Proceedings of the RaPID-4 @LREC 2022, pages 86–93
Marseille, 25 June 2022

A Database of Multimodal Data to Construct a Simulated Dialogue Partner with Varying Degrees of Cognitive Health

Ruihao Pan*, Ziming Liu, Fengpei Yuan**, Maryam Zare*,**
Xiaopeng Zhao, Rebecca J. Passonneau***
*Penn State University, **University of Tennessee
State College, PA, **Knoxville, TN
{rvp5555, muz50, rjp49}@psu.edu, **{zliu68, fyuan6}@vols.utk.edu,
**xzhao9@utk.edu

Abstract

An assistive robot that could communicate with dementia patients would have great social benefit. An assistive robot Pepper has been designed to administer Referential Communication Tasks (RCTs) to human subjects without dementia as a step towards an agent to administer RCTs to dementia patients, potentially for earlier diagnosis. Currently, Pepper follows a rigid RCT script, which affects the user experience. We aim to replace Pepper's RCT script with a dialogue management approach, to generate more natural interactions with RCT subjects. A Partially Observable Markov Decision Process (POMDP) dialogue policy will be trained using reinforcement learning, using simulated dialogue partners. This paper describes two RCT datasets and a methodology for their use in creating a database that the simulators can access for training the POMDP policies.

Keywords: dementia care, referential communication task, dialogue data

1. Introduction

An assistive robot for dementia care that could communicate with dementia patients would have great social benefit, given the high incidence of Alzheimer's disease and similar kinds of cognitive decline in the elderly (AA, 2020), in combination with the scarcity of caregivers to provide one-on-one companionship and assistance (GCOA, 2021). The ultimate goal of our work is to develop a Partially Observable Markov Decision Process (POMDP) policy for an artificial agent to engage in dialogues with elderly patients at different stages of cognitive decline, to provide assistance, companionship or facilitate early detection. As an initial step towards our larger goal, we aim to develop a POMDP policy that can engage in Referential Communication Tasks (RCTs; see below) with Alzheimer's patients. The POMDP dialogue policy will be trained using Reinforcement learning (RL), which requires many thousands of training episodes (trial dialogues). RL of policies for dialogue systems, as well as for robotics and other applications, typically utilizes simulators in place of interactions with the real world. This paper describes two datasets we will harvest to populate a simulator database for training a variety of RCT dialogue policies.

Referential Communication Tasks (RCTs), which have many applications, pertain to referential skills, meaning the way people introduce and refer back to concrete or abstract objects, and the way they interpret others' referring expressions. When humans engage in a dialogue, they can mention and then refer back to different people, objects, locations, plans, complex ideas, and so on. Referring expressions are the noun phrase descriptions, pronouns, and other linguistic devices we use to indicate what entities we are talking about. RCTs have been used to study how people choose referential expressions, e.g., for navigating a map (HCRC, 1993), or as part of studies of turn-taking behavior (Hirschberg et al., 2021), complex scene discrimination (Manuvinakurike et al., 2016), or to identify countries on the world map (Paetzel et al., 2020). RCTs have also been used to investigate communication impairments in children (Bishop and Adams, 1991) or dementia patients (Feyereisen et al., 2007). Typically, an RCT involves a visual stimulus that is fully visible only to one participant, who must describe it to the other dialogue participant in a way that leads to the identification of the correct object. RCTs is used to assess the ability to provide and understand specific information of both ordinary people and people with dementia in our experiment.

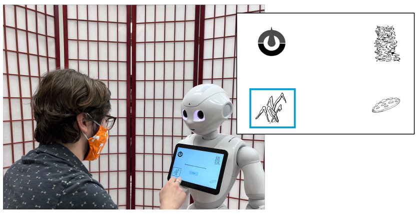
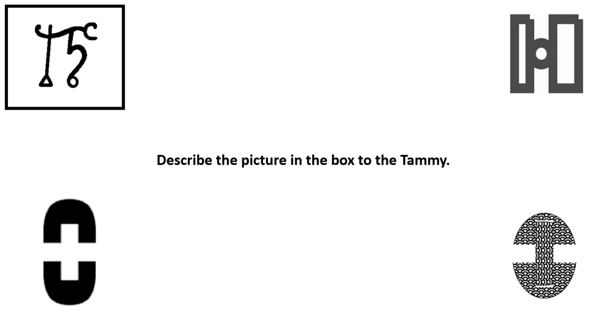


