot platforms are used for designing chatbots. (C2)

We derived C1 as iChatProfile is intended to help chatbot designers especially those with no AI/NLP expertise to design, evaluate, and improve interview chatbots. As a result, our effort will help democratize the applications and adoption of conversational AI. The purpose of C2 is to enable iChatProfile to be used with a wide range of chatbot platforms beyond Juji and benefit more designers. Following the goals (G1-G3) and the criteria (C1-C2), we designed iChatProfile as discussed in section 6.

5 COMPUTATIONAL FRAMEWORK FOR QUANTIFYING INTERVIEW CHATBOT EFFECTIVENESS

Since our formative study indicated that chatbot designers wish to obtain certain quantitative feedback on the performance of their existing chatbot (T1), we first formulated a computational framework that quantitatively measures the effectiveness of such an interview chatbot from multiple aspects. The framework aims at achieving two goals: 1) providing quantified insights into the performance of an interview chatbot; 2) using such insights to provide specific and practical design suggestions for improving the chatbot.

Based on the previous work on assessing human interviews [3, 7, 15, 20, 24, 50], communication theories for conducting effective interviews [47, 67], and evaluating chatbot effectiveness [12, 19, 55],
we formulated a set of performance metrics to quantitatively assess the effectiveness of an interview chatbot from three main dimensions: elicitation ability, user experience, and ethics.

To ensure both the coverage and practicality of chatbot evaluation, we used four criteria to choose our metrics. First, we selected only metrics that can be used to generate design suggestions and help designers improve an interview chatbot. Second, we chose metrics to measure both a chatbot’s abilities to complete an interview task effectively (elicitation abilities) and a user’s experience with the chatbot (user experience) because an ideal interview chatbot should be able to complete interview tasks while delivering a satisfactory user experience. Moreover, we included metrics to evaluate the ethics of an interview chatbot because such a chatbot might engage with a user in a conversation on private and sensitive topics [3, 67]. Third, we chose metrics to measure the performance of an interview chatbot both “locally” (interview question level) and “globally” (interview level). For example, the metric informativeness measures the amount of information conveyed by user responses to each interview question, while the metric user sentiment measures a user’s overall interview experience with a chatbot. This is to ensure specific design suggestions can be generated to help designers improve a chatbot question by question (locally), while providing designers with an assessment of the overall interview experience (globally). Fourth, we chose only metrics that can be easily obtained/computed from available data (e.g., chat transcripts). This is to facilitate real-time, automatic assessment of chatbot performance and design suggestion generation. Table 2 summarizes all the metrics.

5.1 Elicitation Ability

The primary task of interview chatbots is to elicit high-quality responses from participants. Existing literature shows that the success of an interview is often determined by two aspects: the elicited response quality and level of user engagement [7, 15, 19, 43, 55, 65]. We thus model an interview chatbot’s elicitation abilities from two sub-dimensions: response quality and user engagement. While response quality directly assesses the quality of user responses to an interview question, the level of user engagement quantifies how much a participant is engaged with a chatbot from multiple aspects (e.g., how long an engagement is).

5.1.1 Response Quality

We developed a metric to evaluate the quality of user interview responses.

Informativeness. This metric indicates how much information a user’s text response contains. Similar to the metric used in [67], we measure a word’s surprisal—a word’s rareness appearing in modern English [63]. To enable easy reuse of our metric regardless which English dictionary is used, we compute the informativeness of a user input \( U \) as a sum of the normalized surprisal of each word in \( U \):

\[
\text{Informativeness}(U) = \sum_{n=1}^{N} \frac{\text{surprisal(word)}_n - \text{min_surprisal}}{\text{max_surprisal} - \text{min_surprisal}}
\]

Here \( \text{min_surprisal} \) and \( \text{max_surprisal} \) are the minimum and maximum of surprisal, computed among all words in the vocabulary. \( N \) represents the word count within the response. Currently, we use the Wikipedia Corpus [1] to estimate word frequency.

This metric (e.g., a low informativeness score) can signal designers that there are potential issues with an interview question. For example, a question might be too broad and follow-up questions are needed to elicit more informative responses.

5.1.2 Level of User Engagement

In the context of interviews, the level of user engagement measures a user’s behavior during an interview [15]. Specifically, we have defined a set of metrics to assess a respondent’s behavior when engaging with an interview chatbot.

Response Length. This metric computes the word count of a respondent’s free-text response to an interview question. We chose this metric because previous work indicates that engaged respondents are more willing to give long responses [67]. Designers can use this metric to gauge their chatbot performance and to make corresponding design improvements (e.g., adding follow-up questions or changing a yes/no question to an open-ended question to elicit longer responses).

Engagement Duration. This metric indicates how long a participant is willing to engage with an interview question. Although engagement duration alone does not signal the quality of user responses [67], we hope to use it as an indicator of potential issues with an interview question. For example, if the engagement duration of a particular open-ended interview question is exceedingly short, it might signal that the question is too narrow and needs to be rephrased to encourage more open and longer engagement.

Completion Rate. This metric computes the percentage of participants completing an interview question or an entire interview. It is a commonly used metric to measure the effectiveness of an interviewer [7, 20]. To better help designers improve their chatbots question by question (see Section 6), we compute the completion rate for each interview question \( Q \) by counting the number of users who completed the question \( (C_q) \) and the number of users who responded to the question \( (T_q) \):

\[
\text{Completion Rate}(Q) = \frac{C_q}{T_q}
\]

For the first interview question (when \( n = 1 \)), we directly use the number of participants as the numerator. This metric (e.g., a low completion rate) can be used to signal potential issues related to an interview question (e.g., too vague) or the chatbot’s inability to handle user responses to the question (e.g., user’s expressed unwillingness to answer this question). A low interview-level completion rate could also reflect potential issues with an interview (e.g., too many questions). Corresponding design suggestions can then be made to help the designers improve the chatbot.

