Training and evaluation of an MDP model for social multi-user human-robot interaction

Simon Keizer, Mary Ellen Foster, Oliver Lemon
Interaction Lab
Heriot-Watt University
Edinburgh (UK)
{s.keizer,m.e.foster,o.lemon}@hw.ac.uk

Andre Gaschler, Manuel Giuliani
fortiss GmbH
Munich (Germany)
{gaschler,giuliani}@fortiss.org

Abstract

This paper describes a new approach to automatic learning of strategies for social multi-user human-robot interaction. Using the example of a robot bartender that tracks multiple customers, takes their orders, and serves drinks, we propose a model consisting of a Social State Recogniser (SSR) which processes audio-visual input and maintains a model of the social state, together with a Social Skills Executor (SSE) which takes social state updates from the SSR as input and generates robot responses as output. The SSE is modelled as two connected Markov Decision Processes (MDPs) with action selection policies that are jointly optimised in interaction with a Multi-User Simulation Environment (MUSE). The SSR and SSE have been integrated in the robot bartender system and evaluated with human users in hand-coded and trained SSE policy variants. The results indicate that the trained policy outperformed the hand-coded policy in terms of both subjective (+18%) and objective (+10.5%) task success.

1 Introduction

As the use of robot technology in the home as well as in public spaces is increasingly gaining attention, the need for effective and robust models for natural and social human robot interaction becomes more important. Whether it involves robot companions (Vardoulakis et al., 2012), game-playing robots (Klotz et al., 2011; Brooks et al., 2012; Cuayáhuitl and Kruijff-Korbayová, 2012), or robots that help people with exercising (Fasola and Mataric, 2013), human users should be able to interact with such service robots in an effective and natural way, using speech as well as other modalities of communication. Furthermore, with the emergence of new application domains there is a particular need for methods that enable rapid development of models for such new domains. In this respect, data-driven approaches are appealing for their capability to automatically exploit empirical data to arrive at realistic and effective models for interpreting user behaviour, as well as to learn strategies for effective system behaviour.

In spoken dialogue systems research, statistical methods for spoken language understanding, dialogue management, and natural language generation have proven to be feasible for effective and robust interactive systems (Rieser and Lemon, 2011; Lemon and Pietquin, 2012; Young et al., 2010; Young et al., 2013). Although such methods have recently also been applied to (multi-modal) human-robot interaction (Stiefelhagen et al., 2007; Cuayáhuitl et al., 2012), work on *multi-user* human-robot interaction has been limited to non-statistical, hand-coded models (Klotz et al., 2011).

On the other hand, substantial work has been done in the field of situated multi-party interaction in general, including data-driven approaches. In particular, Bohus & Horvitz (2009) have addressed the task of recognising engagement intentions using online learning in the setting of a screen-based embodied virtual receptionist, and have also worked on multi-party turn-taking in this context (Bohus and Horvitz, 2011).

In this paper we describe a statistical approach to automatic learning of strategies for selecting effective as well as socially appropriate robot actions in a multi-user context. The approach has been developed using the example of a robot bartender (see Figure 1) that tracks multiple customers, takes their orders, and serves drinks. We propose a model consisting of a Social State Recogniser (SSR) which processes audio-visual input and maintains a model of the social state, and a Social Skills Executor (SSE) which takes social state updates from the SSR as input and generates robot responses as out-

put. The SSE is modelled as a hierarchy of two connected Markov Decision Processes (MDPs) with action selection policies that are jointly optimised in interaction with a Multi-User Simulation Environment (MUSE).

Figure 1: The robot bartender with two customers

In the remainder of this paper we will describe the robot system in more detail (Section 2), followed by descriptions of the SSR (Section 3), the SSE (Section 4), and MUSE (Section 5). In Section 6 we then discuss in more detail the MDP model for the SSE and the process of jointly optimising the policies, and present evaluation results on simulated data. Next, we present results of the first evaluation of the integrated SSE-MDP component with human users (Section 7). The paper is concluded in Section 8.

2 Robot bartender system

The robot system we used for evaluating the models is equipped with vision and speech input processing modules, as well as modules controlling two robot arms and a talking head. Based on observations about the users in the scene and their behaviour, the system must maintain a model of the social context, and decide on effective and socially appropriate responses in that context. Such a system must be able to engage in, maintain, and close interactions with users, take a user's order by means of a spoken conversation, and serve their drinks. The overall aim is to generate interactive behaviour that is both task- effective and socially appropriate: in addition to efficiently taking orders and serving drinks, the system should, e.g., deal with customers on a first-come, first-served basis, and should manage the customers' patience by asking them politely to wait until the robot is done serving another customer.