Figure 1: Pepper administering an RCT; inset shows Pepper's screen in detail. The subject describes one of the four unfamiliar images that appear in the four corners. Pepper must infer the correct image from the subject's verbal description. For the selected image, identified in the inset with a blue box around it, the subject said ”the picture looks like some bent over grass.”

To illustrate an RCT from one of our two datasets, Figure 1 shows a Pepper robot administering an RCT to a subject with no cognitive deficit. Pepper's screen presents four unfamiliar images to the subject (see inset), while Pepper instructs the subject to provide a ver-



Scripted Pepper

Pepper_1 *Pepper shows four images on its screen to the subject, one of which is boxed.*
Student_1: There are three lines, there is a triangle, a circle and a C at the end of the line.
Pepper_2: Give me a second to figure it out.
Pepper_3: Can you give me more details?
Student_3: It looks like fifteen.
Pepper_4: Give me a second to figure it out.
Pepper_5: Can you give me more details?
Student_5: One has a triangle at the bottom, five has a C at the top and a circle at the bottom.
Pepper_4: Give me a second to figure it out.
Pepper_5: I think I know the one you are trying to describe. I'm ready for the next one.

Future POMDP Version

Pepper_1 *Pepper shows four images on its screen to the subject, one of which is boxed.*
Student_1: There are three lines, there is a triangle, a circle and a C at the end of the line.
Pepper_2: Give me a second to figure it out.
Pepper_3: Does it have any other features?
Student_3: It looks like fifteen.
Pepper_4: It's between the two at the top.
Pepper_5: Can you describe it again?
Student_5: It almost looks like cave art of an animal, with one round hoof and one triangular hoof.
Pepper_4: Give me a second.
Pepper_5: Got it. Thanks for your patient explanation. Let us move on to the next one.

Figure 2: The current scripted version of Pepper administering an RCT, and the envisioned dialogue-enabled version of Pepper administering an RCT.

bal description of one of the four images. The main purpose of this first data collection is to study how use of Pepper to administer an RCT affects subjects' attitudes about robots and the RCT. For this data collection, Pepper followed a rigid script, as in Figure 2. In future work, we aim to carry out a similar RCT where we replace Pepper's script with a POMDP dialogue policy, for more natural interactions with subjects. This dataset and a second one are described in section 3. Briefly, the second dataset consists of RCTs administered by a human researcher to elderly patients, including patients with dementia. We will utilize these two datasets to construct a database of simulator turns-at-talk from three populations engaging in similar RCTs: young individuals with no known cognitive decline, elderly patients with no known dementia, and elderly patients with Alzheimer's.

POMDP dialogue policies can be used for dialogue agents where there is a defined goal to achieve during the dialogue, such as to complete an RCT interview, and where dialogue states are not fully observable. The interpretations of the dialogue partners' intents are only partially observable from the actual words used, and any other relevant behavior, given that human language is highly ambiguous. In reinforcement learning of a POMDP dialogue policy, a fully trained policy will choose each next communicative action a given its current belief state s, based on its expectation of how an action taken in a given state progresses the dialogue towards the agent's goal.