5.2 User Experience

Informed by literature in interaction design [54] and interview design [7, 20], we proposed five metrics to measure a user’s experience with an interview chatbot.

User Satisfaction Rating. This metric is directly computed from participants’ ratings of their chatbot interview experience. This rating can be easily obtained: when piloting an interview chatbot, a question like “How satisfied are you with the interview experience?”
can be added at the end of an interview session for a participant to report their level of satisfaction.

**User Trust Rating.** This metric measures participants’ perceived trust in an interview chatbot. Trust is important because it affects participants’ willingness to share information [37]. Similar to obtaining the user satisfaction rating, a question like "How much do you trust this chatbot? Please rate it on a scale of 1 to 5" can be added at the end of an interview when piloting an interview chatbot.

**User Sentiment.** This metric evaluates participants’ sentiment toward an interview chatbot since such a metric is widely used to measure user satisfaction with interviews/surveys [7]. To obtain user sentiment, one can elicit participants’ rationale (why) when eliciting their satisfaction rating and trust rating during pilot interviews. Currently, we use the Vader model [16] to perform sentimental analysis on the collected users responses, and compute the percentages of positive, neutral and negative responses.

**Level of Empathy.** This measures the level of empathy an interview chatbot has since research shows that an empathetic chatbot is able to elicit higher quality responses [65]. Currently, we compute the level of empathy by the frequency of empathetic words used by a chatbot. Specifically, given a conversation segment associated with an interview question \( Q \), we normalize the number of repeated bi-grams \( R_c \) over the total number of bi-grams \( T_c \) of chatbot’s utterances:

\[
\text{RepetitionRate}(Q) = \frac{R_c}{T_c}
\]  

### 5.3 Ethics

An interview chatbot may engage participants in a conversation on private or sensitive topics or the participants may voluntarily offer private and sensitive information [14, 23, 37, 70]. It thus is important to build ethical chatbots that respect participants as well as protect their privacy. We thus have developed two metrics to evaluate the ethics of an interview chatbot.

**Hate Speech Rate.** This metric assesses how much an interview chatbot includes hate speech in its utterances. Such assessment becomes even more important if a chatbot uses auto-synthesized responses as what Tay was using [58]. Currently, we use an automated hate speech detection algorithm to compute the hate speech rate [10]. This metric can help chatbot designers be better aware of a chatbot’s built-in AI capabilities and correct a chatbot’s behavior if needed.

**Privacy Intrusion Rate.** This metric evaluates how much an interview chatbot elicits private or sensitive information from a participant (e.g., password or social security number). Currently, for each interview question, we first identify "sensitive" words/phrases appearing in chatbot utterances or user responses using Google’s Data Loss Prevention (DLP) API [9]. These words, such as a social security number, might risk a user’s privacy. Given a conversation segment associated with an interview question \( Q \), we compute...
the rate as follows:

\[ \text{PrivacyIntrusion}(Q) = \frac{S_c}{T_c} \] (5)

Here \( S_c \) is the count of sensitive words appearing in the user responses and \( T_c \) is the total word count in user responses.

To better protect a user’s privacy, chatbot designers can use this metric to curb an interview chatbot from eliciting such information or reminding a user of not giving up such information unnecessarily during an interview.

6 ICHATPROFILE

To help chatbot designers evaluate and improve an interview chatbot iteratively, we have developed a tool called *iChatProfile*, following the design goals and design criteria summarized in Sec 4.2. It automatically computes the metrics (Table 2) to assess the performance of an interview chatbot. Each chat transcript is first segmented by interview question and each segment consists of one or more conversation turns (Fig 4(b)). Each metric (e.g., response length, except completion rate and user sentiment) is first computed/extracted per transcript (user) and all the scores are then averaged across all transcripts (users). The completion rate and user sentiment are directly calculated from all the transcripts (e.g., Formula 2).

6.1 System Overview

As shown in Fig 4(a), *iChatProfile* consists of three key components: chatbot profile generator, design suggestion generator, and profile presenter. The *chatbot profile generator* takes a set of chat transcripts as input and automatically computes all the chatbot performance metrics (Table 2) to generate a chatbot profile. In general, chat transcripts are the results of the live chats between an interview chatbot and its pilot/testing users. Once a chatbot profile is created, the *design suggestion generator* automatically generates a set of specific suggestions with conversation evidence for improving the chatbot. The chatbot profile, the design suggestions, and the conversation evidence are then assembled together by the *profile presenter* and displayed in a visual dashboard for easy comprehension (C1). To make *iChatProfile* easily work with any chatbot platforms, it is implemented as an independent tool and takes only chat transcripts as its input (C2).

6.2 Profiling Interview Chatbots

Given a set of chat transcripts, *iChatProfile* automatically computes all the metrics mentioned in Section 5 to assess the performance of an interview chatbot. Each chat transcript is first segmented by interview question and each segment consists of one or more conversation turns (Fig 4(b)). Each metric (e.g., response length, except completion rate and user sentiment) is first computed/extracted per transcript (user) and all the scores are then averaged across all transcripts (users). The completion rate and user sentiment are directly calculated from all the transcripts (e.g., Formula 2).

