A User Simulator Architecture for Socially-Aware Conversational Agents

Alankar Jain, Florian Pecune, Yoichi Matsuyama, Justine Cassell
School of Computer Science, Carnegie Mellon University
{alankarj,fpecune,yoichim,justine}@cs.cmu.edu

ABSTRACT

Over the last two decades, Reinforcement Learning (RL) has emerged the method of choice for data-driven dialog management. However, one of the limitations of RL methods for the optimization of dialog managers in the context of virtual conversational agents, is that they require a large amount of data, which is often unavailable, particularly when the dialog deals with complex discourse phenomena. User simulators help address this problem by generating synthetic data to train RL agents in an online fashion. In this work, we extend user simulators to the case of socially-aware conversational agents, that combine task and social functions. We propose a novel architecture that takes into consideration the user's conversational goals and generates both task and social behaviour. Our proposed architecture is general enough to be useful for training socially-aware conversational agents in any domain. As a proof of concept, we construct a user simulator for training a conversational recommendation agent and provide evidence towards the effectiveness of the approach.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**; *Reinforcement learning*; *Modeling and simulation*;

KEYWORDS

User simulator, dialog system, conversational agent, rapport

ACM Reference Format:
Alankar Jain, Florian Pecune, Yoichi Matsuyama, Justine Cassell . 2018. A User Simulator Architecture for Socially-Aware Conversational Agents. In *IVA '18: International Conference on Intelligent Virtual Agents (IVA '18), November 5–8, 2018, Sydney, NSW, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3267851.3267916

1 INTRODUCTION

In the last few years, people have become increasingly comfortable interacting with conversational agents [35]. During these interactions, people pursue multiple **conversational goals**, such as those that fulfill (a) *propositional* functions: contributing informational content to the conversation, and (b) *interpersonal* functions: managing relational goals such as building rapport [33]. While personal assistants such as Apple Siri, Amazon Echo, Microsoft Cortana and Google Home focus on users' propositional goals by getting a specific task done, other conversational agents simply engage their users in social chit-chat, trying to build a relationship with them. In between the two, socially-aware conversational agents aim at fulfilling both propositional and interpersonal goals, using rapport or interpersonal closeness with a user to improve the effectiveness of task dialog, and using task dialog to modify the social situation [2, 4, 7]. Human-human studies have found that rapport between two people can influence task performance in situations as diverse as peer-tutoring [31] and negotiation [8]. Based on these findings, it becomes important to endow today's conversational agents with the ability to build rapport with their users.

Rapport management is a rule-governed process. [32] shows that *politeness* might build rapport early in a relationship but be detrimental later on. [6] describes the ways in which appropriate levels of *self-disclosure* (speaking about oneself) may strengthen a relationship and, in turn, strong rapport can lead to more intimate self-disclosure. *Reciprocal appreciation* supports rapport management by making both interlocutors feel seen. Other strategies such as *praise* and *violation of social norms* can also be linked to rapport enhancement, depending on the timing and context of their usage. A non-exhaustive list of **conversational strategies** and their influence on rapport can be found in [39]. As described, rapport is a dyadic phenomenon, depending on the behavior of all conversation participants, and it evolves over time. A socially-aware agent will therefore have to detect the user's rapport maintenance strategies, and generate appropriate replies that adapt to the user's current move, conversational goal and the current level of rapport.

In modular conversational agent architectures, this reasoning process is usually handled by a component called the Dialog Manager (DM) which is in charge of selecting the best dialog policy to follow given a particular context or dialog state [37]. For instance, an agent should not recommend something to its users before asking about their preferences first, and everything should probably be preceded by a greeting. Although it is quite easy to handcraft such a simple policy (a mapping from state to actions), manually authoring optimal policies for deep and complex dialogs becomes quickly overwhelming [24]. For instance, there is no straightforward answer to whether all the users should share the same optimal dialogue policy, or if different users should have different policies depending on their conversational goals. Integrating rapport and conversational strategies in the reasoning process makes authoring of dialog policies even more complex because of the real-time adaptive nature of rapport maintenance conversational strategies in task dialog.

Data-driven approaches can help address the aforementioned limitations of hand-crafted authoring techniques, and hence, offer

IVA '18, November 5–8, 2018, Sydney, NSW, Australia

ACM ISBN 978-1-4503-6013-5/18/11…$15.00
https://doi.org/10.1145/3267851.3267916

an attractive alternative to them [17]. However, the dynamic nature of rapport maintenance in task dialog requires long-term planning in order to incorporate the delayed consequences of actions in stochastic environments. Reinforcement Learning (RL) provides a framework to learn through delayed rewards and operates through trial-and-error exploration to learn an optimal policy despite unavailability of ground truth labels. This is the reason why RL is especially attractive in the temporal and dynamic settings in which socially-aware conversational agents operate [25]. Not only can an RL-based agent learn how to optimize its task-performance, but it can also learn the complex strategies that humans use to maximize rapport over time. However, RL-based approaches to dialog management suffer from a major shortcoming: they require a large amount of data for training in order to explore all the possible dialog options (typically $>> 10^4$ dialogs) [28]. This is where user simulators step in. A user simulator can generate a large amount of synthetic dialog data on the basis of a relatively small initial corpus which can then be used to train an RL-based DM.