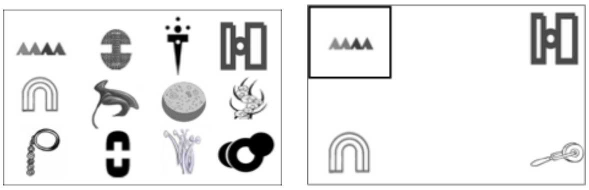
Construction of a simulator for reinforcement learning of a dialogue policy requires a method to sample different outcomes (successor states of s) for agent's communicative actions a taken in s. For example, to simulate the way subjects might respond to Pepper when Pepper displays the image shown in the inset of Figure 1, the initial state s would include a representation of the full display, a set of available actions for Pepper to choose among $\{a_1, \ldots, a_n\}$, and candidate simulator responses to each action. For example, assume we want to test the hypothesis that a policy could be learned for Pepper to respond to a dialogue partner who seems to experience a moment of confusion by selecting an encouragement utterance (e.g., ”`You seem a bit tired, let me know when you are ready for the next picture`”) instead of immediately moving to the next RCT item (e.g., ”`Okay, let's do the next picture`”). During the policy training, the simulator could be designed to choose between a relevant response, such as the one illustrated in Figure 1, or a response that suggests a moment of confusion, such as ”*I forgot what I'm supposed to say now.*” Our method for providing a simulated dialogue partner with this type of functionality involves creation of a database of response types where the values of the attributes of entries in a response table make it possible to control for different response types, during policy training.

The remainder of the paper presents related work, describes the two RCT datasets, and presents our methodology for constructing simulated dialogue partners so that we can train a range of RCT dialogue policies.

2. Related Work

Simulation has been utilized for training dialogue policies for well over two decades (Schatzmann et al., 2006) Eckert et al. (1997) proposed a statistical simulator permitting off-line testing and evaluation in an automated fashion. Scheffler and Young (2000) proposed a graph-based model which produce a probabilistic simulation of mixed initiative dialogue with recognition and understanding errors. Georgila et al. (2005) designed a Markov Model for use with Information State Update dialogue systems. Cuayáhuitl et al. (2005) used a network of hidden Markov models (HMMs) to predict system and dialogue partner intentions, where a statistical language model predicts sequences of goals, and the component HMMs predict sequences of intentions. For robustness to imperfect automatic speech recognition, Schatzmann et al. (2007b) simulated speech recognition errors at random levels, using generative models that conditioned words on the sets of dialogue actions expected from people, with conditioning probabilities estimated from corpora (Schatzmann et al., 2007b). Later work demonstrated bootstrapped policy learning in the absence of domain-specific corpora, using more complex simulators to maintain a dialogue state tuple across simulator turns, consisting of the dialogue goal plus a stack-based agenda to track progress towards the goal (Schatzmann et al., 2007a). Georgila et al. (2010) used simulated users to train dialogue policies for older adults, even though older adults have more complex and diverse interaction. Agenda-based simulators are still used, e.g., in the movie domain (Li et al., 2016). Asri et al. (2016) proposed a sequence-to-sequence model in the restaurant search domain which takes into account the entire dialogue history. Shah et al. (2018) used end-to-end neural models to build an agenda-based simulator. Kreyssig et al. (2018) introduced the Neural User Simulator which trains on corpora to learn how to generate natural language. Shi et al. (2019) developed a rule-based simulator in training reinforcement learning based dialog systems. An alternative to simulation has been explored in incremental dialogue policy learning in the context of fast-paced dialogue games (Manuvinakurike et al., 2017).

We have experience training an adaptive POMDP policy for learning through communication using a simulated dialogue partner that accesses a multi-modal database to look up answers to questions about games, including visual demonstrations of board moves or ways to win (Zare et al., 2022). We refer to this policy as 3GA because it has 3-way grounding in world knowledge, game knowledge, and the discourse context, and it is adaptive. Because the partners answer the agent's questions, rather than asking the agent for help to complete a task, there is no need for an agenda, or an HMM to predict the dialogue partner's intentions. During training, we controlled for the completeness of information in the simulator's answers, so that the trained policy could adapt to different individuals who provide more or less complete information.



(a) Sorting phase.

(b) Testing phase.

Figure 3: RCT Stimuli.