As shown in Figure 1, the robot hardware consists of a pair of manipulator arms with grippers, mounted to resemble human arms, along with an animatronic talking head capable of producing facial expressions, rigid head motion, and lip-synchronised synthesised speech. The input sensors include a vision system which tracks the location, facial expressions, gaze behaviour, and body language of all people in the scene in real time (Pateraki et al., 2013), along with a linguistic processing system (Petrick et al., 2012) combining a speech recogniser with a natural-language parser to create symbolic representations of the speech produced by all users. More details of the architecture and components are provided in (Foster et al., 2012). An alternative embodiment of the system is also available on the NAO platform.

3 Social State Recogniser

The primary role of the Social State Recogniser (SSR) is to turn the continuous stream of messages produced by the low-level input and output components of the system into a discrete representation of the world, the robot, and all entities in the scene, integrating social, interaction-based, and task-based properties. The state is modelled as a set of *relations* such as `facePos(A)=(x,y,z)` or `closeToBar(A)`; see (Petrick and Foster, 2013) for details on the representation used.

In addition to storing all of the low-level sensor information, the SSR also infers additional relations that are not directly reported by the sensors. For example, it fuses information from vision and speech to determine which user should be assigned to a recognised spoken contribution. It also provides a constant estimate of whether each customer is currently seeking attention from the bartender (`seeksAttention(A)`): the initial version of this estimator used a hand-coded rule based on the observation of human behaviour in real bars (Huth et al., 2012), while a later version (Foster, 2013) makes use of a supervised learning classifier trained on labelled recordings of humans interacting with the first version of the robot bartender.

The SSR provides a query interface to allow other system components access to the relations stored in the state, and also publishes an updated state to the SSE every time there is a change which might require a system action in response (e.g., a customer appears, begins seeking attention, or makes a drink order).

4 Social Skills Executor

The Social Skills Executor (SSE) controls the behaviour of the robot system, based on the social state updates it receives from the SSR. The output of the SSE consists of a combination of non-communicative robot actions and/or communicative actions with descriptions of their multi-modal realisations. In the bartender domain, the non-communicative actions typically involve serving a specific drink to a specific user, whereas the communicative actions have the form of dialogue acts (Bunt et al., 2010), directed at a specific user, e.g. `setQuestion(drink)` ("What would you like to drink?") or `initialGreeting()` ("Hello").

In our design of the SSE, the decision making process resulting in such outputs (including the 'no action' output) consists of three stages: 1) ***social multi-user coordination***: managing the system's engagement with the users present in the scene (e.g., accept a user's bid for attention, or proceed with an engaged user), 2) ***single-user interaction***: if proceeding with an engaged user, generating a high-level response to that user, in the form of a communicative act or physical action (e.g., greeting the user or serving him a drink), and 3) ***multi-modal fission***: selecting a combination of modalities for realising a chosen response (e.g., a greeting can be realised through speech and/or a nodding gesture). One advantage of such a hierarchical design is that strategies for the different stages can be developed independently. Another is that it makes automatic policy optimisation more scalable.

5 Multi-User Simulated Environment

In order to test and evaluate the SSE, as well as to train SSE action selection policies, we developed a Multi-User Simulated Environment (MUSE). MUSE allows for rapidly exploring the large space of possible states in which the SSE must select actions. A reward function that incorporates individual rewards from all simulated users in the environment is used to encode preferred system behaviour in a principled way. A simulated user assigns a reward if they are served the correct drink, and gives penalties associated with their waiting time and various other forms of undesired system responses (see Section 6.1 for more details about the reward function). All of this provides a practical platform for evaluating different strategies for effective and socially appropriate behaviour. It also paves the way for automatic optimisation of policies, for example by using reinforcement learning techniques, as we will discuss in Section 6.1.

The simulated environment replaces the vision and speech processing modules in the actual robot bartender system, which means that it generates 1) vision signals in every time-frame, and 2) speech processing results, corresponding to sequences of time-frames where a user spoke. The vision observations contain information about users that have been detected, where they are in the scene, whether they are speaking, and where their attention is directed to. Speech processing results are represented semantically, in the form of dialogue acts (e.g., `inform(drink=coke)`, "I would like a coke"). As described in Section 3, the SSR fuses the vision and speech input, for example to associate an incoming dialogue act with a particular user.