6.3 Generating Design Suggestions

Given the computed performance metrics, *iChatProfile* automatically generates a set of design suggestions using a rule-based approach. To make each design suggestion actionable, we formulate rules based only on the question-level evaluation metrics (e.g., informativeness). Such a design suggestion can be used by a chatbot designer to further customize and tweak chatbot behavior around a specific interview question. In contrast, it is difficult for designers to act upon an interview-level metric, such as user satisfaction rating, although it informs the designers the overall performance of a chatbot.

Below is an example rule. It states that if the computed repetition rate for an interview question \( Q \) is above a certain threshold, it then uses a template to generate a set of design suggestions that could reduce repetitions and improve user experience.

1: if repetition rate\((Q) > \text{threshold} \) then
2: generate-design-suggestions (reduce-repetition-template)
3: end if

In our current implementation, the default thresholds are determined by the corresponding metric scores of the opening question (Q1). This is because a recent study shows that the conversation around the very first question could be used as a good indicator [67]. The only exception is for hate speech rate, where the threshold is set to 0. It means that if any hate speech is detected, design suggestions will be generated. Additionally, the thresholds can also be defined by designers themselves based on their needs.

Once a rule is triggered, *iChatProfile* automatically generates actionable design suggestions in two steps. First, it uses a template-based approach to generate design suggestions in natural language
[41]. Second, it automatically extracts relevant conversation fragments as evidence to substantiate the generated design suggestions.

6.3.1 Template-based Natural Language Generation. For each metric, we have defined a template that contains one or more design guidelines for improving a chatbot (Table 3). These design guidelines are formulated based on previous research findings and commercial product design guidelines (Alexa, Google Home and Juji) for improving interview quality and user experience [8, 11, 14, 22, 28, 44, 45, 55, 65, 70]. For example, there are two guidelines on improving the metric informativeness: one is to better articulate or explain an interview question to minimize ambiguity, while the other is to improve a chatbot with active listening skills to make users feel heard [65].

Given a template, it takes two steps to generate design suggestions in natural language: document planning and surface realization [41]. In document planning, we define the content to be conveyed in four parts: (a) the design guideline (D), (b) the corresponding interview question (Q), (c) the corresponding metric (M), and (d) an explanation on why the design guidelines are given. In surface realization, we generate natural language statements by a template: “For question Q, do D because metric M is Z”. Here Z is either “too low” or “too high”, depending on which metric value triggers the generation. We have used a python library SimpleNLG [26] to automatically generate grammatically correct natural language sentences. The library helps organize basic syntactic structure (e.g., tense) and sentence elements (e.g., punctuation).

Using the example rule mentioned above, assume that the computed repetition rate for the interview question “Where are you located?” exceeds the threshold, signaling potential issues around this interview question. This triggers iChatProfile to generate a set of design suggestions as shown in Fig 1(b).

6.3.2 Conversation Evidence Extraction. To act on design suggestions, chatbot designers may need more information to understand the conversation situations. As we learned from the formative study, it is difficult for designers to anticipate conversation situations. Continuing the above example, although a chatbot designer now knows that the interview question “Where are you located?” has caused high repetitions, s/he might not know what caused the repetitions. In such a case, providing designers with the actual conversation fragments (evidence) might help them better grasp the situations and make chatbot improvements.

Thus, iChatProfile automatically extracts relevant conversation fragments from chat transcripts to give designers more concrete ideas on how to improve a chatbot. These conversation fragments are essentially concrete evidence to show chatbot designers what went wrong. However, such fragments might be too many, which would not help the designers but overwhelm them. We have thus used GloVe embeddings [48] and a k-means algorithm [33] to select the most representative conversation fragments in three steps.

Given an interview question (e.g., “where are you located”) and a performance metric (e.g., repetition rate), iChatProfile first selects all conversation segments (Fig 4(b)) that produced a metric score worse than the threshold. These selected segments are then encoded by GloVe embeddings. Second, these segments are grouped into k clusters based on their cosine similarity. Elbow method is used [32] to find the optimal number of clusters (k). Third, iChatProfile then ranks the clusters by coverage (i.e., the number of segments in each cluster). Within the top-K clusters, one conversation segment is randomly selected per cluster as the representative evidence to substantiate the design suggestions. Currently, K is determined by the available space in the visual dashboard after displaying the design suggestions. The rest of the conversation segments can also be accessed through a hyperlink.

Using the above example on the interview question of “Where are you located”, two clusters are formed, one with the coverage of 75.0% and the other 25.0%. Assuming K=2, iChatProfile selects one conversation segment from each cluster (Table 4 and Table 5).

| Table 4: Conversation Segment from Cluster 1 |
|---------------------------------------------|
| Juji: Where are you located?                 |
| User: You tell me first.                    |
| Juji: Where you are located?                |
| User: What about you?                       |

| Table 5: Conversation Segment from Cluster 2 |
|---------------------------------------------|
| Juji: Where you are located?                |
| User: no                                    |
| Juji: Where are you located?                |
| User: I don’t want to                       |

6.4 Presenting Chatbot Profile and Design Suggestions

To present a generated chatbot profile, design suggestions, and relevant conversation evidence, we used Tableau [57] to implement a web-based, interactive visual dashboard (Fig 1). A displayed chatbot profile consists of all computed metrics visualized in various forms depending on the type of information. For example, response length is visualized in a bar chart while user sentiment is displayed in both a pie chart (showing the percentages of each type of sentiment) and word clouds (Fig 1(a)). The profile also visually indicates the thresholds that would trigger design suggestions, which helps designers better understand the meanings of metric scores and make design decisions. Users can interact with each metric to view corresponding design suggestions if there is any. If a performance metric (e.g., informativeness) deems to be improved, iChatProfile presents the generated design suggestions and conversation evidence (Fig 1(b)).