In this paper, we propose a user simulator architecture that closely approximates the behavior of a real social user. We argue that in order to effectively train a socially-aware dialogue manager, such a simulator should distinguish between the different conversational goals of users. The simulated user should also estimate rapport from sequences of conversational strategies (social cues) taken as input and generate both task and social behavior in return in the form of task and conversational strategies, respectively. The rest of the paper is organized as follows: we discuss the related work in section 2. Next, we describe the general architecture of our social user simulator and its different components in section 3. Further, we construct a data-driven user simulator using the proposed architecture for a conversational recommendation agent, SARA (Socially-Aware Robot Assistant) (section 4). Finally, we discuss our results and their implications for the design of virtual agents in section 5 and 6, respectively.

2 RELATED WORK

Reinforcement Learning for dialog systems has been an active area of research for almost two decades now [30]. Further, as reviewed in [13], a variety of approaches for user simulation have been considered in literature to address the data inefficiency of RL approaches. They can be categorized along two distinct dimensions. The first dimension represents the granularity level at which user action is generated. A user simulator can operate at the utterance level, where the output is in natural language, or at the intention (or dialog act) level, in which case the user simulator only generates higher level intentions. Although the utterance-level approach helps to capture the immense variety of human language, it exponentially increases the number of states and actions the user simulator (and by extension, the agent) would have to deal with. For example, a single high level intention, say, requesting for a flight destination, `request(flight_dest)`, could be translated in many different natural language utterances. Hence, the intention-level approach appears more robust and scalable [14]. The second dimension represents the methodology used to build the user simulator. In an agenda-based setting [28], the user intention at any given turn depends upon the user goal (a set of constraints, C, in the form of slot-value pairs and a set of request slots, R) and the user agenda at the given turn. The agenda is a stack-like structure that contains pending user intentions. In Dynamic Bayesian Network (DBN) approach, the user's decision-making is represented through a probabilistic model trained from data [16]. In Inverse Reinforcement Learning (IRL), data is used to directly reverse-engineer the reward model of the user simulator [5]. However, we choose the agenda-based approach because it less data-hungry.

There is rich literature on inference and incorporation of different user behaviours in user simulators [13]. In particular, a line of work closely related to ours is that by [11], who learn different dialog strategies for younger and older simulated users. [29] infer the hidden agenda of users using an Expectation Maximization approach. However, in contrast to these approaches, we incorporate users' conversational goals in our user simulator, which represent a more fundamental aspect of users' conversational behaviour. Thus, our work can be seen as a logical next step towards accurate simulation of real (social) users' behaviour.

Evaluating the performance and the quality of a user simulator is essential before it can be used to train a dialog manager. If the actions (either intentions or utterances) generated are not realistic (e.g. if the simulator starts the interaction by a farewell, or if it asks the same question many times in succession), the agent will not be able to learn any reasonable dialogue policy. There are several ways to evaluate a user simulator but there doesn't seem to be any consensus about the right metric to use [22]. These metrics can broadly be divided into two categories: turn-level and dialog-level. Turn-level metrics measure local consistency between the generated synthetic data and data from real users. These include KL distance (symmetric Kullback-Leibler divergence) to measure dissimilarity of the two distributions (real vs generated). A lower score implies a more realistic generated behaviour. Another turn-level metric is the F1 score (incorporating precision and recall) for the generated user actions. In contrast, dialog-level metrics evaluate the overall quality of the generated dialogues. The Dialog-BLEU score, for instance, measures the similarity between complete real dialogs and the generated ones. Human subjective evaluation is another way to evaluate the quality of the generated sequences of actions. Next, we take a look at prior work on social user simulators.

In PsychSim [23], authors propose a decision-theoretic framework to simulate interactions between multiple agents in a social context. Although this framework allows manual definition of different types of goals for each agent, it does not offer any learning mechanism, meaning that the dialog policy selected by the agent relies heavily on handcrafted rules. Many recent works have attempted to address this problem by building socially aware RL-based agents [15] [12] [26]. However, all of these works circumvented the need for a user simulator by oversimplifying the state and action spaces to ensure that the RL agent learns effectively from limited data, thereby reducing the usefulness of such agents. There also has been some work on user simulators to train socially-aware dialog systems. [9] derive a socially-inspired reward from positive and negative appraisals inferred from a simulated user's task strategies and use it as a shaping reward function to train their RL agent. [38] consider the problem of learning conversational systems that interleave task and non-task (social) content to improve task success rate and increase user engagement. They used a chatbot as a user simulator. However, none of these user simulators generate both task and social behaviors explicitly, which, as explained previously, are essential to enable rich modeling of the state space and are required as inputs by a socially-aware conversational agent. Further, they do not incorporate the influence of the user's conversational goals on its behavior. To the best of our knowledge, our work is the first attempt to propose an architecture for a social user simulator that generates both task and social behaviors depending on a user's conversational goals.

3 USER SIMULATOR ARCHITECTURE

A schematic diagram of the proposed architecture for a social user simulator is shown in figure 1. At any turn $n \in \{0, 1, ..., L\}$ in the dialog, where L denotes the maximum length of the dialog, the input to the user simulator is comprised of the agent action A_t and the output includes the state S_t and the reward R_t. In the context of a user simulator for a socially-aware conversational agent, the action A_t and state S_t are comprised of the following:

Task Strategies (TS): A task strategy is comprised of two distinct components: a string that succinctly captures the *function* of the strategy, e.g.: `inform`, `request`, `bye`, `confirm`, etc. and a set of *slots/slot-value* pairs, e.g.: `feedback=good`, `primary-goal` , etc. that capture the information provided or requested. For example, a task strategy for requesting feedback for a recommendation made by the agent could be represented as `request(feedback)`. If the recommendation was relevant, the user response could be represented as the task strategy `inform(feedback=good)`. We denote agent and user task strategies by T_n^a and T_n^u, respectively.

