Using Reinforcement Learning to Optimize the Policies of an Intelligent Tutoring System for Interpersonal Skills Training

Socially Interactive Agents Track

Kallirroi Georgila
Institute for Creative Technologies
University of Southern California
kgeorgila@ict.usc.edu

Mark G. Core
Institute for Creative Technologies
University of Southern California
core@ict.usc.edu

Benjamin D. Nye
Institute for Creative Technologies
University of Southern California
nye@ict.usc.edu

Shamya Karumbaiah
Penn Center for Learning Analytics
University of Pennsylvania
shamya16@gmail.com

Daniel Auerbach
Institute for Creative Technologies
University of Southern California
auerbach@ict.usc.edu

Maya Ram
Institute for Creative Technologies
University of Southern California
mayaram@usc.edu

ABSTRACT

Reinforcement Learning (RL) has been applied successfully to Intelligent Tutoring Systems (ITSs) in a limited set of well-defined domains such as mathematics and physics. This work is unique in using a large state space and for applying RL to tutoring interpersonal skills. Interpersonal skills are increasingly recognized as critical to both social and economic development. In particular, this work enhances an ITS designed to teach basic counseling skills that can be applied to challenging issues such as sexual harassment and workplace conflict. An initial data collection was used to train RL policies for the ITS, and an evaluation with human participants compared a hand-crafted ITS which had been used for years with students (control) versus the new ITS guided by RL policies. The RL condition differed from the control condition most notably in the strikingly large quantity of guidance it provided to learners. Both systems were effective and there was an overall significant increase from pre- to post-test scores. Although learning gains did not differ significantly between conditions, learners had a significantly higher self-rating of confidence in the RL condition. Confidence and learning gains were both part of the reward function used to train the RL policies, and it could be the case that there was the most room for improvement in confidence, an important learner emotion. Thus, RL was successful in improving an ITS for teaching interpersonal skills without the need to prune the state space (as previously done).

KEYWORDS

Intelligent Tutoring Systems; Interpersonal Skills Training; Social Agents; Reinforcement Learning

ACM Reference Format:
Kallirroi Georgila, Mark G. Core, Benjamin D. Nye, Shamya Karumbaiah, Daniel Auerbach, and Maya Ram. 2019. Using Reinforcement Learning to Optimize the Policies of an Intelligent Tutoring System for Interpersonal Skills Training. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Intelligent Tutoring Systems (ITSs) have great potential to help learners who may have limited access to human teachers and tutors, but need help deciding what to study next, e.g., [11], or support when attempting to solve a problem, e.g., [5]. Such ITSs face a sequential decision-making task; at each time point, the ITS generally has a variety of possible actions and must choose which one to perform (e.g., do nothing, ask a question, or give a hint). Reinforcement Learning (RL) [16] is a popular machine learning approach to addressing this decision-making task as immediate and delayed rewards can be defined. RL attempts to generate a policy specifying what action to take in each possible system state to maximize rewards. Immediate rewards capture the immediate effectiveness of ITS actions on the learner (e.g., giving a hint may increase the probability that the learner answers a question correctly). Delayed rewards capture the effectiveness of ITS actions on longer-term measures (e.g., rewards based on test scores after tutoring). Including delayed rewards can help avoid a situation in which the learner succeeds during practice with help but is not able to succeed later without help (e.g., during a test).

A serious limitation of the current work on RL for ITSs is a focus on well-defined domains, e.g., physics [4], microbiology [15], computer databases [11]. In contrast, interpersonal skills are an example of an “ill-defined domain” [14] since the problem-solving (e.g., what is actually said) can be hard to support in the same way as problem-solving in a formalism such as mathematical equations or computer programs. In particular, learners view conversations as a real-time activity and although quick pop-up help can be given, it is not appropriate to stop the conversation when an error is made. Another difficulty is the necessity to give mixed feedback during practice. Since you cannot realistically solve someone’s problem with a single response, learners may struggle to identify when they made a good conversational move.

Despite the difficulty in teaching interpersonal skills, they are increasingly being recognized as critical to both social and economic development. Deming [8] focuses on the importance of social skills for the labor market, and notes that the U.S. labor market has been increasingly rewarding social skills in recent years. In fact, between 1980 and 2012, employment and wage growth was particularly strong for jobs requiring high levels of social skills together with

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

STEM-related skills (science, technology, engineering, and mathematics). Deming points out that the return of the labor market to social skills was more prominent in the 2000s, and that the reason for this shift in perspective is that social skills are difficult to automate [1]. Despite their importance, social and interpersonal skills are often left to be learned on the job or through unproven methods such as role-play with untrained partners, e.g., Hays et al. [10] discuss the need for better interpersonal skills training for junior officers in the U.S. Navy.