7 EVALUATION

To evaluate the effectiveness of iChatProfile, we designed and conducted a between-subject user study that compared the performance of 10 chatbots designed with or without using iChatProfile.

7.1 Study Method

Using the same set of interview questions about COVID-19 shown in Table 1, we first built an interview chatbot on Juji using only Juji’s built-in features without making any customization. After asking all the interview questions, the chatbot also included questions to elicit user satisfaction rating and trust rating, as well as their rationale behind each rating. This chatbot served as our baseline. The baseline chatbot was deployed on the web to engage with respondents in a live chat. The pilot study collected a total of 128 chat transcripts. Using these transcripts, iChatProfile automatically generated a chatbot profile and corresponding design suggestions to improve the baseline chatbot. Since we wished to compare chatbot performance with and without using iChatProfile, we recruited 10 chatbot designers, who were
randomly divided into two groups, 5 in each group. Each designer started with a 15-minute tutorial of the Juji platform by watching a tutorial video and learning several key Juji features (e.g., how to customize a chatbot’s actions). They were given additional time to play with Juji and get familiar with various design features. Each designer was then given the baseline chatbot for them to import into their own account so they could preview and improve the baseline. They also had access to the report dashboard and the interviewee responses extracted from the 128 conducted chatbot interviews as described in section 3.2. They were asked to describe the good and bad aspects of the baseline chatbot. Next, they were asked to improve the baseline chatbot along three dimensions: user response quality, user experience, and ethics. They were allowed to use any chatbot customizations (e.g., rewording a question or customizing a chatbot’s actions) as long as all the original interview questions and the question order were kept. All the designers in one group (Group B, w/ iChatProfile) were also given iChatProfile to view the generated profile of the baseline chatbot and corresponding design suggestions, while the other group (Group A, w/o iChatProfile) was not given the tool but only the interviewee responses. We also collected the participants’ demographics, including their gender and age, and their chatbot experience (chatbot interaction or design experience).

Each designer was allotted about 30 minutes to improve their chatbot. A post-task interview was also conducted. The designers in Group w/o iChatProfile were asked about their design and the challenges/difficulties they faced during their chatbot design process. The designers in Group w/ iChatProfile were asked to describe their design and their experience of using iChatProfile. Because of the COVID-19 pandemic, the whole study was conducted online via a 1:1 Zoom meeting. On average, each study session lasted about an hour.

Ten (10) designers from the two groups built a total of 10 chatbots based on the baseline chatbot provided to them. Each of these chatbots was deployed on the web to engage with respondents in a live chat.

### 7.2 Participants

All chatbot respondents, including the ones in the pilot study, were recruited on Amazon Mechanical Turk (MTurk) with an approval rating equal to or greater than 99% and located in the U.S. or Canada. Each participant was paid $12.5/hr.

The 10 chatbot designers (6 males, 4 females, ages 20 to 35) were students recruited from a public university majoring in diverse disciplines, including Computer Science, Information Science, Psychology, and Environmental Studies. Two (2) participants reported prior experience of building chatbots, which five (5) of them reported prior experience of interacting with conversational agents, like Siri, Amazon Alexa or Google Assistant. None of them had built interview chatbots or used Juji. Each participant was paid $20 for their time.

### 7.3 Study Results

From the 10 deployed interview chatbots, we collected a total of 1349 interview transcripts including the transcripts of incomplete interviews. We kept the incomplete ones because they could indicate the performance of a chatbot. On average, each chatbot interviewed 135 users (135 chat transcripts). Given their respective interview transcripts, iChatProfile computed ten (10) chatbot profiles to characterize the performance of each of the chatbots. We then compared the computed chatbot performance between the 5 chatbots (702 transcripts) designed by the participants in Group w/o iChatProfile and another 5 chatbots by Group w/ iChatProfile (647 transcripts) using the tool.

#### 7.3.1 ANCOVA Analyses

Specifically, we performed a series of ANCOVA analyses, which blend analysis of variance (ANOVA) and regression [30], to examine the effect of with or without using iChatProfile (independent variable) on various chatbot performance metrics (dependent variables). We ran ANCOVA analyses on every metric in Table 2 except three due to a lack of data: completion rate (10 samples), user sentiment (10 samples), and hate speech rate (no data). Both completion rate and user sentiment were computed per chatbot (a total of 10 chatbots) and our algorithm did not detect any hate speech in any of the chatbots.