Conversational Strategies (CS): A conversational strategy is a social strategy used by the agent or the user for building, maintaining or destroying rapport. We denote agent and user conversational strategies by C_n^a and C_n^u, respectively. In our work we focus on 6 and 8 different user and agent conversational strategies, respectively, as described in tables 1 and 2.

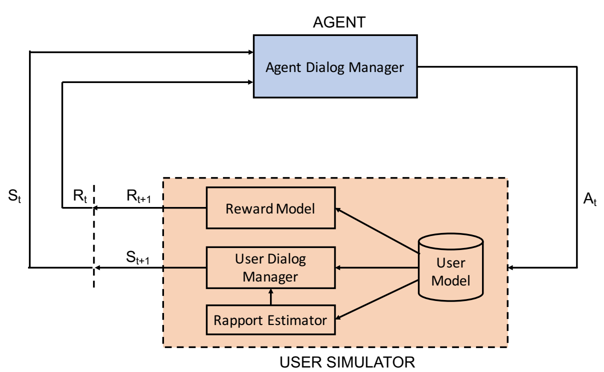

Figure 1: Social User Simulator Architecture

Next, we provide a brief description of all the major components of the social user simulator architecture.

3.1 User Model

The User Model captures all the information relevant for modeling the distinct types of users whose behaviour we wish to simulate through the social user simulator. This information will depend upon the specific task and the goals of the user simulator and can be obtained through analysis of human-human (or human-machine) data such as Wizard of Oz (WoZ) corpus [10]. The user model can influence both the aspects of user behaviour: task and social. In terms of task, in a movie recommendation domain, for example, the user model can take the form of a prior over the slot values the user prefers (certain genres, actors, directors, etc.). In terms of social behavior, the user model could capture aspects such as whether the user prefers self disclosing information over praising for building rapport. Finally, the user model can also decide whether the user prefers longer interactions over shorter interactions. We describe the user model for our case study in section 4.4.

Speaker	Conversational Strategies
User	SD, PR, HE, VSN, QESD, NONE
SARA	SD, PR, HE, VSN, QESD, ACK, ASN, NONE

Table 1: User and agent conversational strategies

Code	Conversational Strategy	Example
SD	Self-disclosure	This is my first time here.
PR	Praise	You are great, SARA.
HE	Hedging (or Indirectness)	That's okay, I guess.
VSN	Violation of Social Norm	So how are things?
QESD	Question to Elicit Self-Disclosure	What are your goals?
ACK	Acknowledgment	uh huh
ASN	Adherence of Social Norm	May I ask your name?
NONE	No Conversational Strategy	I'm sure you'll enjoy it.

Table 2: Conversational Strategy codes and examples

3.2 Rapport Estimator

This module estimates level of rapport (as experienced by the user) at every turn in the interaction. Since we only focus on verbal behaviour in this work, this module is responsible for updating the rapport value by taking into account the past history of agent and user conversational strategies and task strategies. The rapport estimated by the user simulator is taken into account by the user dialog manager to decide the next appropriate task and social strategy. For example, if the rapport value is below a certain threshold, the likelihood of acceptance of a recommendation made by the agent might become lower than its base value. This is one way in which the social aspect of the interaction can affect the task aspect. The estimated rapport value can also be used to modulate the social behaviour of the user simulator. For example, if the rapport value is average and the agent had praised the user in the last turn, then the user simulator might decide to engage in reciprocal appreciation to increase the rapport even further. Section 4.5 describes the rapport estimator for our case study in detail.

3.3 User Dialog Manager

The Dialog Manager module is responsible for generating user task strategy (task reasoner) and conversational strategy (social reasoner) depending upon the user model, past agent actions and the estimated rapport (output of rapport estimator). For example, if the agent makes a relevant recommendation and then requests for feedback (agent task strategy: `request(feedback)`), and if the estimated rapport is high, the user simulator might decide to not only accept the recommendation (user task strategy: `inform(feedback=good)`) but also praise the system for making a relevant recommendation (user conversational strategy: PR). The dialog manager can either be rule-based, data-driven, or a hybrid of the two.

3.4 Reward Model

This module is responsible for providing a numerical reward to the agent. This reward could be task-dependent, social, or a combination of the two, and is used by the RL agent as a learning (reinforcement) signal. For example, the user simulator might provide a positive reward for a relevant recommendation, or negative reward if the agent takes too long to make a recommendation. Section 4.7 describes the reward model for our case study.

4 CASE STUDY

In this section, we explain how we trained a user simulator based on the architecture described in the previous section using an existing dataset. The goal of this user simulator is to simulate the behavior of a conference attendee asking for recommendations about sessions to attend and people to meet. First, we describe the context in which the dataset used to train our simulator was collected, then we discuss the annotations and analyses we performed on this dataset, and finally, we explain how we used the annotated dataset to train the different modules of our user simulator.

4.1 Dataset: Conference Personal Assistant

Socially Aware Robot Assistant (SARA) [20] was deployed as a personal assistant in a large high profile conference in 2017 [21]. The conference lasted four days, and was filled with discussions, lectures, workshops, and showcases. The agent was designed to play the role of a matchmaker in order to help attendees get the most out of the conference. Many attendees interacted with the agent, which in turn, helped them fulfill their goals, whether it be networking, or learning about new technologies, or even getting to know about the best places to eat or party at during the conference. During every interaction, SARA elicited interests and goals of the participants, and recommended them relevant people to meet and sessions to attend. After every recommendation, SARA requested for explicit feedback regarding whether the recommendation matched user's interests or not. If the user liked a recommendation, SARA further asked if the user was willing to accept a private message (on a conference-specific mobile application) as a reminder for the session or for introducing the user to the person recommended. A typical interaction between users and the agent has been shown in table 3. Our corpus contains data from 64 of these interactions and includes both attendee's and SARA's video and speech transcription. This accounts for over 5 hours of interaction (total time = 323.8 min, mean session duration = 5.06 min, standard deviation = 1.06 min).