The simulated signals are the result of combining the output from the simulated users in the environment. Each simulated user is initialised with a random goal (in our domain a type of drink they want to order), enters the scene at some point, and starts bidding for attention at some point. Each simulated user also maintains a state and generates responses given that state. These responses include communicative actions directed at the bartender, which are translated into a multi-channel vision input stream processed by the SSR, and, in case the user realises the action through speech, a speech processing event after the user has finished speaking. Additionally, the simulated users start with a given *patience level*, which is reduced in every frame that the user is bidding for attention or being served by the system. If a user's patience has reduced to zero, s/he gives up and leaves the bar. However, it is increased by a given fixed amount when the system politely asks the user to wait, encoded as a `pausing` dialogue act. The behaviour of the simulated users is partly controlled by a set of probability distributions that allow for a certain degree of variation. These distributions have been informed by statistics derived from a corpus of human-human customer-bartender interactions (Huth et al., 2012).

In addition to information about the simulated users, MUSE also provides feedback about the execution of robot actions to the SSR, in particular the start and end of all robot speech and non-communicative robot actions. This type of information simulates the feedback that is also provided in the actual bartender system by the components that directly control the robot head and arms. Figure 2

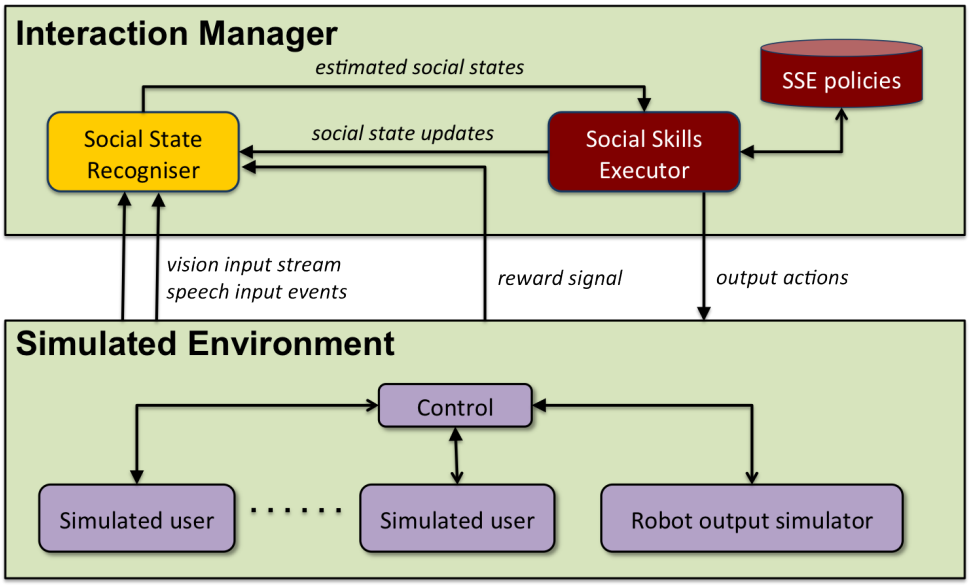

Figure 2: Social state recognition and social skills execution in a multi-user simulated environment.

shows the architecture of the system interacting with the simulated environment.

6 MDP model for multi-user interaction

To enable automatic optimisation of strategies for multi-user social interaction, the SSE model as described in Section 4 was cast as a hierarchy of two Markov Decision Processes (MDPs), corresponding to the *social multi-user coordination* and *single-user interaction* stages of decision making. Both MDPs have their own state spaces $\mathcal{S}_1$ and $\mathcal{S}_2$, each defined by a set of state features, extracted from the estimated social state made available by the SSR—see Tables 1 and 3. They also have their own action sets $\mathcal{A}_1$ and $\mathcal{A}_2$, corresponding to the range of decisions that can be made at the two stages (Tables 2 and 4), and two policies $\pi_1 : \mathcal{S}_1 \rightarrow \mathcal{A}_1$ and $\pi_2 : \mathcal{S}_2 \rightarrow \mathcal{A}_2$, mapping states to actions.

6.1 Policy optimisation

Using the MDP model as described above, we jointly optimise the two policies, based on the rewards received through the SSR from the simulated environment MUSE. Since MUSE gives rewards on a frame-by-frame basis, they are accumulated in the social state until the SSR publishes a state update. The SSE stores the accumulated reward together with the last state encountered and action taken in that state, after which that reward is reset in the social state. After each session (involving interactions with two users in our case), the set of encountered state-action pairs and associated rewards is used to update the policies.

The reward provided by MUSE in each frame is the sum of rewards R_i given by each individual simulated user i, and a number of general penalties arising from the environment as a whole. User rewards consist of a fixed reward in case their goal is satisfied (i.e., when they have been served the drink they wanted and ordered), a penalty in case they are still waiting to be served, a penalty in case they are engaged with the system but have not been served their drink yet, and additional penalties, for example when the system turns his attention to another user when the user is still talking to it, or when the system serves a drink before the user has ordered, or when the system serves another drink when the user already has been served their drink. General penalties are given for example when the system is talking while no users are present.

