

Article

Enhancing Task-Oriented Dialogue Systems through Synchronous Multi-Party Interaction and Multi-Group Virtual Simulation

Ellie S. Paek *, Talyn Fan, James D. Finch and Jinho D. Choi * 

Department of Computer Science, Emory University, Atlanta, GA 30322, USA

* Correspondence: ellie.paek@emory.edu (E.S.P.); jinho.choi@emory.edu (J.D.C.)

Abstract: This paper presents two innovative approaches: a synchronous multi-party dialogue system that engages in simultaneous interactions with multiple users, and multi-group simulations involving virtual user groups to evaluate the resilience of this system. Unlike most other chatbots that communicate with each user independently, our system facilitates information gathering from multiple users and executes 17 administrative tasks for group requests adeptly by leveraging a state machine-based framework for complete control over dialogue flow and a large language model (LLM) for robust context understanding. Assessing such a unique dialogue system poses challenges, as it requires many groups of users to interact with the system concurrently for an extended duration. To address this, we simulate various virtual groups using an LLM, each comprising 10–30 users who may belong to multiple groups, in order to evaluate the efficacy of our system; each user is assigned a persona and allowed to interact freely without scripts. As a result, our system shows average success rates of 87% for task completion and 89% for natural language understanding. Comparatively, our virtual simulation, which has an average success rate of 80%, is juxtaposed with a group of 15 human users, depicting similar task diversity and error trends. To our knowledge, it is the first work to show the LLM's potential in both task execution and the simulation of a synchronous dialogue system to fully automate administrative tasks.

Keywords: synchronous multi-party dialogue system; virtual user simulation; administrative task automation; large language models



Citation: Paek, E.S.; Fan, T.; Finch, J.D.; Choi, J.D. Enhancing Task-Oriented Dialogue Systems through Synchronous Multi-Party Interaction and Multi-Group Virtual Simulation. *Information* **2024**, *15*, 580. <https://doi.org/10.3390/info15090580>

Academic Editor: Luis Martínez López

Received: 6 August 2024

Revised: 11 September 2024

Accepted: 12 September 2024

Published: 19 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Administrative tasks play a pivotal role in upholding organizational efficacy. Though they are typically overseen by human assistants, the increasing demand for swift task execution poses ongoing strains. To tackle this issue, chatbots have emerged as invaluable assets, streamlining tasks such as appointment scheduling and message dissemination [1–3]. Many dialogue systems aiming to tackle knowledge-based inquiries have also emerged, proving to be valuable assets within organizations [4–9]. Nonetheless, most current chatbots assist each user independently, despite many organizational processes requiring the assistant to engage with multiple users simultaneously to complete tasks. As a result, these chatbots often find themselves merely managing personal to-do lists and answering knowledge-based questions, leaving the bulk of the workload still for humans to handle.

Large language models (LLMs) can enable us to develop chatbots that facilitate synchronous multi-party dialogues, where multiple users are simultaneously engaged with a chatbot to fulfill their tasks. With LLMs' notable enhancements, there is a growing interest in training LLMs to build end-to-end dialogue systems as well as in augmenting them to improve upon dialogue systems [7,8,10,11]. However, inherent challenges from LLMs, such as hallucination and lack of controllability [12,13], make deploying pure LLM-based systems difficult in practice. To incorporate LLMs' language capabilities into real systems,

there is a need to couple them with more manageable frameworks to ensure in-control task completion while achieving robust language interactions.

Developing a synchronous multi-party dialogue system is challenging, but evaluating such a system presents yet another hurdle. While a human-based evaluation offers the closest representation of real-life environments, recruiting a large group of users to thoroughly assess these dialogue systems over an extended period is time-consuming and resource-intensive. To address this, automated methods for simulating virtual users have been proposed; however, most of them simulate single users at a time, not simultaneous engagements from multiple users [14–17]. Thus, devising innovative solutions for conducting virtual simulations to assess such workload-intensive systems is imperative to develop dialogue systems that are well-prepared for real-life environments at minimal cost.

We present a dialogue system adept at conducting synchronous dialogues among multiple users for 17 administrative tasks (Section 3). In tandem, we introduce a multi-group simulator to facilitate the evaluation of our system by simulating users among multiple groups (Section 4). Our system is tested across various virtual environments produced by the simulator, showcasing high confidence in handling those tasks, and the results are promising for the effectiveness of our simulator and its resemblance to human interaction (Section 5). To our knowledge, this is the first system that comprehensively manages administrative tasks via synchronous multi-party dialogues and the first simulator accommodating users shared across multiple virtual groups.

2. Related Work

2.1. LLMs in Task-Oriented Dialogue Systems

With the emergence of LLMs for various natural language processing (NLP) tasks [18–20], their applications in task-oriented dialogue systems have grown, particularly for intent classification and dialogue state tracking. For intent classification, Zhang et al. [21] proposed a few-shot model using contrastive pre-training and fine-tuning, outperforming previous transformer-based models. Parikh et al. [22] adapted GPT for zero-shot prompting, showing competitive performance to few-shot prompting. For dialogue state tracking, Heck et al. [23] assessed GPT accomplishment of this task on two popular datasets [24,25], which showed state-of-the-art results. Feng et al. [26] developed a framework that leveraged open-source LLMs, yielding results comparable to those achieved with GPT.

While LLMs have been evaluated for those tasks, the research that has explored their sequential execution in dialogue systems is limited. Bang et al. [27] and Chung et al. [28] adapted GPT to build end-to-end task-oriented dialogue systems, surpassing prior systems in task execution. While such end-to-end systems offer many benefits, achieving fine-level controllability using these systems—a crucial aspect for real applications—proves challenging.

LLMs have also been individually employed and augmented to create systems capable of managing administrative tasks. For instance, frameworks presented by Gebreab et al. [4], Hays and White [5], and Musumeci et al. [6] illustrate how LLM augmentations are applied to administrative tasks across various domains, including cybersecurity and healthcare. These frameworks typically focus on information retrieval and extraction from user input [29]. Works such as Jung et al. [7] and Pan et al. [8] also developed task-oriented dialogue systems in which LLMs were used for knowledge-grounded augmentation. These systems outperformed most systems built on pre-trained models in knowledge-grounded administrative tasks, such as answering frequently asked questions [7]. However, most existing dialogue systems are not equipped to manage interpersonal administrative tasks, which require more intricate frameworks in conjunction with LLMs. Thus, we adopt a state machine-based framework to regulate dialogue flows, while utilizing an LLM for language understanding and generation.

2.2. Synchronous Multi-Party Chatbots

The domain of synchronous multi-party dialogue systems, especially those for task-oriented dialogues, is underexplored. Toxtli et al. [30] introduced the TaskBot, allowing users to request that other users be reminded to complete tasks. The TaskBot effectively facilitated the transfer of reminders between users, fostering asynchronous communication among them. Mendoza et al. [9] presented another chatbot for interactions between students and teachers (STBot), showcasing various features, including scheduling events for teachers, exchanging files among users, and addressing frequently asked questions.

While both chatbots exhibit multi-party dialogue capabilities, they rely on commercial frameworks; TaskBot operates in the Microsoft Bot Framework (Bot Framework: <https://dev.botframework.com>), and STBot is built on Google Dialogflow (Dialogflow: <https://cloud.google.com/dialogflow>). Hence, their functions are limited by the features offered in those frameworks, hindering the adaptation of LLMs. Our system stands out, as it enables real-time synchronous dialogues between the chatbot and users for general tasks without targeting specific groups, and it does so by fully leveraging LLMs.

2.3. Multi-Group Virtual Simulations

Due to the inefficiency in time and labor associated with assessing dialogue systems, several automated evaluation methods have been suggested, such as self-talk, where a chatbot engages in internal conversations by taking both the roles of the bot and users, or interactions between multiple different chatbots [31–34]. The advent of LLMs and their widespread application in diverse dialogue systems has further diversified evaluation approaches, integrating human, automated, and LLM-based assessments to evaluate generated responses [35,36]. Recently, Svikhnushina and Pu [37] introduced a dialogue evaluation pipeline employing an LLM to interact with chatbots on behalf of human users, and it showed promising results, although the LLM interaction was limited to one chatbot at a time.

LLMs have proven to be capable in replicating human-like behavior, and there have been prior works exploring the simulation of virtual users with various personalities using LLMs [38,39]. As a result of this capability, the use of LLMs for user simulation has been studied in task-oriented dialogue systems as well. Lu and Wang [17] developed a framework that generated virtual students using LLMs to evaluate the quality of multiple choice questions. Sekulic et al. [14], Wan et al. [15] employed LLMs fine-tuned on augmented datasets and compared them with existing models such as Llama [18] and GPT. Liu et al. [16] suggested applying multiple simulators, with each simulator behaving as a distinct user, instead of employing a single simulator to represent multiple users when training task-oriented dialogue systems. Despite the advancements in using LLMs to simulate virtual personalities, however, synchronous multi-party simulation is still underexplored, particularly in a multiple group setting where the same user can belong to several groups, as introduced in our work.

3. Assistant Bot

Our dialogue system, the Assistant Bot, operates on the Emora State Transition Dialogue Manager (STDM) framework, which offers versatile ways of defining dialogue states and utilizing third-party APIs for the development of an end-to-end dialogue system. Upon user input, STDM determines a relevant state by interpreting the input context, moves to another state based on this interpretation, and generates an output using a template [40]. While STDM supports a rapid method of building a dialogue system with great controllability, it has several limitations. First, STDM relies on its own formal grammar, NATEX (NATural language EXpression), for language understanding, which often falls short in accommodating the diverse expressions encountered during conversations. Second, each user's database is stored independently, thus, information extracted from one user cannot be shared across other users; although this adds an extra layer of security, it complicates the process of conducting synchronous multi-party dialogues. Third, it was

originally designed for passive interactions, meaning that a chatbot built on STDM waits for users to initiate conversations without proactively engaging with them.

Sections 3.1–3.3 address these three limitations in STDM to develop a more robust dialogue system. Moreover, Section 3.4 outlines the administrative tasks that the Assistant Bot can handle. Figure 1 showcases the framework and execution process of the Assistant Bot, and Figure 2 demonstrates an example of synchronous multi-party dialogue between two mock users, showcasing the functionalities of the Assistant Bot.

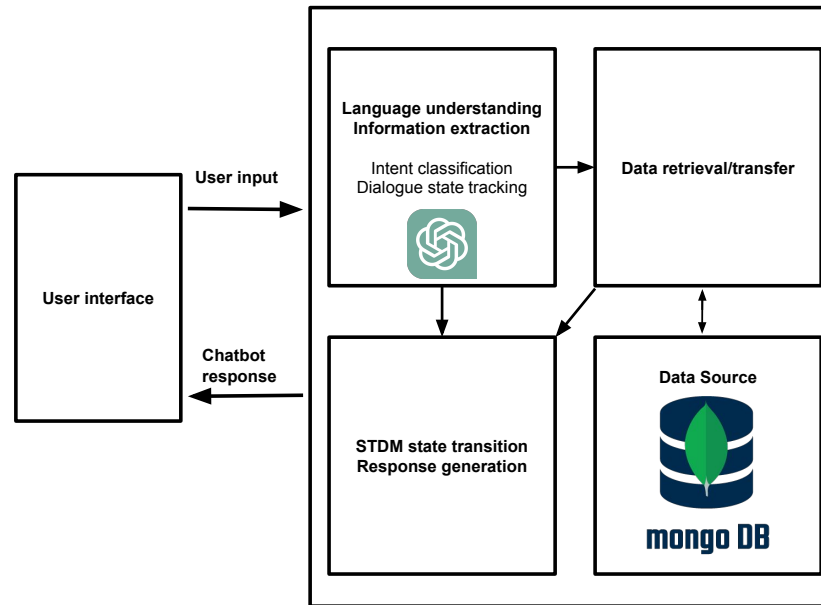


Figure 1. A diagram of the Assistant Bot’s system framework and execution process.