4.2 Dataset Annotation

Before training the different components of the user simulator, we annotated each interaction in our dataset for rapport, conversational strategies and task performance. Human annotators quantified the level of rapport by rating 30-second thin slices [1] of the interactions for rapport on a Likert Scale from 1 to 7 [40], with 1 being the lowest and 7 being the highest. Further, user utterances were annotated for 5 conversational strategies: SD, PR, HE, VSN and QESD. If an utterance didn't contain any of these 5 strategies, it was marked as NONE. Inter-rater agreement was achieved between 4 annotators for annotation of rapport and all conversational strategies (Krippendorff's alpha, $\alpha > 0.7$). Finally, task performance for each interaction was annotated as a categorical variable-length vector with the length of the vector equal to the number of recommendations made during the interaction and each element representing the outcome of the recommendation depending on user's response to SARA's explicit confirmation task strategy requesting for feedback (`request(feedback)`): 0 for rejection and 1 for positive feedback followed by acceptance of SARA's request for sending a message on conference-specific mobile application (`request(send_msg)`) (*"Great recommendation. Could you also send me a reminder?"*). Rejection was further categorized into (a) rejection with (*"Please recommend me someone from Asia, instead."*) or (b) without a specified reason (*"No, I don't like it."*), and (c) positive feedback (*"That's a great recommendation."*) followed rejection of SARA's request for sending a message (*"No, thank you. I'll reach out myself."*). For each session, we used the annotated 30-second thin slice rapport values to construct a cubic spline interpolation as shown in figure 2 in order to learn rapport as a continuous function of time. Further, from the text transcripts of the sessions, we mined the time stamps when SARA asked for feedback on its recommendations.

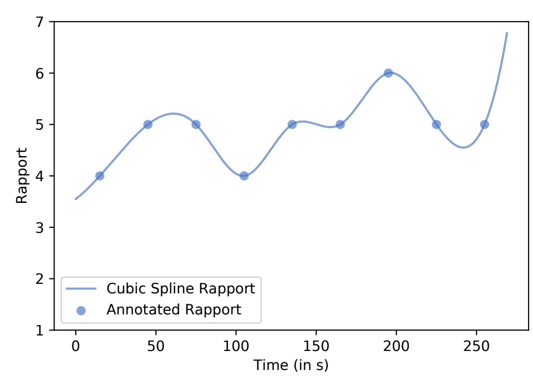

Figure 2: Spline interpolation based on annotated rapport

4.3 Data Analysis: Two Types of Users

We categorized users into two types depending upon the conversational goals they *focused* on accomplishing during their interaction with SARA:

Speaker	Utterance	Task Intention	Conv. Strategy
SARA	I am so glad you're here. I am SARA, what's your name?	`greeting()`	SD
User	My name is X.	`greeting()`	SD
SARA	I'm here in Y to be your personal assistant and help you to get the most out of the experience. So is this your first time at Y or have you been here before?	`introduce()`, `request(first_time)`	NONE
User	I've been a number of times before so this is probably fifth or sixth time.	`inform(first_time=False)`	SD
SARA	Wow! You must be a global rock star to be invited so many times.	`give_feedback()`	PR
User	That's very courteous of you.	`null()`	NONE
SARA	If you wouldn't mind sharing your goals for attending Y this year I can try to help you achieve them.	`request(primary_goal)`	PR
User	Great, so I always like to meet other people learn about concepts that I don't fully understand currently, like this idea of a digital human. Is there an application of that as I'm the general counsel of a big company. Can we use that at our company.	`inform(primary_goal=person)`	SD
SARA	That's really intriguing this is my favorite part, let's see if we can find some interesting people to meet.	`request(person_reco)`	ASN, SD
SARA	Just wait while I look this up. I think you'd enjoy meeting this person on the screen are details	`inform(info_person=Z)`	SD
User	That's a wonderful idea! Z is a client.	`inform(feedback=True)`	PR, SD
SARA	Did I have a good match?	`request(feedback)`	NONE
User	A very good match thank you for that!	`inform(feedback=True)`	PR
SARA	So I can do one of two things: I can send a message to the attendee introducing you or I can give you the information and then you can yourself get in touch. Would you like me to send a message?	`request(send_msg)`	NONE
User	With great pleasure, that would be wonderful if you did.	`inform(send_msg=True)`	PR
SARA	I understand you are busy so I can let you go if you want but before I do you want me to take a quick selfie of us?	`request(selfie)`	NONE
User	Please that would be wonderful.	`inform(selfie=True)`	NONE
SARA	Okay one two three smile cool.	`take_selfie()`	NONE
SARA	Well feel free to come back in the meantime enjoy the meeting, and it was nice working with you bye!	`bye()`	NONE
User	Great, bye!	`bye()`	NONE

Table 3: A typical interaction between SARA and a user.

- **Propositional (*P-Type*)**: If a user prefers accomplishing propositional goals (receiving recommendations) over interpersonal goals (building rapport with SARA).
- **Interpersonal (*I-Type*)**: If interpersonal goals matter to a user as much as (if not more than) propositional goals.