3. Two Datasets of RCT Interactions

The two datasets discussed here are from human-robot RCTs with student participants (HR_RCT_St), as illustrated in Figure 1, and from human-human RCTs with elderly patients (HH_RCT_EP). Initially, we will use the human-robot dataset (HR_RCT_St) to construct a simulator for training a dialogue policy $\pi_{\text{HRI\_RCT}}$ to invest the Pepper robot illustrated in Figure 1 with more natural dialogue capabilities, and eliminate the rigidity of a script. The HH_RCT_EP data dataset will be used to train a dialogue policy $\pi_{\text{HRI\_EP}}$ for interaction with elderly patients. The purpose of this policy is to adapt to the cognitive state of the patient, analogous to (Yuan et al., 2021a; Yuan et al., 2021b), as described in section 5. Based on insights from these initial policies, we will later train a dialogue policy $\pi_{\text{HRI\_RCT\_EP}}$ that can administer RCTs to elderly patients with and without dementia. In short, access to different RCT datasets provides us with the means to design and populate a database for multiple simulators.

The human-robot dataset (HR_RCT_St) was collected to assess how comfortable humans would be with a Pepper-based RCT. The data consists of 98 interactions between Pepper and human subjects, where each RCT interaction consisted of a sorting phase to help participants acclimate to Pepper and orient to the RCT, followed by a testing phase. Each interaction was approximately 15-20 minutes in length. The subjects' audio was recorded for each interaction. The audio has been transcribed using automatic speech recognition (ASR), and will later be manually corrected.[1]

During an initial sorting phase to familiarize the subjects with the task, Pepper's screen displayed 12 unfamiliar images to the subject, as in Figure 3a. Pep-

[1] The HR_RCT_St dataset can be made public once the transcriptions have been corrected.

per would describe one of the images for the subject to select. The subject had three chances to pick the correct image, where each next description from Pepper would have more detail, before Pepper would move on to the next image. During the testing phase it was Pepper's turn to guess a target image from four that would be displayed to the subject, as in Figure 3b, with the subject providing a spoken description. CLIP, a pretrained image captioning model (Radford et al., 2021), was used to compute probabilities for the four images on the screen, given the subject's description. If one image probability was sufficiently high, Pepper would instruct the subject to move to the next display. Otherwise, Pepper would prompt the subject to give more details. After three failed tries, Pepper would move to the next display. The testing phase had 24 trials, with different target images on each trial.

The human-human (HH_RCT_EP) dataset comprised manual transcripts from 12 older adults with mild-to-moderate AD and 16 cognitively healthy older adults (Liu et al., 2022).[2] The experiment also included a sorting phase and a testing phase. Each subject's interaction had an approximate duration of two hours. In the sorting phase, the subject was given a set of 12 abstract images (Figure 3a) in a random order and the experimenter was given the same 12 images in a certain order. The experimenter described each of the 12 images to the subject and the subject rearranged the image cards accordingly. The sorting task was repeated at least four rounds. If subjects made errors, they repeated the task up to nine rounds until successfully sorting the images without errors for two consecutive rounds. Two experimenters carried out the testing phase together. One was the same experimenter from the sorting phase (A), and the other one was a new experimenter (B). The two experimenters and the subject were all shown the same four images, three that had been included in the sorting phase and one new image. For the subject only, one of the four images was highlighted by a black box, as in Figure 3b. The subject was instructed to describe the target image to the knowledgeable experimenter (A) or the naive experimenter (B). The appointed experimenter marked the targeted image and the other experimenter proceeded to the next trial. The testing phase had 24 trials in a set: 12 trials referred to the familiar images and 12 trials referred to the unfamiliar new images.