3.1. Language Understanding and Generation

To enhance natural language understanding (NLU), our system uses GPT-3.5 Turbo (henceforth, GPT), known for its exceptional performance, instead of NATEX [41–44].

For each user input, a few-shot GPT prompt template is employed, pairing a question prompt with the user’s input to extract the necessary information. GPT is directed to respond in a JSON schema, which is then integrated into the system’s framework as a set of variables that the Assistant Bot accesses for subsequent tasks. The few-shot GPT prompt template is as follows:

```
{full_request} Respond only in the JSON schema such as {examples}. Analyze this quote: "{quote_to_analyze}"
```

In this template, `full_request` corresponds to the specific request GPT must fulfill in order to extract the relevant information, `examples` refers to example JSON schemas that the LLM should use as a response format, and `quote_to_analyze` refers to the user’s input. For instance, if the user provides the name of the recipient with whom they wish to schedule a meeting, the variables are populated with information pertinent to the task. The final GPT prompt is as follows:

```
The speaker wants to schedule a meeting with another person. What is that other person’s name or ID? Ignore honorifics such as Dr, Mr, Mrs, Ms, or Mx. Respond only in the JSON schema such as {V.other_person_full_name.name: ‘Nora Allen’} or {V.other_person_full_name.name: ‘N/A’} if unavailable. Analyze this quote: "I need to meet with Dr. Smith"
```

— Gwen Stacy is now talking —

1: (Chatbot) Good evening, Gwen. How may I help you?

2: (U1) I'd like to make an appointment with Dr. Parker to discuss my career aspirations.

3: (Chatbot) Okay! Please tell me your choice for a meeting day and time with Dr. Peter Parker.

4: (U1) How about Thursday at 2:00 PM?

5: (Chatbot) Awesome, I'll check if he is available. Where do you want to meet with Dr. Parker?

6: (U1) In his office.

7: (Chatbot) Thank you! I'll send off your request:
Meeting Request from Gwen Stacy to Peter Parker: 03/21/2024 (2:00 PM - 2:30 PM) at Dr. Parker's office.
Discuss career aspirations.

8: (U1) Thank you for your assistance.

9: (Chatbot) Of course! It's my pleasure.

— Peter Parker is now talking —

10: (Chatbot) On 03/19/2024 (09:59 PM), Gwen Stacy proposed the following new event:
Meeting Request from Gwen Stacy to Peter Parker: ... (the same message as in L7)

11: (Chatbot) However, you already have a conflict scheduled during that time:
Meeting Request from MJ Watson to Peter Parker: 03/21/2024 (02:00 PM - 02:30 PM) at the Library.
Project brainstorming and seeking feedback on ideas.

12: (Chatbot) If the proposed times don't work for you, please suggest an alternate time for Gwen Stacy.

13: (U2) Ask Gwen if she can come to my office on Friday at 3:00 PM instead.

14: (Chatbot) Sure, I'll send off your rescheduling request:
Reschedule Request from Peter Parker to Gwen Stacy: 03/22/2024 (3:00 PM - 3:30 PM) at Dr. Parker's office
Discuss career aspirations.

15: (U2) Sounds good. Thank you.

16: (Chatbot) No problem. Have a great day!

— Gwen Stacy is now talking —

17: (U1) Any update from Dr. Parker?

18: (Chatbot) On 03/19/2024 (11:43 PM), Peter Parker proposed a changed event:
Reschedule Request from Peter Parker to Gwen Stacy: ... (the same message as in L14)

19: (Chatbot) Would that work for you?

20: (U1) Yes, that works for me. Thanks.

21: (Chatbot) Awesome, I'll let Peter Parker know!

Figure 2. An example of synchronous multi-party dialogue among two users, Gwen and Peter, scheduling a meeting assisted by our chatbot. Initially, the chatbot engages with Gwen, receiving her request intended for Peter (L1-9). Subsequently, it interacts with Peter to ascertain availability and receives a request for rescheduling (L10-16). Finally, the chatbot reconnects with Gwen to confirm the rescheduled appointment from Peter (L17-21).

Figure 3 illustrates how GPT is employed for NLU. For each user input, GPT infers the task (*Make Appointment*) and extracts relevant information (*Professor Smith*). If necessary, the extracted information is matched to database entries (*Elizabeth Smith*). When there is an ambiguity (e.g., multiple users with the last name *Smith* who are also professors), it enters an intermediate state for further clarification (e.g., *Which of these people did you mean?*). This information is then used to determine the next state, and an output is created by a template predefined for the state. Finally, the templated output is fed into GPT to diversify the final output. Should GPT fail to extract the necessary information, a combination of NATEX and regular expressions is employed as a backup to extract the information. Alternatively, the conversation flow is redirected to a state where the chatbot asks for clarification and reattempts to extract the required information.

1: (Chatbot) What do you need next, John?
 2: (User) I need to set up an appointment with Professor Smith.

[Natural Language Understanding]
 - Task: *Make Appointment*
 - Recipient: *Professor Smith* → *Elizabeth Smith*

3: (Chatbot) Okay! Please tell me your first choice for a meeting day and time with Elizabeth Smith.

Figure 3. A dialogue example demonstrating how our chatbot utilizes GPT for NLU.

3.2. Shared Database

For seamless information exchange and facilitation of synchronous communication and task execution, our system employs MongoDB, a NoSQL database system (MongoDB: <https://www.mongodb.com>). Each user record in the database contains a unique user ID, groups that the user belongs to, conversation history and relevant information interpreted by GPT (Section 3.1), and an inbox storing the user's appointments, messages, feedback questions, and responses to groups. Access to the records of other users is strictly restricted for any user. When a user requests a task from another user (e.g., making an appointment), the user indicates their preferences (e.g., date and location), stored in the user's record, and grants the system permission to share particular parts of the information with the other user. Only the permitted information is shared among the targeted users to accomplish the task, allowing our system to send requested messages, anonymously or with names attached, to individuals or specific groups.

3.3. Proactive Engagement

Many administrative tasks require the system to actively reach out to users in a timely manner, which is difficult to accomplish without a proper interface. For proactive engagement with our users, a mobile app is created using Flutter, an open-source framework for developing multi-platform applications (Flutter: <https://flutter.dev>). Figure 4 portrays the dashboard and chat interface of the Assistant Bot application. When a user requests a task, the app sends a push notification to the other user's application on their mobile device, facilitating engagement with our system among human users (Section 5) (our app is available on both iOS App Store and Google Play).

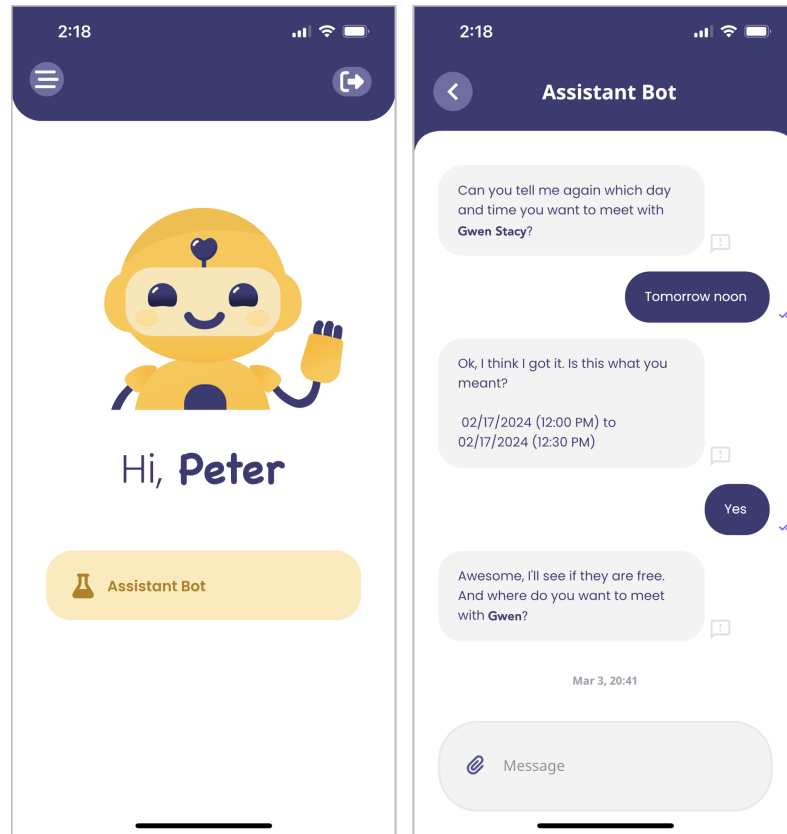


Figure 4. The dashboard (**left**) and chat interface (**right**) of our mobile app hosting the Assistant Bot.

3.4. Administrative Tasks

Our system manages four types of administrative tasks, as outlined in Table 1. Each type of task corresponds to an individual component within the Assistant Bot which the chatbot transitions to if the identified request corresponds to the task listed in Table 1.

Table 1. Description of tasks that Assistant Bot can execute.

| Type | Task | Description | Figures |
|------|------------------------|--|-----------|
| N&H | Set name | Establishes the names that the user wishes to be called by | Figure A1 |
| | List roster | Lists the users who are registered in the database | |
| APP | Make appointment | Sends an appointment request to a different user | Figure A2 |
| | Confirm appointment | Confirms an appointment request between two users and adds to schedules | |
| | Edit appointment | Edits an appointment request’s details | |
| | Reschedule appointment | Reschedules appointment request to different date | |
| | Cancel appointment | Cancels appointment request | |
| MSG | Send message | Sends message to recipient | Figure A3 |
| | Check inbox | Checks inbox for messages and appointment requests | Figure A4 |
| GRP | Make group | Makes group of multiple people | Figure A5 |
| | Delete group | Deletes a group that has been created (must be leader of the group to proceed) | Figure A6 |
| | Leave group | Leaves a group that user has been added to (must be a group member to proceed) | |
| | Check groups | Checks groups that users are in | |
| | Send announcement | Sends an identical message to group members (must be leader of the group to proceed) | Figure A7 |
| | Request feedback | Sends feedback questions to group members (must be leader of the group to proceed) | |
| | Send feedback | Records responses to feedback question (must be a group member to proceed) | Figure A8 |
| | Check feedback | Checks statistics of feedback question (must be leader of the group to proceed) | Figure A9 |

3.4.1. Names and Hub

Upon the initial interaction, the Assistant Bot requests the user’s official name and preferred names. While the official name is stored in the database and matched for task requests from other users, our system dynamically incorporates preferred names into

subsequent responses to enhance user engagement. Upon name confirmation, it moves to the hub state, where it allows the user to request any task, categorizes it into a task in Table 1, and directs the user to the corresponding state.

3.4.2. Appointments

The Assistant Bot streamlines appointment scheduling by managing details such as date, time, location, and purpose. Upon mutual confirmation, it updates users' schedules and resolves any scheduling conflicts. Requesting users can edit or cancel pending appointment requests, while both requesting and requested users can reschedule any appointments with passed or conflicting dates.

3.4.3. Messages

A user can dispatch messages to any other users in the database via the Assistant Bot. Users can send messages either anonymously or with their names; however, anonymity is maintained externally to ensure user safety.

3.4.4. Groups

The Assistant Bot allows users to create groups, with the creator becoming the leader and being able to add other users as members. Groups are identified by unique IDs, and users can check the groups they belong to at any time. Leaders oversee group activities, including creation, deletion, and communication through announcements. They can also request feedback from group members and review statistics on member responses. Members can respond to feedback requests from the leader and leave groups at their discretion; however, rejoining requires the leader's permission.

4. Multi-Group Virtual Simulator

It is challenging to coordinate with human users to assess the Assistant Bot's ability to handle multiple users in different groups, especially for researchers lacking established user bases. Thus, we present a multi-user simulator harnessing GPT's capabilities in persona impersonation and dialogue generation [45,46]. Each user is personalized with a distinct prompt and their past dialogue history with the chatbot. The rest of this section outlines our simulation process.