The policies are encoded as functions that assign a value to each state-action pair; these so-called *Q-values* are estimates of the long-term discounted cumulative reward. Given the current state, the policy selects the action with the highest Q-value:

$$\pi(s) = \arg\max_a \; Q(s,a) \qquad (1)$$

Using a Monte-Carlo Control algorithm (Sutton and Barto, 1998), the policies are optimised by running the SSR and SSE against MUSE and using the received reward signal to update the Q-values after each interaction sequence. During training, the SSE uses an ϵ-greedy policy, i.e., it takes a random exploration action with probability $\epsilon = 0.2$.

Index	Feature	Values
$4 \cdot i$	Interaction status for user $i + 1$	nonEngaged/seeksAttention/engaged
$4 \cdot i + 1$	Location of user $i + 1$	notPresent/!closeToBar/closeToBar
$4 \cdot i + 2$	User $i + 1$ was served a drink	no/yes
$4 \cdot i + 3$	User $i + 1$ asked to wait	no/yes

Table 1: State features for the *social multi-user coordination* policy. For each user, 4 features are included in the state space, resulting in $3^2 \cdot 2^2 = 36$ states for interactions with up to 1 user, increasing to 1296 states for interactions with up to 2 users and 46, 656 states for up to 3 users.

Index	Action
0	No action
$3 \cdot i + 1$	Ask user $i + 1$ to wait
$3 \cdot i + 2$	Accept bid for attention from user $i + 1$
$3 \cdot i + 3$	Proceed interaction with (engaged) user $i + 1$

Table 2: Actions for the *social multi-user coordination* policy.

In the policy update step, a discount factor $\gamma = 0.95$ is used, which controls the impact that rewards received later in a session have on the value of state-action pairs encountered earlier in that session.

Figure 3 shows the learning curve of a joint policy optimisation, showing average rewards obtained after running the SSE with trained policies for 500 runs, at several stages of the optimisation process (after every 2500 sessions/runs/iterations, the trained policy was saved for evaluation). In this particular setup, simulated users gave a reward of 550 upon goal completion but in the total score this is reduced considerably due to waiting time (-2 per frame), task completion time (-1 per frame) and various other potential penalties. Also indicated are the performance levels of two hand-coded SSE policies, one of which uses a strategy of asking a user to wait when already engaged with another user (labelled HDC), and one in which that second user is ignored until it is done with the engaged user (labelled HDCnp). The settings for user patience as discussed in Section 5 determine which of these policies works best; ideally these settings should be derived from data if available. Nevertheless, even with the hand-coded patience settings, the learning curve indicates that both policies are outperformed in simulation after 10k iterations, suggesting that the best strategy for managing user patience can be found automatically.

7 Human user evaluation

The SSE described above has been integrated in the full robot bartender system and evaluated for the first time with human users. In the experiment, both a hand-coded version and a trained version of the SSE component were tested; see Table 6 in Appendix A for the trajectory of state-action pairs of an example session. The hand-coded version uses the policy labelled HDC, not HDCnp (see Section 6.1). In each of the sessions carried out, one recruited subject and one confederate (one of the experimenters) approached the bartender together as clients and both tried to order a drink (coke or lemonade). After each interaction, the subject filled out the short questionnaire shown in Figure 4.

Q1: Did you successfully order a drink from the bartender? [Y/N]

Please state your opinion on the following statements:
[1:strongly disagree; 2:disagree; 3:slightly disagree; 4:slightly agree; 5:agree; 6:strongly agree]

Q2: It was easy to attract the bartender's attention [1–6]

Q3: The bartender understood me well [1–6]

Q4: The interaction with the bartender felt natural [1–6]

Q5: Overall, I was happy about the interaction [1–6]

Figure 4: Questionnaire from the user study.

37 subjects took part in this study, resulting in a total of 58 recorded drink-ordering interactions: 29 that used the hand-coded SSE for interaction management, and 29 that used the trained SSE.

The results from the experiment are summarised in Table 5. We analysed the results using a linear mixed model, treating the SSE policy as a fixed factor and the subject ID as a random factor. Overall, the pattern of the subjective scores suggests a slight preference for the trained SSE version, although

Index	Feature	Values
0	Reactive pressure	none/thanking/greeting/goodbye/apology
1	Status of user goal	unknown/usrInf/sysExpConf/sysImpConf/ grounded/drinkServed/sysAsked
2	Own proc. state	none/badASR

Table 3: State features for the *single-user interaction* policy. In this case, there are $5 \cdot 7 \cdot 2 = 70$ states.