4. POMDP Dialogue Policies

A Markov Decision Process (MDP) models an agent's step-by-step decision making for situations where each decision can have different outcomes with different probabilities. MDPs and their variants simplify the decision making task by adopting the Markov assumption that each action is conditioned only on the current state, not on any prior states. Formally, an MDP is a 5-tuple $< S, A, T, R, s_0 >$, where $\boldsymbol{S}$ is the set of states, $\boldsymbol{A}$ is the set of agent actions, $\boldsymbol{T}$ is the transition model consisting of a probability distribution over successor states $\boldsymbol{s_{i+1}}$ given an action $\boldsymbol{a_i}$ taken in $\boldsymbol{s_i}$, $\boldsymbol{R}$ is a reward function for the outcome of each action, and $\boldsymbol{s_0}$ is the initial state. An MDP dialogue agent's communicative actions are chosen by a trained policy π that maps dialogue states to optimal actions. In a Partially Observable MDP (POMDP), states are not fully observed. For dialogue policies, $\boldsymbol{S}$ consists of the agent's belief states that represent the agent's uncertain interpretations of a human dialogue partner's utterances, $\boldsymbol{A}$ represents communicative actions available to the agent, and the reward function $\boldsymbol{R}$ depends on the application. In our recent work, it is a trade-off between a small cost per turn and metrics that encourage the agent to achieve its dialogue goal, such as to learn a board game, using a measure of the inrease in the total game knowledge (Zare et al., 2022). The turn cost leads to policies where the agent ends the dialogue when the expected penalty outweighs the potential gain. This contrasts with a small turn reward used in (Manuvinakurike et al., 2017), where the goal was for the agent to find an image described by the user, and therefore to give the agent more time. For the RCT task, we will experiment with different reward functions, such as number of RCT steps completed, and possibly signs of fatigue from the human subject.

We apply Q-learning to learn the policy. The Bellman equation shown below illustrates that a Q function from a state s to the optimal action a sums over the cumulative reward of all possible outcomes s' of a, where the cumulative reward is a product of the probability of each outcome s' with the sum of the immediate reward R for that action, and the discounted Q function applied to each successive state, using the discount γ. The best action is the one with the maximum Q value.

$$Q(s,a) = \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma \max_{a'} Q(s',a')]$$

The role of a simulated dialogue partner in training a dialogue policy is to simulate a wide range of partner turns that might ever occur, so that during training, the policy can explore many possible communicative action choices, to learn a good Q function. The policy cannot be learned from static transcripts: any one transcript represents actual turn sequences that occurred, rather than all possible turn sequences that might occur. In contrast, simulator databases can be constructed by harvesting transcripts.

To illustrate using the π_{HRI_RCT} policy, the immediate reward for a communicative action a taken by Pepper in state s would be computed after the simulator responds to the agent, which in turn would contribute to the discounted cumulative sum for the entire dialogue. Say we assume that a human subject will tend towards

[2] The experimental protocol was approved by the Internal Review Board at UTK under the number: UTK IRB-21-06631-XM.

more helpful descriptions if Pepper thanks the subject each time a single description is sufficiently clear for the CLIP model to disambiguate, and if Pepper expresses confusion otherwise. At every turn exchange, the reward includes a small penalty to encourage efficiency. After Pepper picks a correct image, there would be also be a positive reward. The simulator can be used to train the policy when to use a ”thank you” communicative action versus a ”confusion” communicative action through thousands of trials that use the full array of images in the experiment. Note that simulator turns do not need to be identical to turns humans might take, or even realistic. Ai and Litman (2011) showed that given a simulator constrained to a range of behaviors, generating those behaviors randomly leads to better performance. Rather, they need representations that human turns might be mapped to, say by a natural language understanding module.

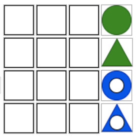
5. Simulated Dialogue Partners

The preceding sections have explained how a Pepper robot can administer an RCT (section 1), described two datasets of RCT sessions (section 3), and outlined Q-learning for POMDP dialogue policies to illustrate the need for simulating many trial dialogues (section 4). In section 2, we have also seen that a wide range of simulator architectures have been used, from those that maintain a stack-based agenda based on the simulator goal for task-oriented systems (Schatzmann et al., 2007a; Li et al., 2016), use of HMMs to predict sequences of dialogue-partner intentions, and turn-by-turn look-up of response sets for our 3GA agent that asks questions. Here we put it all together with a discussion of how to design and populate a database to simulate dialogue partners for the RCT tasks. We first present our previous work on a database for simulators to train an adaptive POMDP that can learn board games through multi-modal communication with people (Zare et al., 2022). Then we describe how we will construct an RCT database by analogy with this prior work.