4.1. Preprocessing

Prior to simulation, personas for u users across g groups are first generated by GPT. Each persona contains the user's unique ID, name, occupation, personality, and associated class IDs. Leveraging GPT for persona generation enhances scalability, as increasing u and g requires no extra human effort with a small additional cost.

4.2. Initialization

Each virtual simulation starts with initialization, when a user greets the Assistant Bot. The Assistant Bot then asks the user for their name. Once our system receives the name, it transitions the user to the hub state, so the user can commence a simulated conversation. This process is applied to designated users until they are all initialized.

4.3. Interactions

During interactions, the simulator dynamically loads the user's dialogue history into a prompt template, which is then fed into GPT to generate an output tailored to the user's persona.

The simulator presents this output as a user turn and generates a subsequent system turn according to the corresponding state. The conversation history, including the last system output to that user, gets updated accordingly.

Upon each system turn, two additional functions are executed using GPT. The first monitors the ongoing conversation between the current user and the chatbot, checking for any pending notifications or tasks that have been resolved. The results, presented as

a dictionary containing notification and task counts for each user, are used to update the relevant users' notifications and tasks in the database. The second function ascertains whether or not the conversation is concluded. If not, it continues with the current user until it meets a natural conclusion.

A conversation concludes once all relevant states have been explored to address the requested task. When a conversation with one user concludes, the simulator selects a different user based on their task and notification status, as follows:

$$\operatorname{argmax}_{\forall i \in U'} \left(\frac{(\alpha \cdot n_i) + (\beta \cdot t_i)}{\sum_{\forall j \in U'} (\alpha \cdot n_j) + (\beta \cdot t_j)} \right)$$

n_i and t_i are the number of notifications and tasks of the i 'th user, respectively. U' is the set of all user numbers, excluding the one assigned to the current user. In our case, $\alpha = 2.5$ and $\beta = 1$ are used such that users with higher notification counts receive greater weight, simulating real-life behavior, where people tend to prioritize notifications. From users with the highest weights, the next user is selected via random sampling to continue the simulation.

The prompts for generating the simulated user's response, as well as those for updating notifications and tasks and checking if a conversation has concluded, are illustrated in Appendix A.1.1.

4.4. Termination

Each simulation is constrained by a maximum number of turns m , encompassing both system and user turns. The simulation terminates either upon reaching the turn limit or resolving all tasks and notifications for the designated users.

5. Experiments

5.1. Virtual Simulation

To assess the efficacy of our system, various college class scenarios are simulated. Given (u, g) , where u/g equates to the respective number of users/groups (Section 4), five rounds of simulation are conducted, each composing different combinations, as follows:

$$(u, g) \in \{(10, 1), (20, 1), (20, 2), (20, 3), (30, 1)\}$$

For instance, in the scenario $(20, 2)$, we simulate 20 students across 2 classes, ensuring every student takes at least one of those two classes. Additionally, one instructor is simulated for each class. Table A1 in Appendix A.1 describes the list of personas generated by GPT during the preprocessing stage (Section 4). Moreover, the maximum number of turns for termination in each round is set to $m = 500$ (Section 4).

5.2. Human Simulation

To compare the behaviors of the virtual classroom with those of humans, a group comprising one professor and 15 college and graduate students is recruited to simulate a classroom scenario, interacting with our Assistant Bot via the mobile app (Section 3.3). Each user is provided a persona description to role-play, in tandem with those assigned to the virtual users in the simulation (Table A1 in Appendix A.1). To ensure fair comparisons, the class details for both the virtual and human simulations are kept consistent. This simulation comprises a total of 1821 turns.

We acknowledge that the sample size in the experiments is relatively small, which may limit the generalizability of the results. Future work will involve conducting experiments with larger sample sizes to ensure broader generalizability and to further validate the effectiveness of the virtual simulation and Assistant Bot.

5.3. Evaluation Metrics

For each round of simulation, the Assistant Bot and the virtual simulator are evaluated and compared to the human simulation. The **task completion** of the Assistant Bot is tested by how successfully it handles each requested task using the metrics below:

- NLU: The failure rate due to NLU shortcomings (in %).
- STD: The failure rate due to inherent limitations in the STDM framework, hindering it from exiting the current dialogue state until it is fully resolved, even when the user requests another task (in %).
- SUC: The overall success rate; $100 - (NLU + STD)$.
- SWS: The success rate without accounting for STD, used to evaluate system performance excluding fails due to limitations of the STDM framework; $100 \times SUC / (100 - STD)$.

Moreover, each turn generated by the Assistant Bot is assessed for its robustness in **natural language understanding** using the following metrics:

- INT: The error rate resulting from inaccurate intent classification by GPT (in %).
- DST: The error rate resulting from inaccurate dialogue state tracking by GPT (in %).
- STD: The error rate due to the STDM limitations, leading to NLU failures (in %).
- SUC: The overall success rate; $100 - (INT + DST + STD)$.
- SWS: The success rate without considering STD; $100 \times SUC / (100 - STD)$.

Finally, the appropriateness rate (APP) of the **simulator** is assessed for each virtual user turn, which addresses various factors such as hallucinations by GPT, responses irrelevant to the Assistant Bot's last response, and repetitive statements disregarding the user's previous responses. If the repetition pertains to the requested task, it is still considered relevant.

5.4. Results

Table 2 presents the evaluation results for our Assistant Bot and Virtual Simulator. For task completion, performance is notably influenced by STDM; when discounting STDM errors, it achieves high success rates of 87.1% and 81.6% for the virtual and human simulations, respectively. Interestingly, comparisons of different class numbers, ($u = 20$, $g \in \{1, 2, 3\}$), show improved SWS with more classes; distributing a fixed number of students across a greater number of classes reduces the size of each class, which may lead to more successful task completion.

For natural language understanding, the average success rates rise by 19.6% and 22.7% for the virtual and human simulations, respectively, when discarding the STDM errors, suggesting that while STDM enables rapid prototyping of dialogue systems, a more robust framework is needed for real applications. Both SUC and SWS remain relatively stable across all scenarios, confirming the robustness of the NLU approach enabled by GPT in our system.

Table 2. Success and error rates for task completion, natural language understanding, and the virtual simulator across the 10 evaluation metrics (Section 5.3). Vir-Avg: the average rates across the five scenarios (Section 5.1). Hum-Avg: the average rates within the human simulation (Section 5.2). Bolded values represent the highest success rates.

| Scenario (u, g) | Task Completion | | | | Natural Language Understanding | | | | | Simulator |
|------------------------|-----------------|------|------|-------------|--------------------------------|-----|------|------|-------------|-------------|
| | NLU | STD | SUC | SWS | INT | DST | STD | SUC | SWS | APP |
| (10, 1) | 9.3 | 36.7 | 54.0 | 85.3 | 7.1 | 1.8 | 22.4 | 68.7 | 88.7 | 69.8 |
| (20, 1) | 7.8 | 34.6 | 57.6 | 87.8 | 4.6 | 3.0 | 23.9 | 68.5 | 90.0 | 81.8 |
| (20, 2) | 6.8 | 38.3 | 54.9 | 88.7 | 4.5 | 2.9 | 22.5 | 70.1 | 90.4 | 79.6 |
| (20, 3) | 6.2 | 37.7 | 56.1 | 90.0 | 3.6 | 5.3 | 20.9 | 70.2 | 88.7 | 83.6 |
| (30, 1) | 9.7 | 41.0 | 49.3 | 83.5 | 5.5 | 4.6 | 20.4 | 69.4 | 87.2 | 77.0 |
| Vir-Avg | 8.0 | 37.6 | 54.4 | 87.1 | 5.1 | 3.5 | 22.0 | 69.4 | 89.0 | 80.3 |
| Hum-Avg | 10.5 | 43.1 | 46.4 | 81.6 | 5.2 | 4.2 | 26.0 | 64.6 | 87.3 | - |

Comparing the virtual and human simulations reveals similar performance trends, underscoring the adaptability of our virtual simulation for evaluating synchronous dialogue system performance. Both simulations produce comparable numbers of tasks per turn; 44% and 40% of turns in the virtual and human simulations, respectively, are task requests. While the human simulation comprises a broader range of tasks (20) compared to the virtual one (14), the virtual simulation still elicits a sufficiently diverse task distribution, facilitating effective evaluation of the bot’s capabilities. It is important to note, however, that it is easy to design a virtual simulation that accommodates a dynamic range of tasks, which can result in a more varied task distribution, as users are completely controllable in this setting.

6. Discussion

6.1. Error Analysis on the Assistant Bot

For each erroneous turn of the Assistant Bot during the virtual simulation, the generated dialogues and preceding context are examined to understand its limitations. Table 3 highlights the common errors that resulted in incorrect responses from the Assistant Bot. Examples of these errors are in Appendix A.3.

Table 3. Analysis of errors in the Assistant Bot. STDM: errors caused by the STDM limitations; NLU-DST/IC: NLU errors caused by GPT’s misinterpretation of dialogue state tracking & intent classification, respectively.

| Category | Error Type | % |
|----------|----------------------------|------|
| STDM | State Lock | 0.68 |
| | NATEX Issues | 0.20 |
| | System Bugs | 0.11 |
| | Multiple Task Requests | 0.01 |
| NLU-DST | Multiple State Extraction | 0.84 |
| | Information Mis-extraction | 0.16 |
| NLU-IC | Context Mis-interpretation | 0.62 |
| | Task Mis-classification | 0.38 |

6.1.1. STDM Limitations

The primary source of the Assistant Bot’s errors stems from limitations in STDM, leading to various issues in dialogue flow. Of these limitations, *g* errors emerge as prominent issues, as the Assistant Bot cannot move on to a new task as a result of the STDM locking it into the current state until fully resolved, even when the user wishes to abandon the current task and transition to the new one (Figure A10). Although our system mainly uses GPT for NLU, it still defaults to NATEX as a fallback NLU method in cases where GPT fails due to internet connection or server issues. NATEX errors arise when NATEX takes over GPT for NLU with incorrect interpretation (Figure A11). Certain errors are induced by unexpected *system bugs* in STDM (Figure A12), while a minor portion of errors are attributed to STDM’s inability to handle *multiple task requests* in a single turn, in which cases, STDM processes one of the requests and disregards the others (Figure A13).

6.1.2. NLU Errors in Dialogue State Tracking (DST)

Most of the NLU errors occur due to GPT shortcomings in dialogue state tracking (DST) and intent classification (IC) tasks. The primary cause of DST errors is GPT’s limited ability to robustly extract *multiple dialogue states* at once, as the GPT prompt becomes too complicated when many states are included (Figure A14). Additionally, GPT variability and inconsistency sometimes result in *incorrect information extraction*.

6.1.3. NLU Errors in Intent Classification (IC)

For IC, the main source of errors stems from context management issues, where user inputs are misinterpreted as certain tasks. Due to overlap among tasks within the same type, such as making and canceling appointments, context ambiguity can result in incorrect intent classification (Figure A15). In addition, GPT may wrongly classify the user's intended task, causing the Assistant Bot to enter an incorrect state.

6.2. Error Analysis on the Virtual Simulation

Every erroneous turn from the virtual simulation is analyzed to assess its effectiveness in handling synchronous multi-party dialogues. Table 4 shows the causes of these inappropriate responses. The most frequent errors occur when the virtual simulation *ignores contexts* from the Assistant Bot's previous response (Figure A16). While these errors disrupt dialogue flow, it is important to note that they are **not** unlike human behavior, as humans often respond in ways that overlook the chatbot's previous response.