Index	Action	Example
0	No action	
1	returnGreeting()	"Hello"
2	autoPositive()	"Okay"
3	acceptThanking()	"You're welcome"
4	autoNegative()	"What did you say?"
5	setQuestion(drink)	"What drink would you like?"
6	acceptRequest(drink=x) + serveDrink(x)	"Here's your coke"

Table 4: Actions for the *single-user interaction* policy, which correspond to possible dialogue acts, except for 'no action' and serving a drink. The specific drink types required for two of the actions are extracted from the fully specified user goal in the social state maintained by the SSR.

only the difference in perceived success was statistically significant at the $p < 0.05$ level. The actual success rate of the trained policy was also somewhat higher, although not significantly so. Also, the interactions with the trained SSE took slightly longer than the ones with the hand-coded SSE in terms of the number of system turns (i.e., the number of times the SSE receives a state update and selects a response action, excluding the times when it selects a non-action); however, this did not have any overall effect on the users' subjective ratings.

The higher success rate for the trained SSE could be partly explained by the fact that fewer ASR problems were encountered when using this version; however, since the SSE was not triggered when a turn was discarded due to low-confidence ASR, this would not have had an effect on the number of system turns. There was another difference between the hand-coded and trained policies that could have affected both the success rate and the number of system turns: for interactions in which a user has not ordered yet, nor been asked for their order, the hand-coded strategy randomly chooses between asking the user for their order and doing nothing, letting the user take the initiative to place the order, whereas the trained policy always asks the user for their order (this action has the highest Q-value, although in fact the value for doing nothing in such cases is also relatively high).

We also carried out a stepwise multiple linear regression on the data from the user experiment to determine which of the objective measures had the largest effect, as suggested by the PARADISE evaluation framework (Walker et al., 2000). The resulting regression functions are shown in Figure 5. In summary, all of the subjective responses were significantly affected by the objective task success (i.e., the number of drinks served); the number of low-ASR turns also affected most of the responses, while various measures of dialogue efficiency (such as the system response time and the time taken to serve drinks) also had a significant impact. In general, these regression functions explain between 15–25% of the variance in the subjective measures.

As an initial analysis of the validity of the simulated environment, we compared the state distribution of the simulated data accumulated during policy optimisation with that of the human user evaluation data. In terms of coverage, we found that only 46% of all states encountered in the real data were also encountered during training. However, many of these states do not occur very often and many of them do not require any action by the robot (a trained policy can easily be set to take no-action for unseen states). If we only include states that have been encountered at least 20 times, the coverage increases to over 70%. For states encountered at least 58 times, the coverage is 100%, though admittedly this covers only the 10 most frequently encountered states. The similarity of the two distributions can be quantified by computing the KL-divergence, but since such a number is

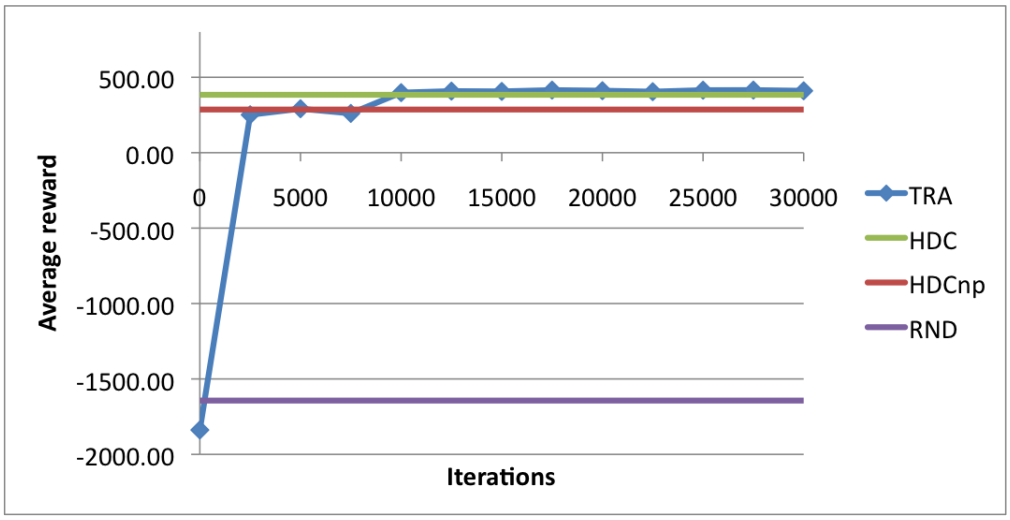

Figure 3: Learning curve for joint optimisation of SSE-MDP policies.