We used k-means clustering algorithm [18] ($k = 2$) where each user was represented by its cumulative use of conversational strategies in a 6-dimensional space with each dimension corresponding to one of the 6 annotated conversational strategies employed by users (table 1). We repeated the algorithm with 5 different random seeds and assigned clusters based on the majority vote to mitigate the impact of random initialization. The Silhouette Score [27] for the final clustering was 0.32 (score for random cluster assignment is 0). We were able to identify two distinct clusters with a clear distinction in terms of their cumulative CS use (figure 3). The difference in average cumulative CS use was statistically significant for the two clusters, as measured by one-way Multivariate Analysis of Variance [3] (MANOVA) (Wilks' Lambda = 0.24, $F(6, 57) = 29.7$, $p < 0.0001$). Further, these differences were significant at $p < 0.05$ with large effect size (Cohen's D) for all strategies, except for QESD and VSN. The results for two-sided Welch's t-test [34] (after Bonferroni correction) were: SD: ($t = 5.06, p = 0.00003, D = 1.48$), PR: ($t = 3.49, p = 0.012, D = 1.16$), HE: ($t = 3.07, p = 0.026, D = 0.92$) and NONE: ($t = 4.42, p = 0.00006, D = 1.03$). While users from the first cluster predominantly used SD and NONE, those from the second cluster employed a greater variety of conversational strategies. We hypothesize that the former cluster corresponds to P-type users as they do not use the verbal behavior (interpersonal strategies) commonly associated with building of rapport, and hence, do not signal a social intention of pursuing interpersonal goals through the conversation. The second cluster, analogously, corresponds to I-type users. It should be noted that the difference in agent average cumulative CS use between the two clusters was not statistically significant (Wilks' Lambda = 0.84, $F(7, 56) = 1.58, p = 0.16$), thus, the difference in cumulative CS use between these two types of users can be attributed to users' conversational goals alone.

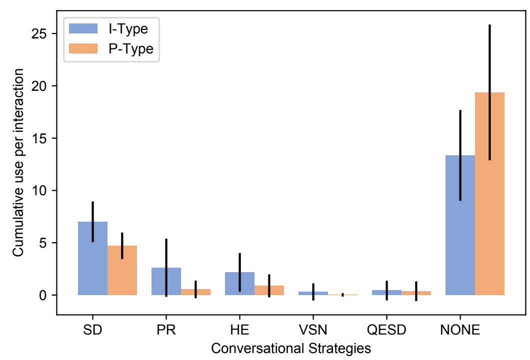

Figure 3: Cumulative CS use for I-type and P-type users

4.4 User Model

We used the following features to model a unique user in the conference. These features are used by the rapport estimator, user dialog manager (social reasoner and task reasoner) modules (as shown in figure 1) to generate appropriate task and social strategies for the simulated user. All these features are random variables, parameters of whose distributions were learned from the data using maximum likelihood estimation (MLE).

- **Number of recommendations (N_r):** The total number of recommendations (between 1 and 6) requested by the user. For a given total number of recommendations, how many recommendations were for people to meet (N_p) and how many were for sessions to attend (N_s). Thus, $N_r = N_p + N_s$.
- **Time at hand:** This feature is derived from N_r. We classify users into three categories depending upon the amount of time they have: a user is assumed to have less time (LT) if $1 \leq N_r \leq 2$, average amount of time (AT) if $3 \leq N_r \leq 4$ and more time (MT) if $5 \leq N_r \leq 6$. In our dataset, 40.6% users had LT, 45.3% had AT and the remaining 14.1% had MT.
- **Primary goal:** This feature is also derived from N_r. If $N_p \geq N_s$, user's primary goal is to meet people, and vice versa. This feature is used by the rule-based agent to determine

which type of recommendations to give first. In our dataset, 57.8% users had meeting people as their primary goal and the remaining 42.2% wanted session recommendations primarily.

- **Conversational Goal:** I-type or P-type, as explained in the previous section. The prior probabilities for a user being I-type or P-type were learned from the data using MLE. In our dataset, 64.1% of the users were P-Type and the remaining 35.9% were I-Type.

4.5 Rapport Estimator (RE)

We estimated rapport using a neural network (multi-layer perceptron) with a single hidden layer (with leaky Rectifier Linear Unit (ReLU) activation [19]). In order to predict the rapport at any given turn, R_n, input to the neural network comprised of binary representations of raw data upto two turns in the past: agent task strategy (T^a), user and agent conversational strategies (C^u, C^a) and rapport values (R). We performed extensive hyperparameter search across a 3×3 grid to select the most optimal combination of features (only CS, CS + rapport, CS + rapport + agent task strategy) and window size (W) (1, 2, linear combination (LC)). For example, the input for window size 2 and CS + rapport is the concatenation of $[C^a_n; C^a_{n-1}]$, $[C^u_{n-1}; C^u_{n-2}]$ and $[R_{n-1}; R_{n-2}]$. The input for linear combination window for the same features is the concatenation of $\alpha_1 C^a_n + \alpha_2 C^a_{n-1}$, and $\beta_1 C^u_{n-1} + \beta_2 C^u_{n-2}$, $\gamma_1 R_{n-1} + \gamma_2 R_{n-2}$, where $\alpha = [\alpha_1, \alpha_2]$, $\beta = [\beta_1, \beta_2]$ and $\gamma = [\gamma_1, \gamma_2]$ are trainable parameters. The best model for each of these 9 configurations was chosen based on the accuracy on the validation set (10% of the full dataset) by tuning for the optimal number of training epochs, hidden layer size (8 or 16) and a grid search over the value of the slope of the leaky ReLU activation, and then the best model out of these 9 different models was selected for rapport estimation (table 5). The rapport estimated at every turn in the dialog by the rapport estimator is used by the dialog manager module, as explained next.