We compared the chatbot performance between Group w/o iChatProfile (702 transcripts) and Group w/ iChatProfile (647 transcripts). The assumption check was conducted to make sure the unequal sample size would not affect the reliability of the results. In each analysis, the independent variable was the group and the dependent variable was one of the chatbot performance metric

### Table 3: Metric-based chatbot design guidelines.

| Design guideline                                      | Metric                                | Reference(s)          |
|-------------------------------------------------------|---------------------------------------|------------------------|
| Add polite probings and explanations to the question  | Informative, Completion Rate          | [11, 45]               |
| Add customizations to show the chatbot is actively listening | Informativeness, Engagement Duration | [65]                   |
| Set a minimum response length to handle short user input | Response Length, Engagement Duration  | [28]                   |
| Add customized chatbot responses to handle user digressions | Response Length, Engagement Duration, Repetition Rate | [55, 65, 67]          |
| Reward the question to make it more acceptable to users | Completion Rate, Repetition Rate      | [11, 14, 22]           |
| Personalize the chat experience, e.g., addressing users their names | Completion Rate                      | [14, 22, 28]           |
| Add default empathetic chatbot responses to handle unknown user input | Level of Empathy                     | [14, 28, 70]           |
| Customize chatbot responses to give empathetic feedback on user input | Level of Empathy                     | [14, 70]               |
| Remove all the hate or offensive speech                | Hate Speech Rate                      | [14]                   |
| Avoid asking private or sensitive information without user consent | Privacy Intrusion Rate                | [14]                   |
Before running ANCOVA analyses, we also examined the correlations among all dependent variables. Consistent with prior findings [67], informativeness was not correlated with engagement duration. Moreover, engagement duration did not significantly correlate with any other metrics except repetition rate. We also noted that a chatbot’s empathy level significantly correlated with informativeness and response length. Intuitively, this result is sensible since respondents would be more cooperative with an empathetic chatbot [65].

7.3.2 iChatProfile helped designers build better chatbots (RQ1). Table 6 and Table 7 summarize the analysis results. The results show that iChatProfile helped create chatbots that performed significantly better at both interview and individual question level. At the interview level, for example, the chatbots in Group B achieved a higher completion rate (78% vs. 74% vs. 74%) and more positive user sentiment (66% vs. 45% vs 46%), than those in Group w/o iChatProfile and the baseline.

At the question level, the chatbots in Group w/ iChatProfile also performed better than those in Group w/o iChatProfile and the baseline on almost all dimensions, including response quality (informativeness), user engagement (response length and engagement duration), and user experience (level of empathy and repetition rate). Only the difference in privacy intrusion rate is insignificant. This is because the interview questions (Table 1) did not elicit much private or sensitive user information. Our results also indicated that the performance differences between the chatbots in Group w/o iChatProfile and the baseline are mostly insignificant. In fact, Group w/o iChatProfile performed worse than the baseline on certain metrics, such as completion rate and user satisfaction rating. This implies that designers had difficulty improving a chatbot without any specific design guidance. Moreover, making improvements without knowing chatbot deficiencies could even hurt the chatbot performance. For example, a designer in Group w/o iChatProfile added a follow-up question “What made you feel that way” to interview question Q4 “What challenges are you facing?”. But he ignored user responses that already talked about their feelings when answering Q4, which made users feel unheard. No wonder one user commented “I have already stated that, you were unable to understand answers.”

In all the analyses, the use of iChatProfile was a significant factor impacting the chatbot performance differences. Three control variables, Q1, gender, and chatbot experience, were shown significant for a few analyses, although none of these control variables had interaction effect with the use of iChatProfile. Control variable Q1 was significant for informativeness, response length, and engagement duration. Since Q1 was used to account for respondents’ differences [67], the effect of Q1 implies the effect of respondents’ behavior (e.g., uncooperation) on interview quality, consistent with previous findings [67]. In addition, control variable gender significantly impacted a chatbot’s empathy level, and one’s chatbot experience influenced informativeness and response length. It is interesting that the chatbots made by male designers were more empathetic than those made by female designers (Male 0.026 vs. Female 0.003, p<0.05). Although one’s chatbot experience helped make chatbots better at eliciting information (e.g., informativeness), it had no effect on user experience, such as user satisfaction rating or user trust rating.

Similarly, we ran ANCOVA analyses to compare each chatbot’s performance metrics between Group w/o iChatProfile and the baseline (128 transcripts), and between Group w/ iChatProfile and the baseline, respectively. Bonferroni correction was applied to adjust p values.

Since all the designers were given the same goal to improve a chatbot along three dimensions, our analyses were to answer two questions:

- **RQ1**: Did iChatProfile help designers build better interview chatbots?
- **RQ2**: How did iChatProfile help make chatbot design decisions?
### Table 7: ANCOVA analysis results for chatbot performance metrics between baseline, Group A (w/o iChatProfile), and Group B (w/ iChatProfile).

| Metrics              | Group B vs. Group A | Group B vs. Baseline | Group A vs. Baseline |
|----------------------|---------------------|----------------------|----------------------|
|                      | F       | p       | η²p    | F       | p       | η²p    | F       | p       | η²p    |
| Informativeness      | 24.28 (1, 1094) | < 0.001 ** | 0.022  | 12.59 (1, 677) | < 0.01 ** | 0.021  | 0.73 (1, 689) | > 0.05  | 0.001  |
| Response Length      | 28.60 (1, 1094) | < 0.001 *** | 0.026  | 7.78 (1, 677) | < 0.05 *  | 0.024  | 0.48 (1, 689) | > 0.05  | 0.001  |
| Engagement Duration  | 6.19 (1, 1094) | < 0.05 *  | 0.004  | 9.37 (1, 677) | < 0.01 ** | 0.007  | 6.47 (1, 689) | < 0.05 * | 0.004  |
| User Satisfaction Rating | 6.68 (1, 1032) | < 0.05 *  | 0.006  | 5.99 (1, 615) | < 0.05 ** | 0.006  | 4.89 (1, 611) | > 0.05  | 0.007  |
| User trust rating    | 5.95 (1, 1035) | < 0.05 *  | 0.004  | 8.72 (1, 616) | < 0.01 ** | 0.007  | 2.04 (1, 613) | > 0.05  | 0.004  |
| Level of Empathy     | 6.07 (1, 1096) | < 0.05 *  | 0.004  | 13.97 (1, 679) | < 0.001 *** | 0.023  | 13.97 (1, 691) | < 0.001 *** | 0.023  |
| Repetition Rate      | 14.30 (1, 1096) | < 0.001 *** | 0.012  | 8.206 (1, 679) | < 0.05 *  | 0.013  | 0.821 (1, 691) | > 0.05  | 0.013  |
| Privacy Intrusion Rate | 0.067 (1,1095) | > 0.05  | 0.000  | 1.378 (1, 678) | > 0.05  | 0.003  | 0.158 (1, 690) | > 0.05  | 0.000  |