Core et al. [7] describes a system in which learners can practice interpersonal skills with virtual role-players who speak using prerecorded audio and act via 3D animations. Users select what to say to the role-player from a menu leading to different points in a branching graph representing the possible conversations. The built-in ITS monitors the interaction and can pop up a “coach” window with a feedback message (positive/negative comments on the previous choice) and/or a hint (indirect reference to the current correct choice). We focus on the version of the system called ELITE Lite Counseling which is designed to teach basic counseling skills to prospective U.S. Army officers. Such counseling skills can be used more generally by any supervisor dealing with the personal and performance problems of subordinates. Later versions of the system deal with the specific issue of preventing and responding to sexual harassment and assault in the workplace.

Given the importance of these interpersonal skills, it is crucial to optimize the ITS such that learners can succeed during their virtual role-play scenarios and more importantly learn the underlying principles for dealing with workplace problems. However, ELITE Lite Counseling, like many other training systems, uses heuristics to guide the ITS’s decision-making process rather than a data-driven approach. We modified the ELITE Lite Counseling system so that it can consult an RL service to make decisions. The modified system can be updated by replacing the data file containing the RL policy. The key question was whether a policy developed through RL could outperform the heuristics of the original system. Although researchers have had positive results in well-defined domains, it was not clear whether such results could be repeated in a domain where problem-solving corresponds to conversational actions.

We recruited human participants and tested the original heuristics-based version of ELITE Lite Counseling versus the modified version that employs RL. As described below, both versions of the system were effective, resulting in a significant increase from pre- to post-test scores. Although test scores and learning gains (a traditional ITS evaluation metric) did not differ significantly between conditions, learners had a significantly higher self-rating of confidence in the RL condition. We discuss the results in more detail below, but given that confidence was part of the RL reward function, and the importance of this learner emotion, we can say that RL was successful in improving our interpersonal skills training ITS.

This paper’s main contributions are two-fold: (1) To our knowledge this is the first time in the literature that RL has been applied to an ITS that teaches interpersonal skills, an “ill-defined domain”. (2) Unlike previous work on using RL for learning tutoring policies, we used Least-Squares Policy Iteration (LSPI) [12], a model-free sample efficient RL algorithm employing linear function approximation, which allowed us to learn successful policies without pruning the state space and without having to build a model of the environment (in our case, the environment is the learner).

The rest of the paper is structured as follows: First we present related work and then the learning environment (the Elite Lite Counseling ITS). After that we describe our experimental design and how we applied RL to our system, followed by our evaluation and results. Finally, we conclude with a summary of our findings and directions for future work.

2 BACKGROUND AND RELATED WORK

RL is used with the decision-making frameworks, Markov Decision Processes (MDPs) and Partially Observable Markov Decision Processes (POMDPs). In these frameworks, each distinct system state is a separate node in an inter-connected network. After taking an action, an agent will move from one state to another with a certain transition probability. Unlike MDPs, POMDPs do not make the assumption that the state is fully known which is beneficial when the state includes the learner’s mental state (e.g., knowledge of a topic). However, this increases the complexity of using RL since the system must track the multiple possible states. In general, tractability is an issue, and although some initial work with POMDPs has taken place, e.g., [3], we use MDPs, like the majority of work in this area, to keep training times manageable without oversimplifying the problem.

There are two approaches to using MDPs for RL, model-free and model-based. The difference is that model-based approaches attempt to estimate the transition probabilities of the MDP allowing for more directed learning whereas model-free methods take more of a trial-and-error approach and do not attempt to estimate the transition probabilities.

The trial-and-error learning of model-free approaches generally requires a large number of interactions during training to learn an effective decision-making policy. Since it is not feasible to have human learners interact with the ITS over thousands of trials, researchers instead create simulated learners using a relatively small amount of data or hand-crafted rules. Such a learner can tirelessly interact with the system during the RL training process. Beck et al. [2] describe a simulated learner built for an arithmetic ITS and used for training an RL policy with temporal-difference learning (specifically TD(0) with state-value learning) using a reward based on minimizing the time spent per problem. Given the possible differences between simulated learners and human learners, Beck et al. evaluated their RL policy against the default ITS with human learners, and verified that the policy did result in a significant decrease in time spent per problem. However, they did not test whether the RL policy achieved this result (i.e., efficient problem-solving within the ITS) at the cost of more shallow learning (e.g., lower performance in class tests).

Iglesias et al. [11] use a combination of a hand-crafted simulated learner and real learners to learn a policy for a computer database ITS. Iglesias et al. employ the Q-learning RL algorithm with a reward based on performance on tests given within the ITS. They find improvements in efficiency while evaluating with human users and comparing to the default ITS, and no significant differences between post-experience test results.