4.6 User Dialog Manager

4.6.1 Social Reasoner (SR). We used the same architecture as that for the rapport estimator except that the output of the neural network for social reasoner is 6-dimensional (number of user conversational strategies). We perform a hyperparameter search similar to the one described for RE, except, we also tune for the threshold used to convert the probability vector (output of the social reasoner neural network) into a binary conversational strategy vector and select the best model out of the 9 configurations (table 5) for final evaluation.

4.6.2 Task Reasoner . We constructed the task reasoner based on the Finite State Machine (FSM) used to construct the rule-based version of SARA deployed in the conference. Task reasoner was largely composed of handcrafted rules, apart from one stochastic decision point: recommendation acceptance. The probability of recommendation acceptance for different rapport levels for different clusters (P-Type, I-Type, Overall) has been given in table 4.

4.7 Reward Model

The user simulator gives a positive numerical reward (5 points) to the agent for every accepted recommendation. However, there are per-turn penalties for dialog length which depend upon the time the user has and the conversational goal of the user. The relative values of the penalties were selected based on the assumption that P-type users and users with less time at their disposal prefer shorter interactions. If the user has less time, it assigns only a small negative penalty (-0.25) if it is I-type, i.e. focuses on both interpersonal and propositional goals, but a large negative penalty (-1) if it is P-type. If the user has average amount of time, it assigns (a small) negative penalty only if it is P-type. Finally, if the user has more than average amount of time, it doesn't penalize for the length of the interaction at all.

Rapport level	P-Type	I-Type	Overall
<3	0.64	0.60	0.63
3-4	0.72	0.56	0.67
4-5	0.70	0.57	0.65
>5	0.56	0.80	0.65

Table 4: Recommendation Acceptance Model

	RE	SR
I-Type	(CS + rapport, LC)	(CS + rapport + agent TS, LC)
P-Type	(CS + rapport, 1)	(CS + rapport, LC)
Overall	(CS + rapport, LC)	(CS + rapport, 1)

Table 5: Best models (feature, window) for RE and SR

5 RESULTS AND ANALYSIS

We consider two variants of the user simulator model, as described below, in order to validate our approach, and support our initial claim that incorporation of user conversational goals leads to more accurate user simulation.

- **Unimodal:** In this model, we do not cluster users based on their conversational strategy use, i.e., we assume that all users pursue similar goals in their interactions with SARA. Hence, we assume that the dataset has a single mode and therefore, interactions can be considered to be identically distributed.
- **Bimodal:** In this model, we cluster users based on their conversational strategy use, i.e., we assume that there are predominantly two types of users: P-type and I-type. In other words, we assume that the dataset has two distinct modes and hence, interactions can be considered identically distributed only conditioned on the type of the user.

For unimodal user simulator, we use the entire dataset to train the rapport estimator, social reasoner and the recommendation acceptance model, whereas, for bimodal user simulator, we segregate the dataset into two clusters (as described in section 4.3) and pool the data belonging to each cluster to train each of the three modules. We hypothesize that the bimodal user simulator will be able to model the behaviour of real users more closely. Note that while we had provided evidence towards existence of two distinct user types in our corpus previously, our hypothesis here is that using this information is also crucial for construction of an accurate social user simulator. Next, we discuss the performance of individual modules and of the user simulator as a whole and provide evidence towards the need for bimodal modeling.

5.1 Intrinsic Evaluation

As described previously, we select best models for RE and SR based on the accuracy on the validation set through an extensive hyperparameter search. Next, we evaluate their performance on two test sets: one containing 10% data for I-type users only and the other containing 10% data for P-type users only. It is evident from table 6 that RE is more accurate for the unimodal model on both the test sets, as measured in terms of Mean Squared Error (MSE). This indicates that greater training data for unimodal model helps the rapport estimation process, and that users' conversational goals do not have a significant bearing on rapport estimation. However, despite being trained on only 30% (I-Type) and 70% (P-Type) data, SR for bimodal model is either comparable or better than the unimodal model, as measured in terms of F1 score for SD and NONE conversational strategies. The F1 scores for all other conversational strategies are 0 due to extreme sparsity in comparison to SD and NONE. Further, SD only makes for 11.1% of all the predictions made by the unimodal model (rest all are NONE), as compared to 19.4% for bimodal model. This is further from the ground truth (25.6%) and indicates that the unimodal model severely underestimates other conversational strategies in favour of the NONE conversational strategy, which is the majority class. Note that this is despite resampling minority classes, and indicates a critical failure point of the unimodal model, especially for I-Type users. For example, in response to agent's (T^a = `introduce()`, C^a = NONE), the bimodal model with P-Type user outputs C^u = SD, while the unimodal model outputs C^u = NONE. Similarly, in response to agent's (T^a = `glad()`, C^a = NONE) (indicating that the agent is glad at receiving positive feedback), the bimodal model with I-Type user outputs C^u = (SD, PR), while the unimodal model outputs C^u = NONE.

5.2 Extrinsic Evaluation

We construct unimodal and bimodal user simulators by integrating the trained unimodal RE (since it is more accurate than bimodal RE) with unimodal or bimodal SR and acceptance models (section 4.6.2) and evaluate them using the following metrics:

- **Kullback-Leibler Distance ($\mathbf{D_{KL}}$):** We measure the KL distance between distributions of conversational strategies generated by the user simulator and real users. KL distance between two distributions P and Q is defined as the average of the KL divergences, i.e., $D_{KL}(P,Q) = \frac{d(P||Q)+d(Q||P)}{2}$, where $d(P||Q)$ is the KL divergence between P and Q.
- **Cramér–von Mises Divergence ($\mathbf{D_{CV}}$):** We measure the Cramér–von Mises divergence [36] between the real and simulated empirical distribution functions of rapport values at the time of agent's `request(feedback)` task strategy.