5.1. The 3GA Simulator Database

We previously developed an adaptive POMDP dialogue policy called 3GA, for learning board games from people through multimodal communication. Figure 4 illustrates an excerpt of a 3GA dialogue to learn the Quarto board game, which is played on a 4-by-4 grid, using 24 pieces in two colors (12 each), differentiated by two heights, two shapes, and hollow or solid. The 3GA policy was trained on three n-in-a-row board games, so it could adapt to the game. It was also trained to adapt to how informative the dialogue partner's answers tended to be. A dialogue partner who responds to questions with *”I don't know”*, or with only partial information, has lower rates of information sharing. The fully trained 3GA policy would ask more open-ended questions (e.g., *”where else can I put this piece?*”) with partners who had high degrees of information sharing.

3GA-Q1 *RequestNewWinCondition()*
NL: Can you show me an example of a way to win?



A1
3GA-Q2 *SameContext(D_1), Confirm(translate(col0))*
NL: Would it still count if the pieces were placed down column one instead of column four?
A2 *Affirm()*
NL: Yeah, it would.

Figure 4: Excerpt of a dialogue where 3GA is learning Quarto, a 4-in-a-row game, showing the natural language (NL) below each MRL expression. These answers are produced by a simulator, but 3GA also communicates well with people, using a text-based interface.

With partners who shared less information, 3GA asked more yes-no questions, as in question 3GA-Q2 in Figure 4. Our experiments showed that the strategy of adapting to partner's information sharing led to more knowledge gain about a game (Zare et al., 2022).

The dialogue excerpt in Figure 4 shows a sequence of two questions from the policy (3GA-Q1 and 3GA-Q2) and the corresponding answers (A1 and A2), the first of which is a demonstration of a way to win the Quarto game. To win, four pieces in a row must share at least one property; here all pieces are tall, represented as solid colors rather than hashed. Apart from the first answer (A1), where the dialogue partner displays a game board to 3GA, each turn from 3GA or the dialogue partner is shown both in a meaning representation language (MRL) that we developed for communicating in unambiguous expressions about board games (e.g., `RequestNewWinCondition()`), and in natural language. For natural language understanding and generation, we trained encoder-decoder models on a corpus of dialogues where the MRL had English translations. We collected a corpus of 960 dialogues where students trained in our MRL added colloquial English translations of all MRL turns to simulated dialogues.

During training, the 3GA policy was exposed to any of the three games, and to different levels of information sharing.[3] A simulator would randomly select the game, and a level of information sharing. To answer questions generated by the policy, the simulator accessed a database. For the simulator to generate responses to the questions shown in Figure 4, it accessed a static database that stored all MRL questions associated with a given game, an exhaustive set of possible MRL answers to each question (including `Unknown`), and for each MRL answer, multiple possible translations of the

[3] We refer the reader to our previous work for further details.

Action	Script
`elicit_next_description`	Press record when you are ready.
`inform_processing_description`	Give me a second to figure it out.
`inform_understood`	I think I know the one you are trying to describe.
`inform_move_on`	Let us move on to the next one.
`request_more_detail`	Can you give me more details?
`end_session`	Thank you for completing the test!

Figure 5: Pepper's scripted communicative actions.

MRL into English text. Requests for demonstrations of game boards were indexed with images showing all possible game boards. In addition, the simulator accessed a dynamic database in which it stored answers it had used already within a given dialogue.

Initial versions of 3GA were MDP policies, in which the entire simulated dialogue would be carried out in MRL. To train POMDP policies, after the simulator accessed an MRL answer, it would also randomly select an English text version. To interpret the English answer, 3GA utilized an encoder-decoder natural language understanding module that produces a probability distribution over possible MRLs (see above). The MRL with the highest probability translation and its probability would then be used.

5.2. RCT Databases

The preceding section described a database we used for a simulator to train MDP and POMDP dialogue policies in which the dialogue policy goal was for the agent to learn board games from people. To train dialogue policies for our Pepper robot in RCTs with different subjects, we will construct an analogous database. As discussed in section 3, we aim to develop simulators to train RCT policies with different behaviors. Here we discuss the database formats required for simulators for two types of policies.