System	NS	PSucc*	PAtt	PUnd	PNat	POv	NDSrvd	NST	NBAsr
SSE-TRA	29	97%	4.10	4.21	3.00	3.83	1.97 (98.5%)	7.38	3.14
SSE-HDC	29	79%	4.14	3.83	2.93	3.83	1.76 (88.0%)	6.86	3.82
TOTAL	58	88%	4.12	4.02	2.97	3.83	1.86 (93.0%)	7.12	3.48

Table 5: Overview of system performance results from the experiment. In the leftmost column SSE-TRA and SSE-HDC refer to the trained and hand-coded SSE versions; the column NS indicates the number of sessions; the columns PSucc (perceived success), PAtt (perceived attention recognition), PUnd (perceived understanding), PNat (perceived naturalness), and POv (perceived overall performance) give average scores resulting from the 5 respective questionnaire questions; NDSrvd indicates the average number of drinks served per session (out of 2 maximum – the percentage is given in brackets); NST indicates the average number of system turns per session; while NBAsr indicates the average number of cases where the user speech was ignored because the ASR confidence was below a predefined threshold. The marked column indicates that the difference between the two SSE versions was significant at the $p < 0.05$ level.

hard to interpret in itself, this will only be useful if there were a state distribution from an alternative simulator or an improved version of MUSE for comparison.

8 Conclusion

In this paper we presented a new approach to automatic learning of strategies for social multi-user human-robot interaction, demonstrated using the example of a robot bartender that tracks multiple customers, takes their orders, and serves drinks. We presented a model consisting of a Social State Recogniser (SSR) which processes audio-visual input and maintains a model of the social state, and a Social Skills Executor (SSE) which takes social state updates from the SSR as input and generates robot responses as output. The main contribution of this work has been a new MDP-based model for the SSE, incorporating two connected MDPs with action selection policies that are jointly optimised in interaction with a Multi-User Simulation Environment (MUSE). In addition to showing promising evaluation results with simulated data, we also presented results from a first evaluation of the SSE component with human users. The experiments showed that the integrated SSE component worked quite well, and that the trained SSE-MDP achieved higher subjective and objective success rates (+18% and +10.5% respectively).

Our model currently only utilises two policies, but in more complex scenarios the task could be further modularised and extended by introducing more MDPs, for example for multimodal fission and natural language generation. The approach of using a hierarchy of MDPs has some similarity with the Hierarchical Reinforcement Learning (HRL) approach which uses a hierarchy of Semi-Markov Decision Processes (SMDPs). In (Cuayáhuitl et al.,

$$\text{PSucc} = 0.88 + 0.14 * \mathcal{N}(\text{NDSrvd}) - 0.07 * \mathcal{N}(\text{NBAsr}) \quad (r^2 = 0.21)$$

$$\text{PAtt} = 4.12 + 0.76 * \mathcal{N}(\text{NDSrvd}) - 0.46 * \mathcal{N}(\text{RTm}) - 0.38 * \mathcal{N}(\text{FDTm}) \quad (r^2 = 0.22)$$

$$\text{PUnd} = 4.02 + 0.41 * \mathcal{N}(\text{NDSrvd}) - 0.36 * \mathcal{N}(\text{NBAsr}) - 0.40 * \mathcal{N}(\text{NST}) - 0.41 * \mathcal{N}(\text{RTm}) - 0.39 * \mathcal{N}(\text{STm}) \quad (r^2 = 0.24)$$

$$\text{PNat} = 2.97 + 0.36 * \mathcal{N}(\text{NDSrvd}) - 0.29 * \mathcal{N}(\text{NBAsr}) - 0.31 * \mathcal{N}(\text{NST}) - 0.44 * \mathcal{N}(\text{RTm}) \quad (r^2 = 0.16)$$

$$\text{POv} = 3.83 + 0.65 * \mathcal{N}(\text{NDSrvd}) - 0.38 * \mathcal{N}(\text{NBAsr}) - 0.52 * \mathcal{N}(\text{RTm}) \quad (r^2 = 0.24)$$

Figure 5: PARADISE regression functions from the user study. The labels are the same as those in Table 5, with the following additions: RTm is the mean system response time per user, STm is the mean serving time per user, and FDTm is the mean time to serve the first drink; all times are measured in milliseconds. $\mathcal{N}$ represents a Z score normalisation function (Cohen, 1995).