Other researchers use a model-based approach in which the transition probabilities of the MDP are estimated from a corpus of learners interacting with the target system. Given that the size of the MDP grows exponentially with the number of variables in the state, researchers highly constrain the state size to prevent training from becoming intractable. For example, Chi et al. [5] discuss the use of an MDP for Cordillera, a physics ITS which has 2 possible tutor actions, a state with 3 binary features, and a resulting 16 possible transitions.

Cordillera's policy was developed using a form of RL called policy iteration and was significantly better than a similar hand-crafted system in terms of a traditional ITS evaluation metric called learning gain [4]. Learning gain is calculated based on scores from tests given after tutoring (post-tests) and tests given before tutoring (pre-tests) which take into account prior knowledge and ability. Post-tests provide a measure of knowledge and/or performance in which the ITS is not allowed to give any support.

Crystal Island [15] is an ITS for microbiology in which problem-solving is situated in a virtual world. Behind the scenes, the ITS makes decisions such as parameters of the problem (e.g., what disease must be diagnosed) and hint-providing behavior of virtual characters in the world. Rowe and Lester [15] use a model-based RL technique called value iteration, and to keep the models to a reasonable size, the decision-making task is split into multiple MDPs each having 8 binary features (e.g., one of the MDPs models the setting of problem parameters). The RL policies were compared against random decision-making but no significant difference in learning gain was seen. Although learning gain has the advantage of being an independent measure, it has the disadvantage that the tests used may not align with material taught by the ITS. In the case of Crystal Island, Rowe and Lester found that learners in the RL condition scored significantly better on in-game problem-solving measures such as use of virtual laboratory tests, which is not something the post-test was designed to capture.

We choose a model-free approach to RL to avoid this need to prune information about the learner and the context (e.g., reaction times, number of questions attempted, scores achieved, difficulty of questions) to a handful of state features (e.g., 8 for Crystal Island) potentially omitting important information. However, we do not follow the standard practice of using simulated learners with model-free RL. Relatively simple simulated learners can mimic the behavior of real learners interacting with the ITS in terms of reaction time and problem-solving behavior, but unlike real learners cannot take tests or surveys (i.e., we would not know the learning gain or confidence gain of a simulated learner after a session). Instead, we take the novel approach of applying Least-Squares Policy Iteration (LSPI) [12] to learning policies for an ITS. During learning, LSPI allows us to estimate the expected reward at the current state given a corpus of interactions even with over 500 features.

We felt that the model-free approach of LSPI would give us the best chance to show that RL can be used successfully in an "ill-defined domain" (i.e., ELITE Lite Counseling). As described in [7], on the surface, ELITE Lite Counseling seems similar to well-defined domains; the ITS provides guidance on problem-solving steps and there are constraints such as ordering (e.g., do not respond with a course of action before checking for underlying causes). However, learners must make the connection between abstract conversational actions and constraints and actual dialogue (e.g., recognize that an emotional outburst calls for active listening, and identify the proper way to summarize what was said with neutral wording). Thus, although the general approach of using RL has been shown to be successful in well-defined domains, it was not clear that it would be successful here.