For both of the metrics above, the lower the value, the more realistic is the user simulator. In order to calculate these metrics, we also learnt a social reasoner for the agent using the same architecture as that for user SR. It is apparent from table 7 that bimodal model performs much better as compared to unimodal model both in terms of D_{KL} and D_{CV}. The differences in means of both the metrics are significant using two-sided Welch's t-test (D_{KL} : ($t = 433.2, p < 0.0001$), D_{CV} : ($t = 626.1, p < 0.0001$)). This provides additional evidence towards the importance of bimodal modeling for construction of a more realistic social user simulator.

Model Type		Rapport Estimator	Social Reasoner	
		MSE	SD	NONE
P-Type	Unimodal	**0.36**	0.32	0.84
	Bimodal	0.41	0.32	**0.85**
I-Type	Unimodal	**0.24**	0.15	**0.67**
	Bimodal	0.76	**0.40**	0.65

Table 6: Test set accuracy for various models

Model Type	D_{KL}	D_{CV}
Unimodal	0.246 ± 0.002	0.514 ± 0.001
Bimodal	$\mathbf{0.109 \pm 0.001}$	$\mathbf{0.316 \pm 0.002}$

Table 7: User Simulator performance

6 CONCLUSION AND DISCUSSION

In this paper, we proposed a novel architecture for a social user simulator that can be used to train RL-based socially-aware conversational agents. The proposed user simulator utilizes social cues such as user and system conversational strategies and user conversational goals to estimate the level of rapport during the interaction and to generate appropriate task and social behaviours. The architecture is general enough to be of use in different task settings. Further, we developed and evaluated a user simulator using the proposed architecture in a personal assistant domain.

Through our analyses we discovered that the different users of a conversational agent can be divided into two salient types depending upon their conversational goal (propositional vs interpersonal). This conversational goal (and hence, the user type) can be inferred from the conversational strategies employed by a given user during an interaction. This result should encourage researchers to pay closer attention to conversational strategies in discourse analysis. Further, we posit that users focused on propositional goals (P-Type) prefer more efficient interactions as compared to those focused on interpersonal (I-Type) goals. Thus, it is essential for a virtual agent to identify users' conversational strategies in order to infer their conversational goal and then adapt its task and social behaviour in order to maximize user satisfaction. Thus, there is a need to endow virtual agents with the ability to detect and generate appropriate conversational strategies in real time.

There are certain limitations of our work that we would like to address in the future. We did not account for potential errors due to bad speech recognition (ASR), or natural language understanding (NLU) in our modeling in order to model the dataset better, which was collected using a semi-autonomous setup and hence resulted in a few NLU or ASR errors. However, it would be important to incorporate these noise models in the user simulator in order to train robust RL-based conversational agents. Moreover, our user simulator only generates verbal social behaviour (conversational strategies) and doesn't generate other non-verbal (multimodal) behaviour (such as eye gaze, smile, etc.) which are important for building rapport. In order to accomplish this, we require accurate models for simultaneous generation of verbal and non-verbal social behaviour (NVB) that allow for NVB to cut across turns. This constitutes an important direction for future research. In this work, we also assumed that the conversational goal of the user remains fixed during the interaction, which might suffice for shorter interactions, but as a next step, it would be interesting to study and incorporate evolution of conversational goals during longer or multiple interactions for the purpose of developing dynamic social user simulators. We observed in our dataset that there were differences in relationship between rapport and task performance across the two clusters of users, however, they were not statistically significant. We believe that this aspect needs to be explored further through a careful experimental design and deserves more attention from the research community. Finally, we would like to use our user simulator to train an RL agent and test it with real users to further validate the proposed architecture.

ACKNOWLEDGMENTS

This work was supported in part by generous funding from Microsoft, LivePerson, Google, and the IT R&D program of MSIP/IITP [2017-0-00255, Autonomous Digital Companion Development]. We would also like to thank all the members of Articulab in Carnegie Mellon University for their useful feedback.

REFERENCES

[1] Nalini Ambady and Robert Rosenthal. 1992. Thin slices of expressive behaviour as predictors of interpersonal consequences: a meta-analysis. *Psychological Bulletin* 111, 2 (1992), 256–274.

[2] Timothy W Bickmore, Laura M Pfeifer Vardoulakis, and Daniel Schulman. 2013. Tinker: a relational agent museum guide. *Autonomous agents and multi-agent systems* 27, 2 (2013), 254–276.

[3] James H Bray, Scott E Maxwell, and Scott E Maxwell. 1985. *Multivariate analysis of variance*. Number 54. Sage.

[4] Justine Cassell and Timothy Bickmore. 2003. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User modeling and user-adapted interaction* 13, 1-2 (2003), 89–132.

[5] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. 2011. User simulation in dialogue systems using inverse reinforcement learning. In *Conference of the International Speech Communication Association*.

[6] Nancy L Collins and Lynn Carol Miller. 1994. Self-disclosure and liking: a meta-analytic review. *Psychological bulletin* 116, 3 (1994), 457.

[7] David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, et al. 2014. SimSensei Kiosk: A virtual human interviewer for healthcare decision support. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1061–1068.