2012) for example, this hierarchy is motivated by the identification of multiple tasks that the robot can carry out and for which multiple SMDP agents are defined. In every step of the interaction, control lies with a single SMDP agent somewhere in the hierarchy; once it arrives at its final state it returns control to its parent SMDP. An additional transition model is introduced to permit switching from an incomplete SMDP to another SMDP at the same level, making interactions more flexible. In our approach, control always starts at the top level MDP and lower level MDPs are triggered depending on the action taken by their parent MDP. For social interaction with multiple users, flexible switching between interactions with different users is important, so an arguably more sophisticated HRL approach to multi-user interaction will rely heavily on the transition model. Another approach to modularising the task domain through multiple policies is described in (Lison, 2011), where 'meta-control' of the policies relies on an activation vector. As in the HRL SMDP approach, this approach has not been applied in the context of multi-user interaction. In any case, a more thorough and possibly experimental analysis comparing our approach with these other approaches would be worth investigating.

In the future, we plan to extend our MDP model to a POMDP (Partially Observable MDP) model, taking uncertainty about both speech and visual input into account in the optimisation of SSE policies by incorporating alternative hypotheses and confidence scores provided by the input modules into the social state. Since hand-coding strategies becomes more challenging in the face of increased uncertainty due to noisy input, the appeal of automatic strategy learning in a POMDP framework becomes even stronger. In a previous offline version of our combined SSR and SSE, we have shown in preliminary simulation experiments that even in an MDP setting, an automatically trained SSE policy outperforms a hand-coded policy when noise is added to the speech channel (Keizer et al., 2013).

Another direction of research is to annotate the data collected in the described experiment for further analysis and use it to improve the features of the simulated environment. The improved models should lead to trained policies that perform better when evaluated again with human users. We will also make use of the findings of the PARADISE regression to fine-tune the reward function used for policy optimisation: note that two of the main features indicated by the PARADISE procedure—task success and dialogue efficiency—are already those included in the current reward function, and we will add a feature to account for the effects of ASR performance. We are also considering using collected data for direct supervised or off-policy reinforcement learning of SSE strategies.

Finally, we aim to extend our domain both in terms of interactive capabilities (e.g., handling communication problems, social obligations management, turn-taking) and task domain (e.g., handling more than the current maximum of 2 users, group orders, orders with multiple items). In order to make the (PO)MDP model more scalable and thus keeping the learning algorithms tractable, we also aim to incorporate techniques such as value function approximation into our model.

Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 270435, JAMES: Joint Action for Multimodal Embodied Social Systems, `http://james-project.eu/`. Thanks to Ingmar Kessler for help in running the user experiment.

References

Dan Bohus and Eric Horvitz. 2009. Learning to predict engagement with a spoken dialog system in open-world settings. In *Proceedings SIGdial*, London, UK.

Dan Bohus and Eric Horvitz. 2011. Multiparty turn taking in situated dialog: Study, lessons, and directions. In *Proceedings SIGdial*, Portland, OR.

A. Brooks, J. Gray, G. Hoffman, A. Lockerd, H. Lee, and C. Breazeal. 2012. Robot's play: Interactive games with sociable machines. *Computers in Entertainment*, 2(3).

H. Bunt, J. Alexandersson, J. Carletta, J.-W. Choe, A.C. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum. 2010. Towards an ISO standard for dialogue act annotation. In *Proceedings LREC*, Valletta, Malta.

Paul R. Cohen. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press, Boston.

Heriberto Cuayáhuitl and Ivana Kruijff-Korbayová. 2012. An interactive humanoid robot exhibiting flexible sub-dialogues. In *Proceedings NAACL HLT*, Montreal, Canada.

H. Cuayáhuitl, I. Kruijff-Korbayová, and N. Dethlefs. 2012. Hierarchical dialogue policy learning using flexible state transitions and linear function approximation. In *Proceedings COLING*, Mumbai, India.

Juan Fasola and Maja J. Mataric. 2013. A socially assistive robot exercise coach for the elderly. *Journal of Human Robot Interaction*, 2(3). To appear.

Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald P. A. Petrick. 2012. Two people walk into a bar: Dynamic multi-party social interaction with a robot agent. In *Proceedings ICMI*, Santa Monica, CA.

Mary Ellen Foster. 2013. How can I help you? Comparing engagement classification strategies for a robot bartender. Submitted.

K. Huth, S. Loth, and J.P. De Ruiter. 2012. Insights from the bar: A model of interaction. In *Proceedings of Formal and Computational Approaches to Multimodal Communication*.

Simon Keizer, Mary Ellen Foster, Zhuoran Wang, and Oliver Lemon. 2013. Machine learning of social states and skills for multi-party human-robot interaction. Submitted.