Table 4. Error analysis on the virtual simulation.

| Error Type | % |
|----------------------------------|------|
| Ignoring Context | 0.39 |
| Forgetting Conversation History | 0.30 |
| Forgetting User Information | 0.19 |
| Confusing Conversational Partner | 0.06 |
| Role Reverse | 0.06 |

Another primary cause stems from the LLM's inability to focus on the provided information to accurately simulate the user. The simulated user frequently *forgets parts of the conversation history*, leading to requests for tasks that have already been completed (Figure A17). Furthermore, it fails to recall the provided information, generating irrelevant or incorrect content in relation to the given context.

The third type of error occurs when the simulation *forgets information about the user*, resulting in content that is inconsistent with the simulated user's information (Figure A18). A small portion of errors occurs when the simulation becomes *confused about whom it is conversing with*, leading to responses that address the Assistant Bot incorrectly. This often happens when the bot delivers a message from another (simulated) user, causing the simulation to address the user rather than the Assistant Bot (Figure A19). Finally, *role reversal* occurs when the simulation fails to act as the user, instead generating responses that resemble the characteristics of the Assistant Bot (Figure A20).

6.3. Limitations

The limitations encountered by the Assistant Bot are largely rooted in the inherent constraints of the STDM framework. One significant issue is the use of NATEX. While NATEX is effective for the rapid prototyping of conversation transitions, it falls short in terms of natural language understanding due to its manually constructed nature. Although NATEX can be expanded to handle a variety of user inputs, creating a comprehensive dictionary that captures all possible transitions is impractical. This challenge, combined with the inherent ambiguity of natural language, often leads to categorization and transition errors. Furthermore, state lock errors prevent the Assistant Bot from smoothly handling multiple tasks. This is particularly problematic when users request multiple tasks within the same utterance or when they wish to switch tasks mid-conversation.

Given these limitations, exploring an alternative framework to replace the STDM could provide significant improvements. Since the STDM is robust in generating relevant natural language, a framework that can replicate the STDM's robust natural language generation while also dynamically navigating between different task-oriented conversation states is crucial for enhancing our chatbot's performance.

Additionally, the limitations of GPT contribute to errors in both the Assistant Bot and the virtual simulation. For the Assistant Bot, GPT's variability and its limited understanding of administrative task categorization can lead to incorrect information extraction, resulting in errors in dialogue state tracking and intent classification. For the virtual simulation, maintaining in-character responses that aligned with user profiles and conversation history proved challenging, especially when dealing with extensive context. Fine-tuning user prompts or the model specifically for simulation tasks could improve accuracy in these areas. Furthermore, as GPT-3.5 is currently used for both the Assistant Bot and the virtual simulation, utilizing more powerful models such as GPT-4 could enhance the performance of both the simulation responses and the Assistant Bot's natural language understanding.

7. Conclusions

While addressing the complexities of synchronous multi-party dialogue and multi-group virtual simulation presents challenges, this work is the first to demonstrate promising potential of a task-oriented dialogue system in simultaneously handling multiple users and leveraging large language models (LLMs) to supplement human evaluation. Despite limitations in the STDM framework and variabilities in GPT, our system exhibited robust natural language understanding and task execution, proving its practicality as a real-world application. Our virtual simulation also demonstrated a diversity in task requests similar to human interactions, underscoring its reliability as an evaluative tool to overcome the challenge of recruiting human groups.

However, the limitations of the STDM framework, coupled with the inherent variability of GPT models, suggests several directions for future research. Future work should focus on refining context management to further improve simulation performance and reliability. Enhancing the Assistant Bot's ability to understand situational nuances could also strengthen its natural language comprehension. Exploring alternative frameworks that offer stronger language generation capabilities, while addressing the constraints of the current STDM approach, could further advance the system's effectiveness. Additionally, applying the virtual simulation methodology to a wider range of group settings beyond the context of college classrooms could help generalize its applicability and adaptability.

While the Assistant Bot and virtual simulation help make significant advancements in the automation of administrative tasks and multi-user dialogue management using LLMs, continued research and refinement are necessary to fully realize the potential of these innovative approaches in diverse and dynamic environments. We look forward to advancements in task-oriented dialogue systems that employ synchronous multi-user features, as well as in evaluation frameworks that leverage the robust capabilities of large language models.

Author Contributions: Conceptualization, J.D.C.; methodology, J.D.C. and J.D.F.; software, E.S.P., T.F., and J.D.C.; validation, E.S.P., T.F., J.D.F., and J.D.C.; formal analysis, E.S.P.; investigation, E.S.P., J.D.C., and J.D.F.; resources, J.D.C.; data curation, E.S.P. and T.F.; writing—original draft preparation, E.S.P. and J.D.C.; writing—review and editing, E.S.P. and J.D.C.; visualization, E.S.P. and J.D.C.; supervision, J.D.C. and J.D.F.; project administration, J.D.C.; funding acquisition, J.D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Emory University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: We gratefully acknowledge the support of Emory University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Emory University.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Virtual Simulation Prompts and Personas

Table A1. Personas given to the virtual users in the simulation. The first group of rows lists the instructors of each course (course written in **Class**), and the second group of rows lists the students. CS: Introduction to Computer Science, PSY: Introduction to Psychology, HIS: United States History: From Colonial Period to Present.

| Name | Class | Personality | Tendencies |
|--------------------|--------------|---------------|---|
| Joseph Cho | CS | enthusiastic | organized and plans material meticulously |
| Amanda Johnson | PSY | approachable | interactive and encourages participation |
| Michael Brown | HIS | analytical | enjoys solving complex problems, often takes a methodical approach |
| Olivia Adams | CS | creative | approaches problems with unconventional thinking |
| Leila Ahmed | CS | creative | enjoys experimenting, thinks outside the box |
| Mohammed Ali | CS, PSY | analytical | enjoys solving challenging problems |
| Michael Brown | CS | analytical | enjoys solving complex problems, often takes a methodical approach |
| Valerie Chen | CS, HIS | enthusiastic | organized and plans study schedules meticulously |
| Diego Fernandez | CS | outgoing | prefers group study sessions, actively participates in class discussions |
| Amelia Garcia | CS, PSY, HIS | independent | prefers to work alone, excels in self-paced learning |
| Rohan Gupta | CS, PSY, HIS | analytical | approaches problems logically |
| Mary Johnson | CS | introverted | enjoys individual projects, excels in structured assignments |
| Fatima Khan | CS, PSY, HIS | creative | applies creative problem-solving techniques |
| Daniel Lee | CS | team-oriented | works well in groups, enjoys collaborative projects |
| Maria Lopez | CS, PSY, HIS | curious | likes exploring new concepts, often conducts independent research |
| Rachel Martinez | CS | outgoing | works best in groups, enjoys discussing ideas with classmates |
| Ethan Miller | CS | diligent | thrives in structured learning environments, maintains organized notes |
| Yuki Nakamura | CS | focused | attentive in lectures |
| Sophia Nguyen | CS | curious | explores additional resources beyond lecture material |
| Aisha Patel | CS, PSY | reserved | prefers self-study, takes detailed notes during lectures |
| Aaliyah Patel | CS, HIS | friendly | enjoys group discussions |
| Nadia Ramirez | CS, PSY, HIS | organized | creates detailed study guides, follows a structured study routine |
| Emily Rodriguez | CS | friendly | prefers group study sessions, good at explaining concepts to peers |
| Isabella Rodriguez | CS, HIS | enthusiastic | eager to learn, actively participates in class discussions |
| Elijah Scott | CS | persistent | consistently follows through on projects, enjoys mastering difficult concepts |
| Alexandre Silva | CS | introspective | prefers self-paced learning |
| Kumar Singh | CS, HIS | enthusiastic | eager learner, actively participates in group discussions |
| Ben Smith | CS, PSY | easygoing | prefers studying in a group, collaborates on group projects |
| Haruto Suzuki | CS, HIS | reserved | prefers studying alone, methodical in approach |
| Sato Tanaka | CS, PSY | friendly | prefers collaborative learning |
| Ryne Waters | CS | analytical | works well under pressure, completes assignments last minute |
| Chloe White | CS | curious | enjoys exploring new topics |
| James Wilson | CS | adventurous | enjoys hands-on projects and experimentation |

Appendix A.1.1. Student and Professor Response Generation Prompts

The prompt used for generating student simulation responses is as follows:

You are {name}, a student in the {sim_class} class. You are {personality}, with studying tendencies of {studying_tendencies}. Online, you tend to {online_tendencies}. Your professor is {professor}, and your classmates are {classmates}. The class history so far is as such: {class_history}. This week, the material covered was {current_material}.

You have {num_notifs} notifications you MUST check through the bot first. After the bot APPROPRIATELY RESPONDS, you want to complete this task: {todo}.

You are talking to an assistant chatbot. Its functionalities are ONLY as follows:

- makes appointments with other students or professor
- sends messages to people (anonymized or not anonymized) to other students or professor
- create and delete groups/classes with leaders and members
- leave a group if you are a MEMBER and in a GROUP
- check what groups you are in

- make announcements within groups if you are a LEADER and in a GROUP

You will be given a conversation history and the assistant chatbot's last response. Using the information and the given identity and personality, respond IN CHARACTER and APPROPRIATELY to the assistant chatbot's last response. Pay attention to what the bot is asking for and carefully consider your own tasks that you want to get done. Respond as if YOU are the one typing and asking for assistance; do NOT add role-playing details such as "John Doe: hi" or "user: hi" or "how can I assist you today?". In addition, do NOT REPEAT yourself, such as repeating "Thank you" or "add all students". If the conversation has ended, consider asking for any updates if you received notifications. If it is not, respond with something DIFFERENT. If the bot is repeating itself, CANCEL the current functionality the bot is in to return to the hub state.

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

user: hi

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Before we begin, please provide your full name.

CORRECT RESPONSE:

John Smith

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

user: hi

bot: Before we begin, please provide your full name.

user: Gwen Stacy

bot: Thank you Gwen Stacy. Welcome! How should I call you?

user: Gwen

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Nice to meet you, Gwen. Anything I can do for you, Gwen?

CORRECT RESPONSE:

any updates?

ANOTHER CORRECT RESPONSE:

actually call me Stacy

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

user: Can you help me schedule a study group session with my classmates for this week?

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Sure, I can help you schedule an appointment with someone. What is their full name?

CORRECT RESPONSE:

let's see if David is free

ANOTHER CORRECT RESPONSE:

Never mind

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

user: Call me Jasmine. Any updates?

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Nice to meet you, Jasmine. So, how's it going, Jasmine?

CORRECT RESPONSE:

so any mail

ANOTHER CORRECT RESPONSE:

I want to send a message to Dr Yin about a bug in the hw

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

user: You can call me Fatima. Thank you.

bot: Nice to meet you, Fatima. So what's up, Fatima?

user: I wanted to ask if there's a way for me to find a classmate to work together on the Data Structures homework?

bot: You're not in any groups right now, or you do not have any groups that you can request feedback from!!

user: Actually, I wanted to ask if there's a way for me to find a classmate to work together on the Data Structures homework. Can you help me with that?

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Do you also still want help with check the members in a group?

CORRECT RESPONSE:

no, it's ok

AN EXAMPLE of how it should be responded is provided below:

bot: Understood, I'll move on. Anything I can do for you, Joseph?

user: Any updates?

bot: There are no new messages in your inbox. Anything I can do for you, Joseph?

user: No, I'm all set for now. Thank you.

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: You have 1 message in your appointment inbox. Would you like to respond to it now?

CORRECT RESPONSE:

yes

History:

{history}

Using the conversation history provided above and the character's personality and class details as context, respond IN CHARACTER TO the chatbot's last response below:

Chatbot's Last Response:

{bot_last_response}

The prompt used for generating professor simulation responses is as follows:

You are {name}, a professor teaching the {sim_class} class. You are {personality}, with studying tendencies of {teaching_tendencies}. Online, you tend to {online_tendencies}. Your students are {students}. The class history so far is as such: {class_history}. This week, the material covered was {current_material}.

You have {num_notifs} notifications you MUST check through the bot first. After the bot APPROPRIATELY RESPONDS, you want to complete this task: {todo}.