The current Pepper script for the first dataset described above (HR_RCT_St) has the six atomic actions shown in Figure 5. A POMDP dialogue policy to replace this script could be trained that could use a natural language generation sequence-to-sequence model, as in our previous work (Zare et al., 2022), to produce alternative verbalizations for the same dialogue action. The advantage of a policy instead of a script would be to extend the range of communicative actions, and the states in which they could be selected, so as to influence subjects to produce better descriptions. As noted in an earlier section, subjects could be thanked when the first description leads Pepper (via the CLIP model) to pick the correct image. Another way to influence subjects descriptions would be to replace the single action `request_more_detail` with a larger range of actions, given an initial description that is not understood, depending on different dialogue states, such as different probability distributions from CLIP over the four possibilities. If two images were equally probable, Pepper could say *That rules two out, but I'm still unsure.* During training, the input to the simulator would consist of a representation of the current state of the dialogue, and the dialogue action chosen by the policy. The database for generating the simulator responses would require tables for each image that contain alternative natural language descriptions harvested from the previous data collection. We would develop an automated procedure to sort each set of descriptions by various criteria, such as length in words and concreteness of the vocabulary, as well as ability of CLIP to discriminate the image from the various combinations of other images, so that the simulator could select new descriptions based on the dialogue state. To continue our example, if Pepper is confused between two images, one of which is the target image, the input to the simulator could include this information, and the simulator could be designed to produce a description that is highly ranked as a descriptor of the target and very improbable as a descriptor of the confounding image.

Difficulty	Example
Easy	Would you like some tea?
Moderate	What would you like to drink?
Difficult	What do you think about this tea?

Figure 6: Question difficulty

In our previous work to apply Q-learning for interactions in patients with dementia (Yuan et al., 2021a), we investigated a simulator to encourage a policy to adaptively respond to the simulator with easy, moderate or difficult questions, depending on different simulator settings to reflect different degrees of dementia. The simulator could be set to have different rates of producing relevant versus irrelevant versus non-responses to questions from the agent. We found that an adaptive policy could be trained to follow up with difficult, moderate and easy questions. Figure 6 illustrates three categories of question difficulty. *Yes/No* questions tend to be very specific, can be answered in the affirmative, negative, or unknown. The moderate question is open-ended, thus more difficult, but elicits a response that is very concrete. In contrast, the difficult open-ended question elicits an opinion that requires reflection and reasoning. The data we collected from the dementia patients in RCT tasks can be used test whether a similar policy could be trained that utilizes a database of actual responses to RCT questions from elderly and dementia patients, categorized by relevance, and other properties of the utterance, such as coherence.

6. Conclusion

The development of artificial agents to interact with dementia patients is a challenging task. Referential communication tasks (RCTs) have been used to assess dementia, but in practice, administering these such tasks is labor intensive. Our work addresses how to develop an agent that can administer RCT tasks to human subjects from youthful versus elderly populations, the latter includes individuals with dementia. Many techniques are available to create simulated dialogue partners that can be used to train MDP and POMDP dialogue policies. We have illustrated how two data from our RCT datasets could be used to train a range of dialogue policies to enhance an existing robot that administers RCTs using a scripted dialogue, and replace the script with more naturalistic RCT interviews. Our future work will test a range of simulators, investigate policy performance, and ultimately test the trained policies with humans.

7. Acknowledgements

This work was partly funded by the National Institute on Aging under the grant number R01AG077003. The experimental protocol was approved by the Internal Review Board at UTK under the number: UTK IRB-21-06631-XM."

8. Bibliographical References

AA. (2020). 2020 Alzheimer's disease facts and figures. *Alzheimer's and Dementia*, 16(3):391–481. Alzheimer's Association (AA).

Ai, H. and Litman, D. (2011). Comparing user simulations for dialogue strategy learning. *ACM Transactions on Speech and Language Processing*, 7(3):1–18, May.

Asri, L. E., He, J., and Suleman, K. (2016). A sequence-to-sequence model for user simulation in spoken dialogue systems. *arXiv preprint arXiv:1607.00070*.