David Klotz, Johannes Wienke, Julia Peltason, Britta Wrede, Sebastian Wrede, Vasil Khalidov, and Jean-Marc Odobez. 2011. Engagement-based multiparty dialog with a humanoid robot. In *Proceedings SIGdial*, Portland, OR.

Oliver Lemon and Olivier Pietquin, editors. 2012. *Data-driven Methods for Adaptive Spoken Dialogue Systems: Computational Learning for Conversational Interfaces*. Springer.

Pierre Lison. 2011. Multi-policy dialogue management. In *Proceedings SIGdial*, Portland, OR.

Maria Pateraki, Markos Sigalas, Georgios Chliveros, and Panos Trahanias. 2013. Visual human-robot communication in social settings. In *the Workshop on Semantics, Identification and Control of Robot-Human-Environment Interaction, held within the IEEE International Conference on Robotics and Automation (ICRA)*.

Ronald P. A. Petrick and Mary Ellen Foster. 2013. Planning for social interaction in a robot bartender domain. In *Proceedings ICAPS*, Rome, Italy.

Ronald P. A. Petrick, Mary Ellen Foster, and Amy Isard. 2012. Social state recognition and knowledge-level planning for human-robot interaction in a bartender domain. In *AAAI 2012 Workshop on Grounding Language for Physical Systems*, Toronto, ON, Canada, July.

Verena Rieser and Oliver Lemon. 2011. *Reinforcement Learning for Adaptive Dialogue Systems*. Springer.

R. Stiefelhagen, H. Ekenel, C. Fügen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and A. Waibel. 2007. Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot. *IEEE Transactions on Robotics*, 23(5):840–851.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

L. Pfeifer Vardoulakis, L. Ring, B. Barry, C. Sidner, and T. Bickmore. 2012. Designing relational agents as long term social companions for older adults. In *Proceedings IVA*, Santa Cruz, CA.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3–4):363–377.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Blaise Thomson, and Kai Yu. 2010. The Hidden Information State model: a practical framework for POMDP based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.

Steve Young, M. Gašić, B. Thomson, and J. Williams. 2013. POMDP-based statistical spoken dialogue systems: a review. *Proceedings of the IEEE*. To appear.

Appendix A: Example session with two guests ordering a drink

Timestamp	Level 1 MDP		Level 2 MDP		Description
	State features	Action	State features	Action	
13:28:45:966	0 1 0 0 0 0 0 0	0	-	-	A1 visible, but not close to bar; no response generated yet.
13:28:48:029	1 2 0 0 0 0 0 0	2	-	-	A1 not close to bar and seeking attention: BT acknowledges this and engages with A1.
13:28:53:680	3 2 0 0 1 2 0 0	4	-	-	A2 visible, close to the bar, and seeking attention; BT is already engaged with A1 and therefore asks A2 to wait.
13:28:55:715	3 2 0 0 1 2 0 1	3	0 0 0	1	BT continues his interaction with A1 and asks for their order.
13:28:56:928	3 2 0 0 1 2 0 1	3	0 6 0	0	BT continues with A1 and waits for them to order.
13:28:56:928	3 2 0 0 1 2 0 1	3	0 6 0	0	Same as above: BT still waiting for A1's order.
...					Due to repeated ASR failures, this state action pair is encountered several times.
13:29:52:066	3 2 0 0 1 2 0 1	3	0 1 0	2	A1's has now been successfully recognised; BT serves the ordered drink to A1.
13:30:12:013	3 2 1 0 1 2 0 1	5	-	-	A2 still seeking attention; BT can now acknowledge this and engage with A1.
13:30:13:307	1 2 1 0 3 2 0 1	6	0 0 0	1	BT continues with A2 and asks for their order.
13:30:14:475	1 2 1 0 3 2 0 0	6	0 6 0	0	BT continues with A2 and waits for them to order
13:30:17:737	1 2 1 0 3 2 0 0	6	0 1 0	2	A2's recognised; BT serves ordered drink to A2.
13:30:37:623	1 2 1 0 3 2 1 0	0	-	-	Both A1 and A2 have been served; BT does nothing
13:30:41:440	1 2 1 0 3 2 1 0	0	-	-	Same as above.
...					

Table 6: SSE-MDP trajectory for one session from the evaluation data, showing the states and response actions taken for both MDPs. The states are represented via their value indices, corresponding to Tables 1 and 3; the action indices similarly correspond to the actions in Tables 2 and 4. In the descriptions, A1 and A2 refer to the first and second user detected; BT refers to the bartender.