You are talking to an assistant chatbot. Its functionalities are ONLY as follows:

- makes appointments with other students or professor
- sends messages to people (anonymized or not anonymized) to other students or professor

- create and delete groups/classes with leaders and members
- leave a group if you are a MEMBER and in a GROUP
- check what groups you are in
- make announcements within groups if you are a LEADER and in a GROUP

You will be given a conversation history and the assistant chatbot's last response. Using the information and the given identity and personality, respond IN CHARACTER and APPROPRIATELY to the assistant chatbot's last response. Pay attention to what the bot is asking for and carefully consider your own tasks that you want to get done. Respond as if YOU are the one typing and asking for assistance; do NOT add role-playing details such as "John Doe: hi" or "user: hi" or "how can I assist you today?". In addition, do NOT REPEAT yourself, such as repeating "Thank you" or "add all students". If the conversation has ended, consider asking for any updates if you received notifications. If it is not, respond with something DIFFERENT. If the bot is repeating itself, CANCEL the current functionality the bot is in to return to the hub state.

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

user: hi

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Before we begin, please provide your full name.

CORRECT RESPONSE:

John Smith

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

user: hi

bot: Before we begin, please provide your full name.

user: Gwen Stacy

bot: Thank you Gwen Stacy. Welcome! How should I call you?

user: Gwen

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Nice to meet you, Gwen. Anything I can do for you, Gwen?

CORRECT RESPONSE:

any updates?

ANOTHER CORRECT RESPONSE:

actually call me Stacy

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

user: Can you help me schedule a study group session with my classmates for this week?

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Sure, I can help you schedule an appointment with someone. What is their full name?

CORRECT RESPONSE:

let's see if David is free

ANOTHER CORRECT RESPONSE:

Never mind

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

user: Call me Jasmine. Any updates?

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Nice to meet you, Jasmine. So, how's it going, Jasmine?

CORRECT RESPONSE:

so any mail

ANOTHER CORRECT RESPONSE:

I want to send a message to Dr Yin about a bug in the hw

AN EXAMPLE of how it should be responded is provided below:

user_prompt =

user: You can call me Fatima. Thank you.

bot: Nice to meet you, Fatima. So what's up, Fatima?

user: I wanted to ask if there's a way for me to find a classmate to work together on the Data Structures homework?

bot: You're not in any groups right now, or you do not have any groups that you can request feedback from!!

user: Actually, I wanted to ask if there's a way for me to find a classmate to work together on the Data Structures homework. Can you help me with that?

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: Do you also still want help with check the members in a group?

CORRECT RESPONSE:

no, it's ok

AN EXAMPLE of how it should be responded is provided below:

bot: Understood, I'll move on. Anything I can do for you, Joseph?

user: Any updates?

bot: There are no new messages in your inbox. Anything I can do for you, Joseph?

user: No, I'm all set for now. Thank you.

Using the conversation history provided above as context, respond in character to the chatbot's last response below:

bot: You have 1 message in your appointment inbox. Would you like to respond to it now?

CORRECT RESPONSE:

yes

History:

{history}

Using the conversation history provided above and the character's personality and class details as context, respond IN CHARACTER TO the chatbot's last response below:

Chatbot's Last Response:

{bot_last_response}

For both the student and professor prompts, the brackets {} are replaced with the information provided in the selected user profile. Each label corresponds to the following information:

- name: Name of user
- sim_class: Classes that the user is part of
- personality: Personality of the user
- studying_tendencies: Studying tendencies of the user
- online_tendencies: Online tendencies of the user

- `professor`: If the user is a student, professors that user has for classes
- `classmates`: If the user is a student, classmates that user has in classes
- `students`: If the user is a professor, students that user has in class
- `class_history`: History of what material was taught and what the user did in the past weeks of classes
- `current_material`: What material is being taught at the point in time in which the user is conversing with the Assistant Bot
- `num_notifs`: Number of notifications the user has
- `todo`: Task(s) that user seeks to complete through Assistant Bot
- `history`: Chat history with Assistant Bot.
- `bot_last_response`: Assistant Bot's last response

Appendix A.1.2. Tasks and Notifications Prompt

The prompt used for determining the number of tasks and notifications for a given user is as follows:

You will be given a conversation history between a user and an assistant chatbot. Using it and the information provided, determine if there are any notifications are NOT resolved and tasks that are NOT completed. A notification is ONLY resolved when the user has completely received the message that the bot is delivering. A task is ONLY completed when the bot has completed the task the user requests and concludes as such. Respond ONLY in a dictionary format such as `{'notifications': 1, 'tasks': 1}`, or `'N/A'` if unable to be determined.

An EXAMPLE of how it should be responded is provided below:

`user_prompt =`

This user has received 0 notification and their task is as follows: send a message to Gwen about studying together. The number of notifications and tasks is presented in a dictionary format below:

`{'notifications': 0, 'tasks': 1}`

History:

— smith is now talking —

user: hi

bot: Before we begin, please provide your full name.

user: Gabriel Smith

bot: Thank you Gabriel Smith. Welcome! How should I call you?

user: How about Darth Vader

bot: Nice to meet you, Darth Vader. How can I assist you, Darth Vader?

Based on the history above, how many remaining tasks and notifications does this user have?

CORRECT RESPONSE:

`{'notifications': 0, 'tasks': 1}`

An EXAMPLE of how it should be responded is provided below:

`user_prompt =`

This user has received 2 notifications and their task is as follows: send a message to Ben about studying together. The number of notifications and tasks is presented in a dictionary format below:

`{'notifications': 2, 'tasks': 1}`

History:

— fan is now talking —

user: You can call me Gwen.

bot: Nice to meet you, Gwen. What do you need next, Gwen?
 user: I need to send a message to Ben about studying together.
 bot: Okay! Would you like to send this message anonymously?
 user: No, I don't need to send the message anonymously.
 bot: Alright! Can you please tell me the message that you want to send to Ben Smith?
 user: Sure! Here's the message I want to send to Ben: "Hey Ben, I was wondering if you'd be interested in studying together for our Intro to CS class. I find that collaborating with someone helps me understand the material better. Let me know if you're available and we can set up a study session. Thanks!"
 bot: Thank you! I'll send off your message.

Based on the history above, how many remaining tasks and notifications does this user have?

CORRECT RESPONSE:

{'notifications': 2, 'tasks': 0}

An EXAMPLE of how it should be responded is provided below:

user_prompt =

This user has received 2 notifications and their task is as follows: send a message to Dr. Cho about bug. The number of notifications and tasks is presented in a dictionary format below:

{'notifications': 2, 'tasks': 1}

History:

— ali is now talking —

user: I need to send a message to Dr. Cho.

bot: Okay! Would you like to send this message anonymously?

user: No, I don't need to send the message anonymously.

bot: Alright! Can you please tell me the message that you want to send to Ben Smith?

user: Subject: Possible Bug in Homework Assignment

Hi Dr. Cho,

I hope you're doing well. I wanted to bring to your attention a possible bug that I've noticed in the current homework assignment. After reviewing the requirements, I believe there may be an issue with the expected output for a specific problem. I have detailed my findings and suggestions for resolving this in the document attached to this message. Thank you for your attention to this matter.

Best regards,

Muhammad Ali

bot: Welcome! How should I call you?

Based on the history above, how many remaining tasks and notifications does this user have?

CORRECT RESPONSE:

{'notifications': 2, 'tasks': 1}

An EXAMPLE of how it should be responded is provided below:

user_prompt =

This user has received 1 notifications and their task is as follows: create a group for school project with Daniel and Han. The number of notifications and tasks is presented in a dictionary format below:

{'notifications': 1, 'tasks': 1}

History:

— tenali is now talking —

user: any updates?

bot: You have 1 message in your appointment inbox. Would you like to respond to it now?

user: yes

Based on the history above, how many remaining tasks and notifications does this user have?

CORRECT RESPONSE:

```
{'notifications': 1, 'tasks': 1}
```

An EXAMPLE of how it should be responded is provided below:

user_prompt =

This user has received 0 notifications and their task is as follows: make an appointment with Professor Johnson. The number of notifications and tasks is presented in a dictionary format below:

```
{'notifications': 0, 'tasks': 1}
```

History:

— patel is now talking —

user: You can call me Aisha.

bot: Nice to meet you, Aisha. What can I do for you, Aisha?

user: I would like to make an appointment with Professor Johnson for help with hash tables.

bot: Okay! Please tell me your first choice for a meeting day and time with Amanda Johnson.

user: I would prefer to meet with Professor Johnson on Thursday afternoon around 3:00 PM.

bot: Ok, I think I got it. Is this what you meant?

03/07/2024 (03:00 PM) to 03/07/2024 (03:30 PM)

user: Yes, that's correct. Thank you for scheduling the appointment with Professor Johnson on Thursday afternoon from 3:00 PM to 3:30 PM.

bot: Awesome, I'll see if they are free. Furthermore, where do you want to meet with Amanda Johnson?

Based on the history above, how many remaining tasks and notifications does this user have?

CORRECT RESPONSE:

```
{'notifications': 0, 'tasks': 1}
```

This user has received {notifs} notifications and their task is as follows: {todo}. The number of notifications and tasks is presented in a dictionary format below:

```
{user_notif_data}
```

History:

```
{history}
```

Based on the history above, how many remaining tasks and notifications does this user have?

Each label corresponds to the following information from the selected user profile:

- **notifs:** Number of notifications the user has
- **todo:** Task(s) that user seeks to complete through Assistant Bot
- **user_notif_data:** The user's number of notifications and tasks to complete, in dictionary format. {'notifications': the number of notifications, 'tasks': the number of tasks}
- **history:** Chat history with Assistant Bot.

Appendix A.1.3. Termination Prompt

The prompt used for determining whether a conversation between the user and system has ended or is repeating is as follows:

You will be given a conversation history between a user and an assistant chatbot. Using it, determine if the conversation has ENDED or is REPEATING between the bot and user. Your output should be EITHER YES or NO.

An EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

— smith is now talking —

user: hi

bot: Before we begin, please provide your full name.

user: Gabriel Smith

bot: Thank you Gabriel Smith. Welcome! How should I call you?

user: How about Darth Vader

bot: Nice to meet you, Darth Vader. How can I assist you, Darth Vader?

Based on the history above, is the conversation between the user and the bot ENDED or REPEATING?

CORRECT RESPONSE:

NO

An EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

— stacy is now talking —

user: You can call me Gwen.

bot: Nice to meet you, Gwen. What do you need next, Gwen?

user: I need to send a message to Ben about studying together.

bot: Okay! Would you like to send this message anonymously?

user: No, I don't need to send the message anonymously.

bot: Alright! Can you please tell me the message that you want to send to Ben Smith?

user: Sure! Here's the message I want to send to Ben: "Hey Ben, I was wondering if you'd be interested in studying together for our Intro to CS class. I find that collaborating with someone helps me understand the material better. Let me know if you're available and we can set up a study session. Thanks!"

bot: Thank you! I'll send off your message.

user: Great, thank you! I appreciate your help. I'll let you know if there's anything else I need help with. Have a great day!

bot: Of course! It's my pleasure.

Based on the history above, is the conversation between the user and the bot ENDED or REPEATING?

CORRECT RESPONSE:

YES

An EXAMPLE of how it should be responded is provided below:

user_prompt =

History:

— patel is now talking —

user: You can call me Aisha.

bot: Nice to meet you, Aisha. What can I do for you, Aisha?

user: I would like to make an appointment with Professor Johnson for help with hash tables.

bot: Okay! Please tell me your first choice for a meeting day and time with Amanda Johnson.

user: I would prefer to meet with Professor Johnson on Thursday afternoon around 3:00 PM.

bot: Ok, I think I got it. Is this what you meant?

03/07/2024 (03:00 PM) to 03/07/2024 (03:30 PM)

user: Yes, that's correct. Thank you for scheduling the appointment with Professor Johnson on Thursday afternoon from 3:00 PM to 3:30 PM.

bot: Awesome, I'll see if they are free. Furthermore, where do you want to meet with Amanda Johnson?