Bishop, D. V. M. and Adams, C. (1991). What do referential communication tasks measure? A study of children with specific language impairment. *Applied Psycholinguistics*, 12(2):199–215.

Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. (2005). Human-computer dialogue simulation using hidden markov models. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 290–295. IEEE.

Eckert, W., Levin, E., and Pieraccini, R. (1997). User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87. IEEE.

Feyereisen, P., Berrewaerts, J., and Hupet, M. (2007). Pragmatic skills in the early stages of Alzheimer's disease: an analysis by means of a referential communication task. *International Journal of Language & Communication Disorders*, 42(1):1–17.

GCOA. (2021). Building the caregiving workforce our aging world needs. Global Coalition on Aging (GCOA).

Georgila, K., Henderson, J., and Lemon, O. (2005). Learning user simulations for information state update dialogue systems. In *Ninth European Conference on Speech Communication and Technology*.

Georgila, K., Wolters, M., and Moore, J. (2010). Learning dialogue strategies from older and younger simulated users. In *Proceedings of the SIGDIAL 2010 Conference*, pages 103–106, Tokyo, Japan, September. Association for Computational Linguistics.

HCRC. (1993). Map task corpus. LDC Catalogue number LDC93S12, University of Edinburgh, Human Communication Research Center (HCRC).

Hirschberg, J., Gravano, A., Benus, S., Ward, G., and German, E. S. (2021). Columbia games corpus. LDC Catalogue number LDC2021S02.

Kreyssig, F., Casanueva, I., Budzianowski, P., and Gasic, M. (2018). Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. *arXiv preprint arXiv:1805.06966*.

Li, X., Lipton, Z. C., Dhingra, B., Li, L., Gao, J., and Chen, Y.-N. (2016). A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Liu, Z., Paek, E. J., Yoon, S. O., Casenhiser, D., Zhou, W., and Zhao, X. (2022). Detecting Alzheimer's disease using natural language processing of referential communication task transcripts. *Journal of Alzheimer's Disease*, Preprint.

Manuvinakurike, R., Kennington, C., DeVault, D., and Schlangen, D. (2016). Real-time understanding of complex discriminative scene descriptions. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 232–241, Los Angeles, September. Association for Computational Linguistics.

Manuvinakurike, R., DeVault, D., and Georgila, K. (2017). Using reinforcement learning to model incrementality in a fast-paced dialogue game. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 331–341, Saarbrücken, Germany, August. Association for Computational Linguistics.

Paetzel, M., Karkada, D., and Manuvinakurike, R. (2020). RDG-map: A multimodal corpus of pedagogical human-agent spoken interactions. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 600–609, Marseille, France, May. European Language Resources Association.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Schatzmann, J., Weilhammer, K., Stuttle, M., and Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2):97–126.

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., and Young, S. (2007a). Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York, April. Association for Computational Linguistics.

Schatzmann, J., Thomson, B., and Young, S. (2007b). Error simulation for training statistical dialogue systems. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 526–531, The Westin Miyako Kyoto. IEEE.

Scheffler, K. and Young, S. (2000). Probabilistic simulation of human-machine dialogues. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 2, pages II1217–II1220. IEEE.

Shah, P., Hakkani-Tur, D., Liu, B., and Tür, G. (2018). Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51.

Shi, W., Qian, K., Wang, X., and Yu, Z. (2019). How to build user simulators to train RL-based dialog systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1990–2000, Hong Kong, China, November. Association for Computational Linguistics.

Yuan, F., Sadovnik, A., Zhang, R., Casenhiser, D., Paek, E. J., Yoon, S. O., and Zhao, X. (2021a). A simulated experiment to explore robotic dialogue strategies for people with dementia. *CoRR*, abs/2104.08940.

Yuan, F., Zhang, R., Bilal, D., and Zhao, X. (2021b). Learning-based strategy design for robot-assisted reminiscence therapy based on a developed model for people with dementia. In *International Conference on Social Robotics*, pages 432–442. Springer.

Zare, M., Wagner, A., and Passonneau, R. J. (2022). A POMDP dialogue policy with 3-way grounding and adaptive sensing for learning through communication. In *Submission*.