[8] Aimee L Drolet and Michael W Morris. 2000. Rapport in conflict resolution: Accounting for how face-to-face contact fosters mutual cooperation in mixed-motive conflicts. *Journal of Experimental Social Psychology* 36, 1 (2000), 26–50.

[9] Emmanuel Ferreira and Fabrice Lefèvre. 2015. Reinforcement-learning based dialogue system for human–robot interactions with socially-inspired rewards. *Computer Speech and Language* 34, 1 (2015), 256–274.

[10] Norman M Fraser and G Nigel Gilbert. 1991. Simulating speech systems. *Computer Speech and Language* 5, 1 (1991), 81–99.

[11] Kallirroi Georgila, Maria K Wolters, and Johanna D Moore. 2010. Learning dialogue strategies from older and younger simulated users. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 103–106.

[12] Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. 2016. Affective personalization of a social robot tutor for children's second language skills. In *Proceedings of the Conference on Artificial Intelligence (AAAI-16)*. 3951–3957.

[13] Matt Stuttle Steve Young Jost Schatzmann, Karl Weilhammer. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *The Knowledge Engineering Review* 1, 24 (2006).

[14] Simon Keizer, Stéphane Rossignol, Senthilkumar Chandramohan, and Olivier Pietquin. 2012. User simulation in the development of statistical spoken dialogue systems. In *Data-Driven Methods for Adaptive Spoken Dialogue Systems*. 39–73.

[15] Iolanda Leite, André Pereira, Ginevra Castellano, Samuel Mascarenhas, Carlos Martinho, and Ana Paiva. 2011. Modelling Empathy in Social Robotic Companions. In *International Conference on User Modeling, Adaptation, and Personalization*. 135–147.

[16] Oliver Lemon and Olivier Pietquin. 2012. *Data-Driven Methods for Adaptive Spoken Dialogue Systems: Computational Learning for Conversational Interfaces*. Springer-Verlag New York.

[17] Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688* (2016).

[18] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.

[19] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. 3.

[20] Yoichi Matsuyama, Arjun Bhardwaj, Ran Zhao, Oscar Romeo, Sushma Akoju, and Justine Cassell. 2016. Socially-aware animated intelligent personal assistant agent. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 224–227.

[21] Florian Pecune, Jingya Chen, Yoichi Matsuyama, and Justine Cassell. 2018. Field Trial Analysis of Socially Aware Robot Assistant. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1241–1249.

[22] Olivier Pietquin and Helen Hastie. 2013. A survey on metrics for the evaluation of user simulations. *The Knowledge Engineering Review* 28, 1 (2013), 59–73.

[23] David V Pynadath and Stacy C Marsella. 2005. PsychSim: modeling theory of mind with decision-theoretic agents. In *Proceedings of the 19th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 1181–1186.

[24] Charles Rich and Candace L Sidner. 2012. Using collaborative discourse theory to partially automate dialogue tree authoring. In *International Conference on Intelligent Virtual Agents*. Springer, 327–340.

[25] Verena Rieser and Oliver Lemon. 2011. *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Springer-Verlag Berlin Heidelberg.

[26] Hannes Ritschel, Tobias Baur, and Elisabeth Andre. 2017. Adapting a Robot's Linguistic Style Based on Socially-Aware Reinforcement Learning.

[27] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[28] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *NAACL-Short '07 Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Association for Computational Linguistics, Stroudsburg, PA, USA, 149–152.

[29] Jost Schatzmann, Blaise Thomson, and Steve Young. 2007. Statistical user simulation with a hidden agenda. *Proc SIGDial, Antwerp* 273282, 9 (2007).

[30] Satinder P Singh, Michael J Kearns, Diane J Litman, and Marilyn A Walker. 2000. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*. 956–962.

[31] Tanmay Sinha and Justine Cassell. 2015. We click, we align, we learn: Impact of influence and convergence processes on student learning and rapport building. In *Proceedings of the 1st Workshop on Modeling INTERPERsonal SynchrONy And infLuence*. ACM, 13–20.

[32] Helen Spencer-Oatey. 2005. (Im) Politeness, face and perceptions of rapport: unpackaging their bases and interrelationships.

[33] Karen Tracy and Nikolas Coupland. 1990. Multiple goals in discourse: An overview of issues. *Journal of Language and Social Psychology* 9, 1–2 (1990), 1–13.

[34] Bernard L Welch. 1947. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika* 34, 1/2 (1947), 28–35.

[35] Yorick Wilks. 2010. *Close engagements with artificial companions: key social, psychological, ethical and design issues*. Vol. 8. John Benjamins Publishing.

[36] Jason D Williams. 2008. Evaluating user simulations with the Cramér–von Mises divergence. *Speech communication* 50, 10 (2008), 829–846.

[37] Steve Young. 2010. Still talking to machines (cognitively speaking). In *Eleventh Annual Conference of the International Speech Communication Association*.

[38] Zhou Yu, Alan W Black, and Alexander I. Rudnicky. 2017. Learning Conversational Systems that Interleave Task and Non-Task Content. In *International Joint Conference on Artificial Intelligence*. arXiv:1703.00099.

[39] Ran Zhao, Alexandros Papangelis, and Justine Cassell. 2014. Towards a dyadic computational model of rapport management for human-virtual agent interaction. In *International Conference on Intelligent Virtual Agents*. Springer, 514–527.

[40] Ran Zhao, Tanmay Sinha, Alan W Black, and Justine Cassell. 2016. Socially-Aware Virtual Agents: Automatically Assessing Dyadic Rapport from Temporal Patterns of Behavior. In *International Conference on Intelligent Virtual Agents*. 218–233.