Based on the history above, is the conversation between the user and the bot ENDED or REPEATING?

CORRECT RESPONSE:

NO

History:

{history}

Based on the history above, is the conversation between the user and the bot ENDED or REPEATING?

Each label corresponds to the following information from the selected user profile:

- `history`: Chat history with Assistant Bot.

Appendix A.2. Conversation Examples Demonstrating Assistant Bot Tasks

1: (User) hi

2: (Chatbot) Before we begin, please provide your full name.

3: (User) Ryne Waters

[Natural Language Understanding]
- Full name: *Ryne Waters*

4: (Chatbot) Thank you Ryne Waters. Welcome! How should I call you?

5: (User) just ryne is fine

[Natural Language Understanding]
- Names to call: *[Ryne]*

6: (Chatbot) Nice to meet you, Ryne. What do you need next, Ryne?

Figure A1. An example of an initial conversation between the Assistant Bot and a mock user with the name calling function.

1: (User) Great, I'd like to make an appointment with Prof Smith

[Natural Language Understanding]
 [Natural Language Understanding]
 - Task: *Make Appointment*
 - Recipient: *Prof Smith → John Smith*

2: (Chatbot) Okay! Please tell me your first choice for a meeting day and time with John Smith.

3: (User) Tomorrow at 5:20

[Natural Language Understanding]
 - Appointment Date: *10/05/2023 05:20 PM to 10/05/2023 05:50 PM*

4: (Chatbot) Ok, I think I got it. Is this what you meant?

10/05/2023 (05:20 PM) to 10/05/2023 (05:50 PM)

5: (User) yup

[Natural Language Understanding]
 - Affirmation (Yes/No): *Yes*

6: (Chatbot) Awesome, I'll see if they are free. And where do you want to meet with John Smith?

7: (User) In his office

[Natural Language Understanding]
 - Appointment Location: *his office*

8: (Chatbot) Ok, I'll mark the location as his office. Would you like to include another alternative time to meet?

9: (User) no

[Natural Language Understanding]
 - Affirmation (Yes/No): *No*

10: (Chatbot) Understood, that's it. Please tell me your reason for wanting to meet with John Smith.

11: (User) Discussing thesis things

[Natural Language Understanding]
 - Appointment Reason: *Discussing thesis*

12: (Chatbot) Thank you! I'll send off your request:

Meeting, Haruto Suzuki and John Smith: 10/05/2023 (05:20 PM) to 10/05/2023 (05:50 PM) at his office. (Discussing thesis)

Figure A2. An example of the Assistant Bot handling an appointment creation request.

1: (User) Can I send a message to Gaia?

[Natural Language Understanding]
 - Task: *Send Message*
 - Recipient: *Gaia → Gaia*

2: (Chatbot) Okay! Would you like to send this message anonymously?

3: (User) No! It's okay

[Natural Language Understanding]
 - Affirmation (Yes/No): *No*
 - Anonymous Message: *No*

4: (Chatbot) Alright! Can you please tell me the message that you want to send to Gaia?

5: (User) Hey Gaia! I just wanted to ask if you still have the textbook that I lend you.

[Natural Language Understanding]
 - Message Contents: *Hey Gaia! I just wanted to ask if you still have the textbook that I lend you.*

6: (Chatbot) Thank you! I'll send off your message.

Figure A3. An example of the Assistant Bot handling a message sending request.

1: (User) so I got notifis, show me what you got

[Natural Language Understanding]
 - Task: *Check Inbox*

2: (Chatbot) You have 2 messages in your inbox. Would you like to read them now?

3: (User) sure

[Natural Language Understanding]
 - Affirmation (Yes/No): *Yes*

4: (Chatbot) On 03/24/2024 (08:32 PM), Ryne Waters wanted me to relay you this message:
 Hey Gaia! I just wanted to ask if you still have the textbook that I lend you.
 In summary: Requesting return of lent textbook

5: (User) oh shoot ok, I forgot

[Natural Language Understanding]
 None

6: (Chatbot) I don't have any messages for you right now. Let me check for any appointments.

7: (User) cool

[Natural Language Understanding]
 None

8: (Chatbot) On 03/24/2024 (10:30 AM), Mitron proposed the following new event:
 Meeting, Mitron and Gaia: 03/25/2024 (03:00 PM) to 03/25/2024 (03:30 PM) at online. (Discussing group project)

Does that work for you?

Figure A4. An example of the Assistant Bot handling an inbox check.

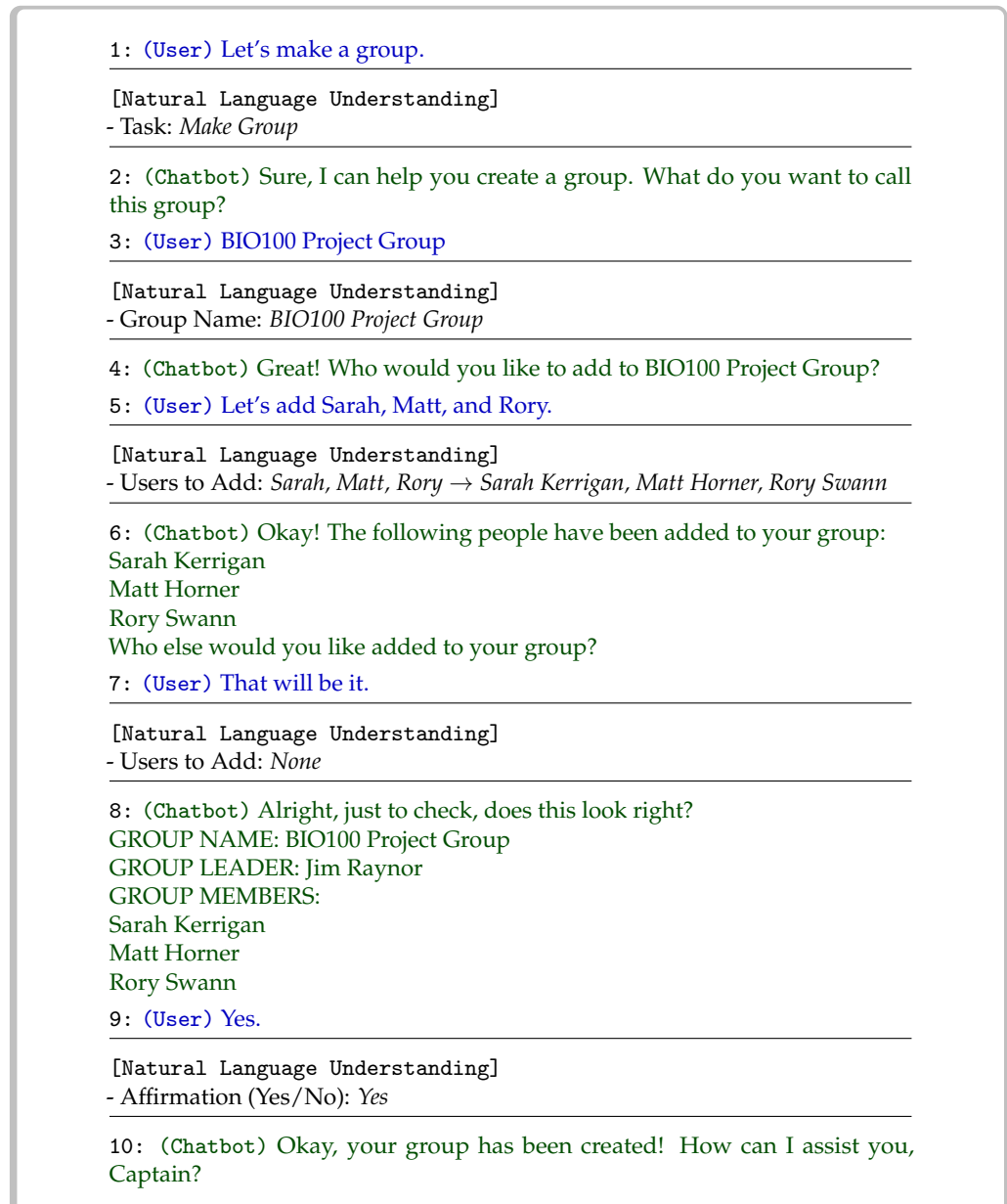


Figure A5. An example of the Assistant Bot handling a group creation request.

1: (User) what groups am I in

[Natural Language Understanding]
- Task: *Check Groups*

2: (Chatbot) Here is a list of all of the groups that you're in!

CS Study Buddies (Leader: Sarah Kerrigan)
BIO100 Project Group (Leader: Jim Raynor)

3: (User) ok

[Natural Language Understanding]
None

4: (Chatbot) What's on your mind, Sarah?

5: (User) who's in CS Study Buddies

[Natural Language Understanding]
- Task: *Check Groups*
- Group: *CS Study Buddies*

6: (Chatbot) Here is the current status of CS Study Buddies!
Name: CS Study Buddies
Leader: Sarah Kerrigan
Members:
Zagara
Niadra
Abathur

Figure A6. An example of the Assistant Bot handling a group identification request.

1: (User) I'd like to request feedback from my LING 200 class

[Natural Language Understanding]
- Task: *Request Feedback*
- Group: *LING 200* → *LING200*

2: (Chatbot) Alright! Can you please tell me the feedback question that you want to send to LING200?

3: (User) I'd like to request feedback for the midterm that everyone took. Was it too easy, difficult, or just right? If there're any questions that you felt were confusing or difficult, let me know.

[Natural Language Understanding]
- Feedback Question: *I'd like to request feedback for the midterm that everyone took. Was it too easy, difficult, or just right? If there're any questions that you felt were confusing or difficult, let me know.*

4: (Chatbot) Thank you! I'll send off your message.

Figure A7. An example of the Assistant Bot sending feedback requests to groups.



Figure A8. An example of the Assistant Bot recording feedback from group members.

1: (User) Have there been any feedback updates?

[Natural Language Understanding]
- Task: *Check Feedback*

2: (Chatbot) Sure, I can help you check the status of your feedback. Which feedback question do you want to check?

3: (User) The one about the midterm, please.

[Natural Language Understanding]
- Feedback Question: *one about the midterm* → *I'd like to request feedback for the midterm that everyone took. Was it too easy, difficult, or just right? If there're any questions that you felt were confusing or difficult, let me know.*

4: (Chatbot) One student found the midterm to be ok, with one question being a little weird and hard to understand. Another student found the midterm to be too difficult, with the topics being hard and questions making no sense. (2 responses)

Figure A9. An example of the Assistant Bot handling feedback checks and providing a cohesive summary.

Appendix A.3. Examples of Assistant Bot and Simulation Fails

— nakamura is now talking —

1: (Chatbot) Thank you Yuki Nakamura. Welcome! How should I call you?

2: (User) Just call me Yuki. Any updates?

3: (Chatbot) Nice to meet you, Yuki. What do you need next, Yuki?

Figure A10. An example of the Assistant Bot not handling the check inboxes tasks due to the STD M State Lock. The user's request for a particular task is ignored by the Assistant Bot due to the chatbot being locked in the name-calling component, thus not allowing it to identify the task requested. The inappropriate response is in red.

— ali is now talking —

1: (User) Please let Kumar Singh know that I'm interested in forming a study group to work on the Data Structures homework together. Thank you!

2: (Chatbot) Of course! It's my pleasure.

Figure A11. An example of the Assistant Bot not handling the user's requested task due to STD M NATEX issues. Expressions of gratitude are captured by NATEX groups, but despite having lower priority, the Assistant Bot captured the user's "Thank you!" rather than the task request through NATEX and generated the inappropriate response. The inappropriate response is in red.