a. p-value in this table is Bonferroni corrected.
b. All analyses were controlled for designers’ differences (gender and chatbot experience). Whenever applicable, analyses (e.g., informativeness) were additionally controlled for respondents’ differences (the corresponding metric in Q1).
c. informativeness and length were additionally controlled for engagement duration; and engagement duration was controlled for length [65].

#### 7.3.3 iChatProfile helped designers make chatbot design decisions (RQ2)

During the post-task interviews, all participants from Group w/ iChatProfile confirmed the helpfulness of iChatProfile and benefited from the displayed chatbot profile, design suggestions, and evidential conversation examples. Specifically, a chatbot profile provided designers with an overview of the chatbot performance. They liked the visual display because it provided them with *straightforward insights*. They also mentioned that they used the metric scores shown in the chatbot profile (Fig 1(a)) to quickly locate design deficiencies in the baseline chatbot.

**Guiding Designers to Make Practical Chatbot Improvements**

Recall that at the beginning of their task, all designers were asked to comment on the baseline chatbot and their plan to improve it. All of them gave vague descriptions or improvement plans. However, after the designers in Group w/ iChatProfile had access to iChatProfile, they seemed knowing what they needed to do. On average, the designers in Group w/ iChatProfile did 49% (99 vs. 20) of chatbot customizations compared to those in Group w/o iChatProfile (Table 8).

Specifically, the designers in Group w/ iChatProfile appreciated the design suggestions and evidential conversation examples. We examined the chatbots made by the designers in Group w/ iChatProfile and observed that all of them followed one or more design suggestions given by iChatProfile. For example, the two design suggestions, “add customizations to show the chatbot is actively listening” and “reword the question to make it more acceptable to users”, were followed by all 5 designers in Group w/ iChatProfile to customize and improve the baseline chatbot behavior around interview questions Q2 and Q4 (Table 1). In addition to following the design suggestions, the designers in Group w/ iChatProfile also found the conversation examples very helpful. While the design guidelines informed designers what to do (e.g., rewording a question), the conversation examples helped them figure out how to do it. For example, Q2 “where are you located” asked respondents about their location. When chatting with the baseline chatbot, some respondents were unclear about the question, which caused a higher repetition rate. iChatProfile generated a design suggestion “reword the question to make it more acceptable to users” with conversation examples (Table 4). One designer who followed the suggestion reworded the original question to “May I ask where are you located? No need to be very specific, just city name would do :)”. He stated in the post-task interview that “I read the bad example and realized that some people might not like this kind of questions directly asking for their personal information, so I changed it.”

In comparison, the chatbot customizations made by designers in Group w/o iChatProfile were fewer and with a higher percentage of unmatched chatbot customizations (43% vs. 33% in Group w/ iChatProfile) — customizations that were not suggested by iChatProfile. Table 8 shows the design suggestions given by iChatProfile and the chatbot customizations made by designers two groups, respectively. From our observations, none of the designers from Group w/o iChatProfile went through all the 128 interviewee responses. They mainly reviewed the report dashboard and randomly selected a few responses to examine, without specifically knowing what to learn from the interview results let alone how to improve the chat based on the results. Without any guidance from a tool like iChatProfile, the designers in Group w/o iChatProfile made their customizations based on their intuition or ad hoc reasons. For example, when asked “why did you decide to add these customizations”, one designer in Group w/o iChatProfile said “Because the functionality of adding customization provided by Juji is a large block (a big area of the interface). It is very noticeable and I decided to try it out.”. Another designer also stated that “Juji provides so many features to choose from and I’m not sure which one to use ... I decide to add them all in the end.”. Since the designers in Group w/o iChatProfile didn’t use iChatProfile, a lack of guidance for identifying chatbot deficiencies or improvements definitely contributed to their inferior chatbot performance.

**Inspiring Designers to Make Creative Chatbot Improvements**
Table 8: A summary of chatbot customizations made by designers in Group A (w/o iChatProfile) and Group B (w/ iChatProfile)

|                         | Group A |          |          | Group B |          |          |
|-------------------------|---------|----------|----------|---------|----------|----------|
|                         | Mismatched | Matched | Total count | Mismatched | Matched | Total count |
| Improving response quality | 5 (55.6%) | 4 (44.4%) | 9         | 10 (45.5%) | 12 (54.5%) | 22         |
| Improving user engagement | 5 (71.4%) | 2 (28.6%) | 7         | 15 (42.9%) | 20 (57.1%) | 35         |
| Improving user experience | 0 (0%) | 4 (100%) | 4         | 4 (9.5%) | 38 (90.5%) | 42         |
| Improving ethics | 0 | 0 | 0 | 0 | 0 | 0 |

*Mismatched*/*Matched* indicate whether designers’ chatbot customizations matched with what iChatProfile suggested.

In addition to guiding designers to make practical chatbot improvements, iChatProfile also inspired designers to make creative chatbot improvements beyond what was suggested by the tool. For example, one designer in Group w/ iChatProfile decided to add transitions between interview questions, "Come on, get up and do 10 bobby jumps before we continue. Cheer up!" During the post-task interview, when asked why he made such a design decision, he mentioned that "I noticed the profile shows the original design (baseline) did not engage people well. So I thought why not engage them physically?" We also checked the feedback left by the respondents who chatted with this chatbot. We noticed the positive comments such as "You (Juji) asked me to do bobby jumps. I didn’t actually do it but it’s interesting and I like it."

8 DISCUSSIONS

While our study results are encouraging, the study also revealed several limitations. Here we discuss these limitations and future work. We also briefly discuss design implications of our work on building chatbots beyond interview chatbots.

8.1 Limitations

8.1.1 Study Scope and Participants. While our results should be widely applicable for building a class of interview chatbots, the scope and the participants of our study present its limitations. Our study reported here focused on a low-stakes interview task (e.g., user or market research interviews) and recruited all the interviewees on Amazon Mechanical Turk. It is unclear whether our results would hold for different types of interview tasks, such as high-stakes tasks like job interviews with much more motivated interviewees and additional chatbot requirements (e.g., detecting faking [71]). Moreover, in our study all the chatbot designers were university students. It would be interesting to investigate how our results would hold or change with different chatbot designer groups (e.g., experienced chatbot designers).

8.1.2 Offering Finer-Grained Design Suggestions. Currently, iChatProfile often offers multiple design suggestions per performance metric. For example, it offers two suggestions if the response length is below a threshold (Table 3). However, under certain circumstances, one suggestion might be more useful than others. Using the example for improving response quality due to a vague question, adding explanations to the question would be more useful than making the question more acceptable to users. This requires that iChatProfile further discerns the causes to the chatbot performance so it can narrow down the design suggestions and recommends the most suitable one. One potential method to address this is to analyze user responses and identify different semantic themes, similar to the data-driven methods used by others to recognize the semantic themes in user input [65]. Based on the recognized themes and the computed performance metric, iChatProfile can recommend the most suitable design suggestion(s).

Additionally, iChatProfile currently produces evidential conversation examples along with design suggestions, which proved to be helpful for designers in our study. However, these examples are the "negative examples" extracted from existing chats and no positive examples are given. For example, if the design suggestion is to "give empathetic feedback", it would be helpful for a designer to see a "positive example"—what an empathetic feedback is like. Again, this would require more usage data, which will then allow iChatProfile to extract "good examples." 8.1.3 Evaluating iChatProfile Usability. Although our ultimate goal is to help designers build effective interview chatbots, we have not yet evaluated the usability of iChatProfile for two reasons. First, we want to verify its usefulness and effect before evaluating its usability. Second, our current implementation is standalone and not integrated with any chatbot platforms. Thus certain operations are cumbersome involving much manual work (e.g., manually downloading all the chat transcripts from a chatbot platform and then uploading them into iChatProfile). A more integrated version should be created and then usability evaluation makes better sense.

8.2 Future Work and Design Implications

There are several directions that we can extend iChatProfile to refine its functions and expand its uses.

8.2.1 System Explainability. As mentioned in Section 6.3.1, iChatProfile provides explanations on why certain design suggestions are given. Our study participants expressed their appreciation of such explanations. However, when system suggestions were inconsistent with designers’ belief, current explanations need to be expanded. For example, one designer stated "The score of trust level doesn’t actually reflect my experience, I am wondering why." In such cases, deeper explanations would be helpful than just stating "the score is lower than a threshold". One possible direction of future work is to construct a multi-layer framework for evaluating the performance of interview chatbots. Following [52], this framework could have three layers: design...
quality (e.g., the question-level performance metrics used in our current work), user belief (user perceived chatbot performance, such as perceived usefulness [51] and ease of use [6]), and user attitude (users’ overall feelings towards the whole chatbot, such as perceived trust and satisfaction in our work). A path model can then be generated to reveal causal relationships between different layers to make chatbot profiling and design suggestions more explainable. In particular, such relationships could explicate how design qualities may influence users’ attitude more clearly, or why a certain metric could contribute to the success/failure of the overall design through an influence path across layers [52].

8.2.2 Benchmarking Interview Chatbot Evaluation. As our studies show, it is difficult for designers to evaluate chatbot performance without any guidance. While iChatProfile helps designers make specific chatbot improvements, it could not inform the designers how much test an interview chatbot actually needs before achieving an acceptable performance. For example, it would be very valuable to inform a chatbot designer that a minimal of N pilot users or at least M rounds of evaluations are needed to test a particular interview chatbot and achieve an acceptable performance. As iChatProfile collects more usage data, offering designers with the above suggestions will become more feasible. Specifically, an evaluation benchmark can be established for each type of interview question as well as for a particular type of interview chatbot. This will also enable us to establish different thresholds for different chatbot tasks.

8.2.3 Real-Time Chatbot Evaluation and Feedback. Although we showed how iChatProfile was able to help improve chatbot performance significantly within just one iteration of design, building an effective interview chatbot often takes multiple iterations. Currently, iChatProfile generates a chatbot profile and design suggestions after a designer deploys the chatbot and collects a set of transcripts from live chats. However, designers may wish to receive prompt feedback while designing a chatbot, since early and timely feedback could improve creative work [34, 35]. To enable continuous chatbot evaluation and improvement, one approach is to integrate iChatProfile with crowdsourcing tools that can recruit testers, administer live chats, and provide a chatbot profile and design suggestions to improve the chatbot, all near real time [39]. Another approach is to employ deep learning algorithms to simulate real-world user behaviors so that chat transcripts can be obtained in real time [12] to aid the iterative evaluation and improvement of chatbots in real time.