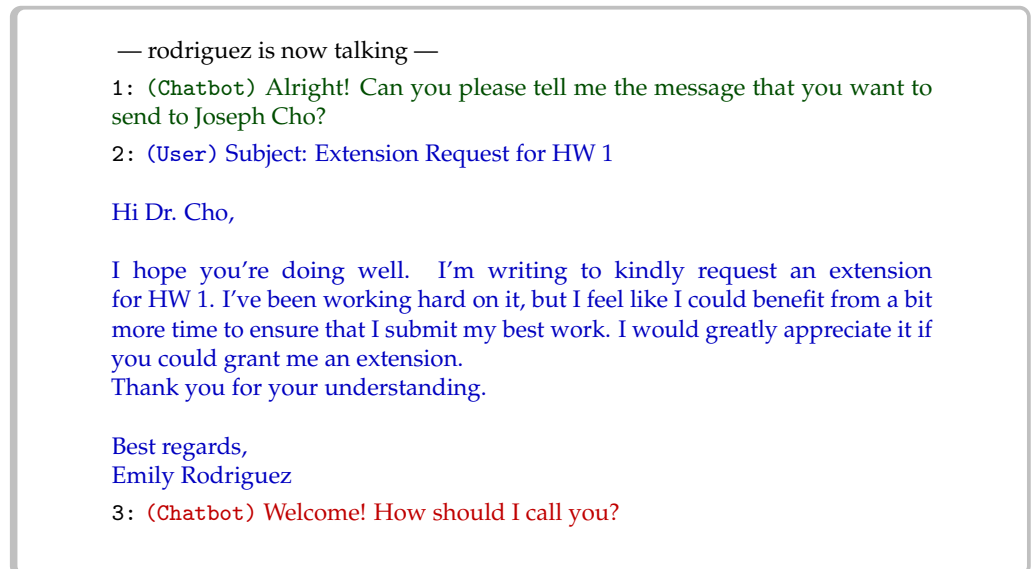


Figure A12. An example of the Assistant Bot returning to the name calling component due to STDM system bugs. The user's request for a message included multiple new lines, which the STDM cannot handle, and thus, the Assistant Bot transitioned back to the start to mitigate the issue. The inappropriate response is in red.

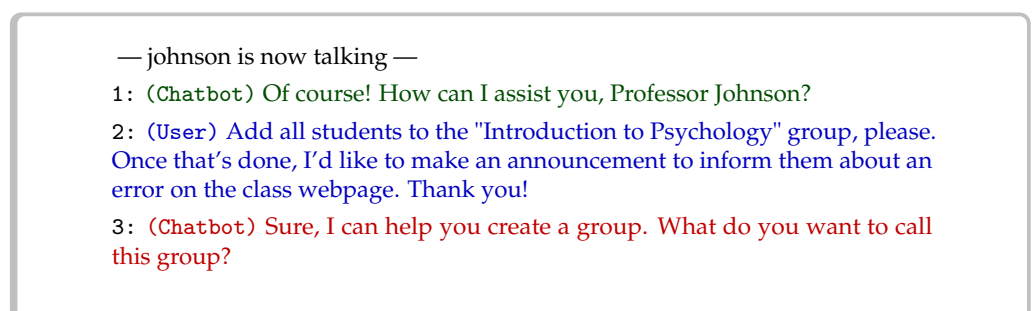


Figure A13. An example of the Assistant Bot being unable to address multiple tasks due to the STDM multiple task requests error. When the user requested multiple tasks, the Assistant Bot's response addressed only one of the tasks, as it is only able to move to one component which in turn handles one of the tasks. The inappropriate response is in red.

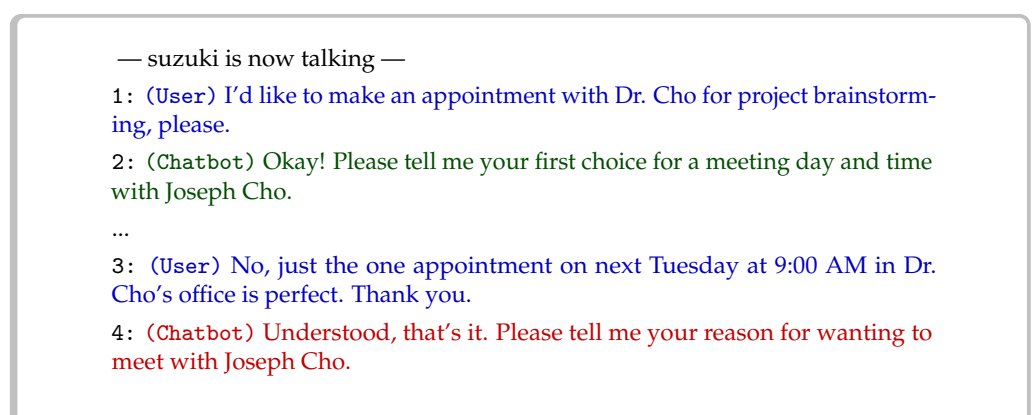


Figure A14. An example of the Assistant Bot being unable to extract appointment reason due to the NLU-DST multiple state extraction error. The appointment reason that is mentioned in L1 is not properly extracted due to the STDM framework in which the reason was extracted last. The inappropriate response is in red.

— silva is now talking —

- 1: (User) Actually, do you mind checking if there are any new announcements from Dr. Cho?
- 2: (Chatbot) You're not in any groups right now, or you do not have any groups that you can request feedback from!!

Figure A15. An example of the Assistant Bot incorrectly identifying a task due to NLU-IC context misinterpretation. The system misinterprets "checking for new announcements" as "checking for feedback." The inappropriate response is in red.

— cho is now talking —

- 1: (Chatbot) Before we begin, please provide your full name.
- 2: (User) Joseph Cho
- 3: (Chatbot) Thank you Joseph Cho. Welcome! How should I call you?
- 4: (User) any updates?

Figure A16. An example of the virtual simulation's ignoring chatbot issue. The simulated user ignores the Assistant Bot's last response, instead directly asking for a task to be completed. The inappropriate response is in red.

— fernandez is now talking —

- 1: (User) Actually, call me Diego. So, any updates?
- 2: (Chatbot) Nice to meet you, Diego. Do you also still want help with checking messages?
- 3: (User) No, it's okay. So, any updates?
- 4: (Chatbot) Understood, I'll move on. What can I do for you, Diego?
- 5: (User) Actually, call me Diego. So, any updates?

Figure A17. An example of the virtual simulation's conversation history error. The simulated user had forgotten that the user already requested the Assistant Bot to "call me Diego," thus repeating the task. The inappropriate response is in red.

— johnson is now talking —

- 1: (User) You can call me Professor Johnson. Any updates?
- 2: (Chatbot) Nice to meet you, Professor Johnson. What can I do for you, Professor Johnson?
- 3: (User) Actually, I wanted to ask if there's a way for me to find a classmate to work together on the Data Structures homework. Can you help me with that?

[Prompt] You are Amanda, a professor teaching the Introduction to Psychology class.

Figure A18. An example of the virtual simulation's user information error. The simulated user, posing as a professor teaching Introduction to Psychology, asks the Assistant Bot for assistance finding a classmate for Data Structures homework, a task that students would request rather than a professor. The inappropriate response is in red. The relevant user information is provided below the conversation.

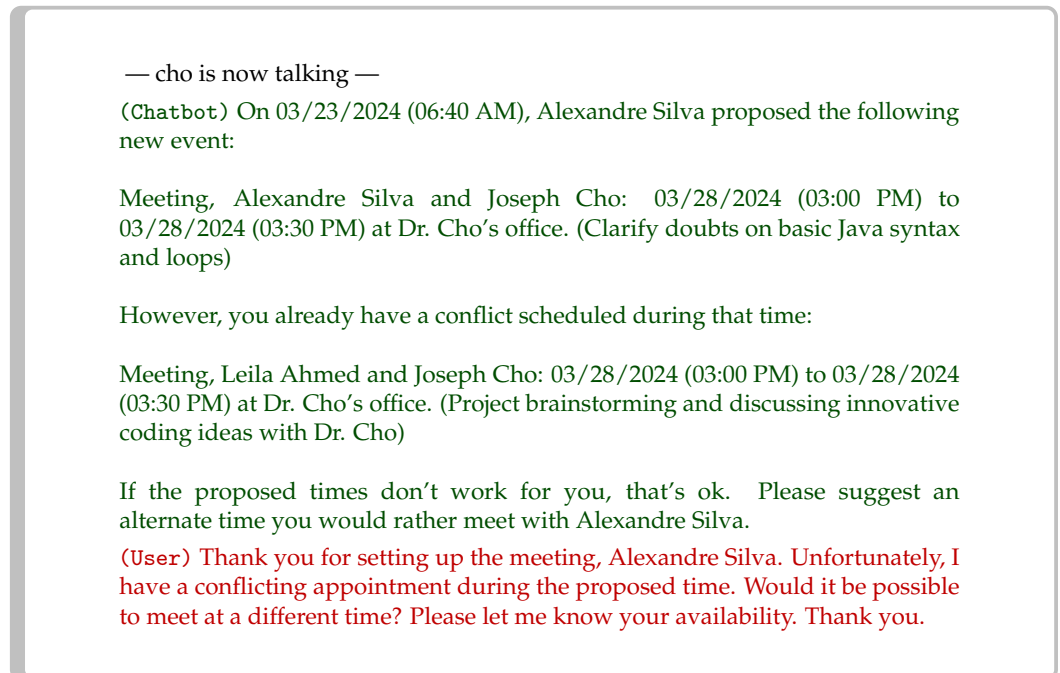


Figure A19. An example of the virtual simulation's conversational partner error. The simulation forgets that it is speaking to the Assistant Bot instead of the sender of the appointment request, and thus, the response directly address the sender of the request rather than the Assistant Bot. The inappropriate response is in red.

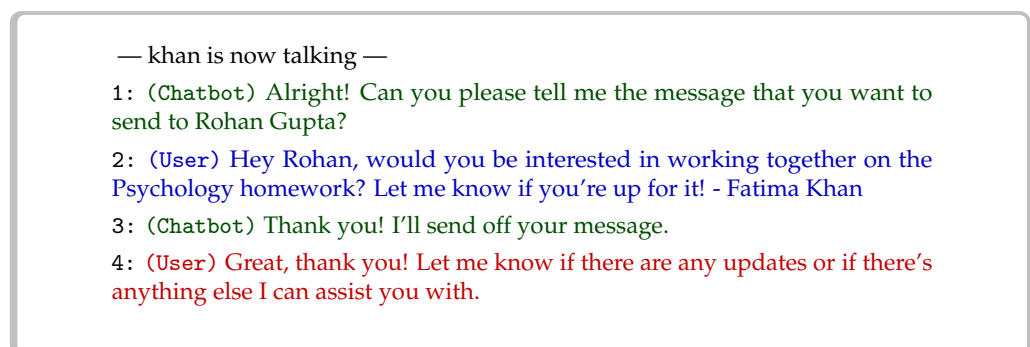


Figure A20. An example of the virtual simulation's role reversal error. The simulation forgets that it is posing as the user and not the Assistant Bot, and thus, the response resembles that of an assistant chatbot rather than the user. The inappropriate response is in red.

References

- Chandan, A.J.; Chattopadhyay, M.; Swaroop Sahoo, S. Implementing Chatbot in Educational Institutes. *IJRAR J.* **2019**, *6*, 44–47.
- Oliveira, P.F.; Matos, P. Introducing a Chatbot to the Web Portal of a Higher Education Institution to Enhance Student Interaction. *Eng. Proc.* **2023**, *56*, 128. [[CrossRef](#)]
- Xiao, Z.; Zhou, M.X.; Chen, W.; Yang, H.; Chi, C. If I Hear You Correctly: Building and Evaluating Interview Chatbots with Active Listening Skills. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020. [[CrossRef](#)]
- Gebreab, S.A.; Salah, K.; Jayaraman, R.; Habib ur Rehman, M.; Ellaham, S. LLM-Based Framework for Administrative Task Automation in Healthcare. In Proceedings of the 2024 12th International Symposium on Digital Forensics and Security (ISDFS), San Antonio, TX, USA, 29–30 April 2024; pp. 1–7. [[CrossRef](#)]
- Hays, S.; White, D.J. Employing LLMs for Incident Response Planning and Review. *arXiv* **2024**, arXiv:2403.01271.
- Musumeci, E.; Brienza, M.; Suriani, V.; Nardi, D.; Bloisi, D.D. LLM Based Multi-agent Generation of Semi-structured Documents from Semantic Templates in the Public Administration Domain. In Proceedings of the International Conference on Human-Computer Interaction, Washington, DC, USA, 29 June–4 July 2024; pp. 98–117.