8.2.4 Assistive Design of Chatbots beyond Interview Chatbots. Our work demonstrates the effectiveness of iChatProfile for helping designers to evaluate and improve an interview chatbot iteratively. Since interview chatbots share many characteristics as other types of chatbots, such as counseling or training chatbots, iChatProfile could be extended to help designers build such chatbots as well. Especially iChatProfile aims at helping non-AI experts in chatbot design, it could help professional coaches or trainers design, evaluate, and improve their own chatbots. We hope that our work can serve as a stepping-stone on the path to democratize chatbot design for a wide variety of applications beyond interview tasks.

9 CONCLUSIONS

We described a computational framework for evaluating interview chatbots and presented iChatProfile, a tool that helps designers to evaluate and improve interview chatbots iteratively. Given a set of chat transcripts, it automatically quantifies the performance of a chatbot and generates a chatbot profile. Based on the generated chatbot profile, it also offers design suggestions in natural language with evidential conversation examples, which help guide designers to improve the chatbot. To validate the effectiveness of iChatProfile, we designed and conducted a between-subject study that compared the performance of chatbots designed with and without using iChatProfile. Based on the transcripts collected from the live chats between 10 chatbots and 1394 users, our results show that iChatProfile helped produce interview chatbots with significantly better performance across almost all dimensions, including response quality, user engagement, and user experience.

REFERENCES

[1] Wikipedia 2020. The Wikipedia Corpus. Wikipedia. https://www.english-corpora.org/wiki/  
[2] ManyChat 2020. ManyChat. ManyChat. https://manychat.com  
[3] Peter Allmark, Jonathan Boote, Eleni Chambers, Amanda Clarke, Ann McDonnell, Andrew Thompson, and Angela Mary Tod. 2009. Ethical issues in the use of in-depth interviews: literature review and discussion. Research Ethics 5, 2 (2009), 48–54.
[4] Dan Rohus and Alexander I Rudnicky. 2009. The RavenClaw dialog management framework: Architecture and systems. Computer Speech & Language 23, 3 (2009), 352–361.
[5] Susan E Brennan. 1990. Conversation as direct manipulation: An iconoclastic view. The art of human-computer interface design n.a. (1990), 393–404.
[6] Li Chen and Pearl Pu. 2009. Interaction design guidelines on criticizing-based recommender systems. User Modeling and User-Adapted Interaction 19, 3 (2009), 167.
[7] Pu-Shih Daniel Chen. 2011. Finding quality responses: The problem of low-quality survey responses and its impact on accountability measures. Research in Higher Education 52, 7 (2011), 659–674.
[8] Frederick G Conrad, Mark P Coupier, Roger Tourangeau, Mirta Galesic, and T Yan. 2005. Interactive feedback can improve the quality of responses in web surveys. In Proc. Surv. Res. Meth. Sect. Am. Statist. Ass. Reseachgate, n.a., 3835–3840.
[9] Google Cloud 2020. Data Loss Prevention. Google Cloud. https://cloud.google.com/dlp.
[10] Thomas Davidson, Dana Warmesley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In Proceedings of the International AAAI Conference on Web and Social Media, Vol. 11. AAAI, Palo Alto, California, USA, 1.
[11] Edith D de Leeuw, Joop J Hox, and Anja Boeve. 2016. Handling do-not-know answers: Exploring new approaches in online and mixed-mode surveys. Social Science Computer Review 34, 2 (2016), 116–132.
[12] Jan Deriu, Alvaro Rodrigo, Arantza Otegi, Guillermo Echehogary, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2019. Survey on evaluation methods for dialogue systems. arXiv preprint arXiv:1905.04071 n.a. (2019), n.a.
[13] Google Cloud 2020. Dialogflow. Google Cloud. https://cloud.google.com/dialogflow.
[14] Amazon Alexa 2020. Establish and Maintain Trust. Amazon Alexa. https://developer.amazon.com/en-US/docs/alexa-design-trustbusters.html.
[15] Dana Garbarksi, Nora Cate Schaeffer, and Jennifer Dykema. 2016. Interviewing practices, conversational practices, and rapport: Responsiveness and engagement in the standardized survey interview. Sociological methodology 46, 1 (2016), 1–38.
[16] CHE Gilbert and Erric Hutto. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp.soc. gatech.edu/papers/icwsm14/ vader_hutto.pdf, Vol. 81. AAAI, Palo Alto, California, USA, 82.
[17] H Paul Geirce. 1975. Logic and conversation, syntax and semantics. Speech Acts (1975), 41–58.
[18] Jonathan Grudin and Richard Jacques. 2019. Chatbots, Humbots, and the Quest for Artificial General Intelligence. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Glasgow, Scotland Uk) (CHI ’19). Paper 209. Association for Computing Machinery, New York, NY, USA, 1–11.
[19] Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. 2018. Topic-based evaluation for conversational bots. arXiv
[70] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics* 46, 1 (2020), 53–93.

[71] Michelle X Zhou, Gloria Mark, Jingyi Li, and Huahai Yang. 2019. Trusting virtual agents: the effect of personality. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 9, 2-3 (2019), 1–36.