7. Jung, H.; Yeen, H.; Lee, J.; Kim, M.; Bang, N.; Koo, M.W. Enhancing Task-Oriented Dialog System with Subjective Knowledge: A Large Language Model-based Data Augmentation Framework. In *Proceedings of the 11th Dialog System Technology Challenge, Prague, Czech Republic, 11 September 2023*; Chen, Y.N., Crook, P., Galley, M., Ghazarian, S., Gunasekara, C., Gupta, R., Hedayatnia, B., Kottur, S., Moon, S., Zhang, C., Eds.; Association for Computational Linguistics : Kerrville, TX, USA, 2023; pp. 150–165.
8. Pan, Y.; Cadamuro, D.; Groh, G. Data-Augmented Task-Oriented Dialogue Response Generation with Domain Adaptation. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation, Hong Kong, China, 2–4 December 2023*; Huang, C.R., Harada, Y., Kim, J.B., Chen, S., Hsu, Y.Y., Chersoni, E., Pranav, A., Zeng, W.H., Peng, B., Li, Y., et al., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 96–106.
9. Mendoza, S.; Hernández-León, M.; Sánchez-Adame, L.M.; Rodríguez, J.; Decouchant, D.; Meneses-Viveros, A. Supporting Student-Teacher Interaction through a Chatbot. In *Proceedings of the Learning and Collaboration Technologies. Human and Technology Ecosystems, Copenhagen, Denmark, 19–24 July 2020*; Zaphiris, P., Ioannou, A., Eds.; Springer: Cham, Switzerland, 2020; pp. 93–107.
10. Wang, H.; Wang, L.; Du, Y.; Chen, L.; Zhou, J.; Wang, Y.; Wong, K.F. A Survey of the Evolution of Language Model-Based Dialogue Systems. *arXiv* **2023**, arXiv:2311.16789.
11. Yi, Z.; Ouyang, J.; Liu, Y.; Liao, T.; Xu, Z.; Shen, Y. A Survey on Recent Advances in LLM-Based Multi-turn Dialogue Systems. *arXiv* **2024**, arXiv:2402.18013.
12. Sun, J.; Tian, Y.; Zhou, W.; Xu, N.; Hu, Q.; Gupta, R.; Wieting, J.; Peng, N.; Ma, X. Evaluating Large Language Models on Controlled Generation Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023*; Bouamor, H., Pino, J., Bali, K., Eds.; Singapore: Cham, Switzerland, 2023; pp. 3155–3168. [[CrossRef](#)]
13. Zhu, Z.; Sun, Z.; Yang, Y. HaluEval-Wild: Evaluating Hallucinations of Language Models in the Wild. *arXiv* **2024**, arXiv:2403.04307.
14. Sekulic, I.; Terragni, S.; Guimarães, V.; Khau, N.; Guedes, B.; Filipavicius, M.; Manso, A.F.; Mathis, R. Reliable LLM-based User Simulator for Task-Oriented Dialogue Systems. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*; Graham, Y., Liu, Q., Lampouras, G., Iacobacci, I., Madden, S., Khalid, H., Qureshi, R., Eds.; Association for Computational Linguistics: St. Julians, Malta, 2024; pp. 19–35.
15. Wan, D.; Zhang, Z.; Zhu, Q.; Liao, L.; Huang, M. A Unified Dialogue User Simulator for Few-shot Data Augmentation. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022*; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2022; pp. 3788–3799. [[CrossRef](#)]
16. Liu, Y.; Jiang, X.; Yin, Y.; Wang, Y.; Mi, F.; Liu, Q.; Wan, X.; Wang, B. One Cannot Stand for Everyone! Leveraging Multiple User Simulators to train Task-oriented Dialogue Systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023*; Long Papers; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; Volume 1, pp. 1–21. [[CrossRef](#)]
17. Lu, X.; Wang, X. Generative Students: Using LLM-Simulated Student Profiles to Support Question Item Evaluation. In *Proceedings of the 11th ACM Conference on Learning @ Scale, New York, NY, USA, 18–20 July 2024*; pp. 16–27. [[CrossRef](#)]
18. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288.
19. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825.
20. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774.
21. Zhang, J.; Bui, T.; Yoon, S.; Chen, X.; Liu, Z.; Xia, C.; Tran, Q.H.; Chang, W.; Yu, P. Few-Shot Intent Detection via Contrastive Pre-Training and Fine-Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021*; Moens, M.F., Huang, X., Specia, L., Yih, S.W.t., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2021; pp. 1906–1912. [[CrossRef](#)]
22. Parikh, S.; Tiwari, M.; Tumbade, P.; Vohra, Q. Exploring Zero and Few-shot Techniques for Intent Classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2024*; Industry Track; Sitaram, S., Beigman Klebanov, B., Williams, J.D., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 744–751. [[CrossRef](#)]
23. Heck, M.; Lubis, N.; Ruppik, B.; Vukovic, R.; Feng, S.; Geishausser, C.; Lin, H.C.; van Niekerk, C.; Gasic, M. ChatGPT for Zero-shot Dialogue State Tracking: A Solution or an Opportunity? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2024*; Volume 2: Short Papers; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; Volume 2, pp. 936–950. [[CrossRef](#)]
24. Zang, X.; Rastogi, A.; Sunkara, S.; Gupta, R.; Zhang, J.; Chen, J. MultiWOZ 2.2: A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, Online, 9 July 2020*; Wen, T.H., Celikyilmaz, A., Yu, Z., Papangelis, A., Eric, M., Kumar, A., Casanueva, I., Shah, R., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2020; pp. 109–117. [[CrossRef](#)]
25. Rastogi, A.; Zang, X.; Sunkara, S.; Gupta, R.; Khaitan, P. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020*; pp. 8689–8696.

26. Feng, Y.; Lu, Z.; Liu, B.; Zhan, L.; Wu, X.M. Towards LLM-driven Dialogue State Tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023*; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 739–755. [[CrossRef](#)]
27. Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, Bali, Indonesia, 1–4 November 2023; Long Papers*; Park, J.D.C., Arase, Y., Hu, B., Lu, W., Wijaya, D., Purwarianti, A., Krisnadhi, A.A., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; Volume 1, pp. 675–718. [[CrossRef](#)]
28. Chung, W.; Cahyawijaya, S.; Wilie, B.; Lovenia, H.; Fung, P. InstructTODS: Large Language Models for End-to-End Task-Oriented Dialogue Systems. In *Proceedings of the 2nd Workshop on Natural Language Interfaces, Bali, Indonesia, 1 November 2023*; Chen, K., Ku, L.W., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 1–21. [[CrossRef](#)]
29. Hudeček, V.; Dusek, O. Are Large Language Models All You Need for Task-Oriented Dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, 11–15 September 2023*; Stoyanchev, S., Joty, S., Schlangen, D., Dusek, O., Kennington, C., Alikhani, M., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 216–228. [[CrossRef](#)]
30. Toxtli, C.; Monroy-Hernández, A.; Cranshaw, J. Understanding Chatbot-mediated Task Management. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 21–26 April 2018*; pp. 1–6. [[CrossRef](#)]
31. Deriu, J.M.; Cieliebak, M. Towards a Metric for Automated Conversational Dialogue System Evaluation and Improvement. In *Proceedings of the 12th International Conference on Natural Language Generation, Tokyo, Japan, 29 October–1 November 2019*; van Deemter, K., Lin, C., Takamura, H., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2019; pp. 432–437. [[CrossRef](#)]
32. Ghandeharioun, A.; Shen, J.H.; Jaques, N.; Ferguson, C.; Jones, N.; Lapedriza, A.; Picard, R. Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, QC, Canada, 8–14 December 2019*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
33. Deriu, J.; Tuggener, D.; von Däniken, P.; Campos, J.A.; Rodrigo, A.; Belkacem, T.; Soroa, A.; Agirre, E.; Cieliebak, M. Spot The Bot: A Robust and Efficient Framework for the Evaluation of Conversational Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020*; Webber, B., Cohn, T., He, Y., Liu, Y., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2020; pp. 3971–3984. [[CrossRef](#)]
34. Yang, R.; Li, Z.; Tang, H.; Zhu, K. ChatMatch: Evaluating Chatbots by Autonomous Chat Tournaments. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland 22–27 May 2022*; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2022; Volume 1, pp. 7579–7590. [[CrossRef](#)]
35. Abeysinghe, B.; Circi, R. The Challenges of Evaluating LLM Applications: An Analysis of Automated, Human, and LLM-Based Approaches. *arXiv* **2024**, arXiv:2406.03339.
36. Chiang, C.H.; Lee, H.y. Can Large Language Models Be an Alternative to Human Evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2022; Long Papers*; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; Volume 1, pp. 15607–15631. [[CrossRef](#)]
37. Svikhnushina, E.; Pu, P. Approximating Online Human Evaluation of Social Chatbots with Prompting. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, 11–15 September 2023*; Stoyanchev, S., Joty, S., Schlangen, D., Dusek, O., Kennington, C., Alikhani, M., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 268–281. [[CrossRef](#)]
38. Park, J.S.; O'Brien, J.D.C.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior. *arXiv* **2023**, arXiv:2304.03442.
39. Zhou, X.; Zhu, H.; Mathur, L.; Zhang, R.; Yu, H.; Qi, Z.; Morency, L.P.; Bisk, Y.; Fried, D.; Neubig, G.; et al. SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents. *arXiv* **2024**, arXiv:2310.11667.
40. Finch, J.D.; Choi, J.D. Emora STDM: A Versatile Framework for Innovative Dialogue System Development. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Virtual Meeting, 1–3 July 2020*; Pietquin, O., Muresan, S., Chen, V., Kennington, C., Vandyke, D., Dethlefs, N., Inoue, K., Ekstedt, E., Ultes, S., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 261–264. [[CrossRef](#)]
41. Madotto, A.; Liu, Z.; Lin, Z.; Fung, P. Language Models as Few-Shot Learner for Task-Oriented Dialogue Systems. *arXiv* **2020**, arXiv:2008.06239.
42. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020, Volume 33, pp. 1877–1901.
43. Yuan, C.; Xie, Q.; Ananiadou, S. Zero-shot Temporal Relation Extraction with ChatGPT. In *Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, Toronto, ON, Canada, 13 July 2023*; Demner-Fushman, D., Ananiadou, S., Cohen, K., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 92–102. [[CrossRef](#)]

44. Wei, X.; Cui, X.; Cheng, N.; Wang, X.; Zhang, X.; Huang, S.; Xie, P.; Xu, J.; Chen, Y.; Zhang, M.; et al. Zero-Shot Information Extraction via Chatting with ChatGPT. *arXiv* **2023**, arXiv:2302.10205.
45. Salewski, L.; Alaniz, S.; Rio-Torto, I.; Schulz, E.; Akata, Z. In-Context Impersonation Reveals Large Language Models' Strengths and Biases. In Proceedings of the 37th Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
46. Kim, H.; Hessel, J.; Jiang, L.; West, P.; Lu, X.; Yu, Y.; Zhou, P.; Bras, R.; Alikhani, M.; Kim, G.; et al. SODA: Million-scale Dialogue Distillation with Social Commonsense Contextualization. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 12930–12949. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.