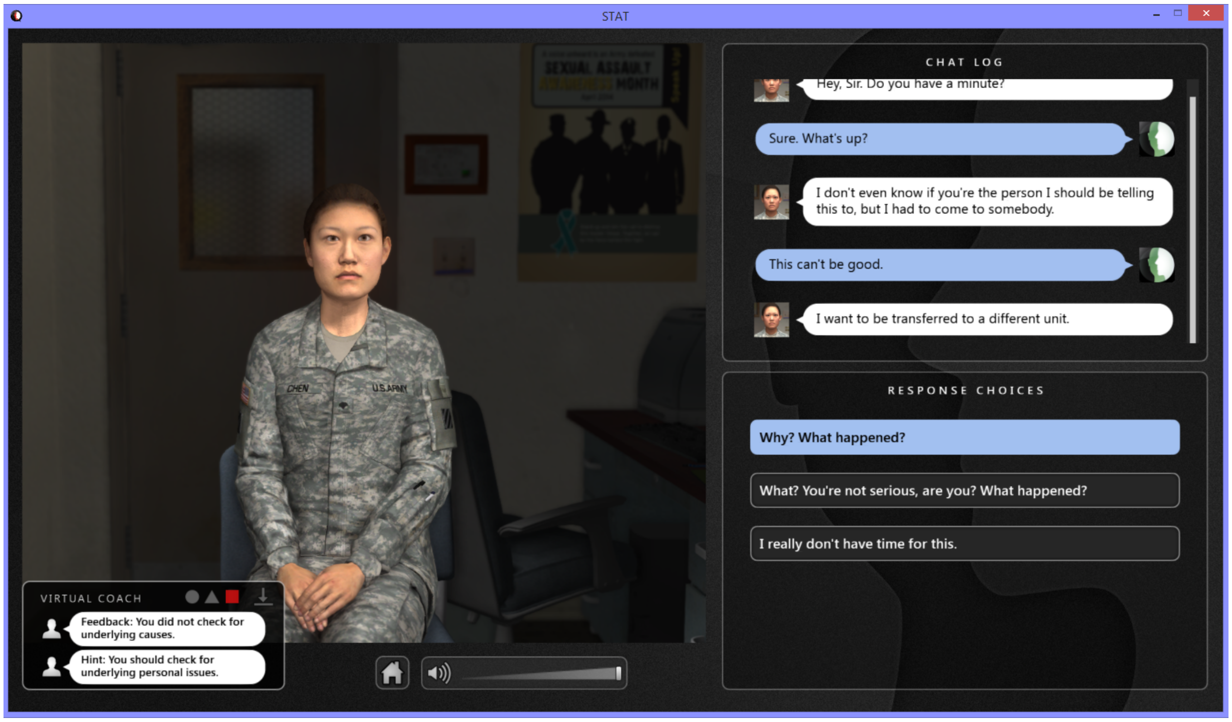
3 LEARNING ENVIRONMENT

Core et al. [7] introduces a system called ELITE Lite Counseling used by U.S. Army officers in training to learn leadership counseling skills, the interpersonal skills (e.g., active listening, checking for underlying causes, responding with a course of action) necessary to help subordinates with personal and performance problems. A feature of the system is the ability to interact with virtual subordinates who speak using pre-recorded audio and act via 3D animations (see Figure 1). Users select what to say to the subordinate from a menu leading to different points in a branching graph representing the possible conversations. Each choice can have both positive and negative annotations. Positive annotations correspond to correctly applying a skill such as active listening, and negative annotations correspond to omissions or misconceptions. Based on these annotations, a choice can be correct (only positive annotations), incorrect (only negative annotations), or mixed (both positive and negative annotations).

The ITS monitors the interaction and can pop up a "coach" window (lower left corner of Figure 1) with a feedback message (positive/negative comments on the previous choice corresponding to positive/negative annotations) and/or a hint (corresponding to a positive annotation of the correct choice in the menu). In the deployed system, this coach is governed by heuristics: give hint if last choice is not correct, give feedback if last choice is incorrect, or last choice is mixed and performance is less than a pre-set threshold. The coach also has icons that light up to indicate whether the last choice was correct, incorrect, or mixed, regardless of whether textual feedback is given.

Each simulated conversation is followed by a self-directed After Action Review (AAR) which allows learners to view all their decisions, the possible choices, and the underlying annotations. Core et al. [7] noted that the average number of clicks in the AAR was sometimes as low as 0.12 which indicates little if any use. To overcome this lack of engagement, we created a new AAR for our experiments in which every non-correct response was replayed and a multiple choice question asked (i.e., choose-again or identify-the-error). For each question, learners selected answers until they chose correctly.

We also modified ELITE Lite Counseling to use our RL service to make decisions instead of relying upon the heuristics described above. After each learner choice in the simulated conversation, ELITE Lite Counseling will ask the RL service whether it should give feedback on the choice, give a hint about the next choice to be made, give both feedback and a hint, or do nothing. During this process, the system actually makes two separate queries (hint or not; feedback or not), but all guidance is presented at the same time (see Figure 1). ELITE Lite Counseling also uses the RL service during the AAR to decide whether to ask a choose-again question or identify-the-error question for a particular mistake. Since the



Hey, Sir. Do you have a minute?

Sure. What's up?

I don't even know if you're the person I should be telling this to, but I had to come to somebody.

This can't be good.

I want to be transferred to a different unit.

Why? What happened?

What? You're not serious, are you? What happened?

I really don't have time for this.

Feedback: You did not check for underlying causes.

Hint: You should check for underlying personal issues.

Figure 1: ELITE Lite Scenario.

state in RL is a simplified version of the variables in the actual system, we modified ELITE Lite Counseling to send messages when important changes occurred so that the RL service could update its state representation. The state captured by the RL service is outlined later in the paper.

4 EXPERIMENTAL DESIGN

Two studies were conducted: a data collection to obtain training data for RL, and an evaluation comparing the RL policies against the heuristics in the original system (for when to give feedback and hints) and random choices for the AAR questions. In previous empirical work with the ELITE system, Hays et al. [10] developed survey questions for confidence and experience as well as a test for the target counseling skills. This test includes basic knowledge questions (i.e., true-false questions), and a situational judgment test (SJT) in which learners read problem descriptions and rank the appropriateness of different actions that could be taken. Given before and after practice with ELITE as a pre-test and post-test, it can be used as a measure of learning gain. The questions on confidence can also be asked before and after the experience to calculate a confidence gain. Core et al. [7] adapted this material to a non-military setting (i.e., a workplace environment) and added additional survey items to the pre- and post-experiment surveys.

To allow comparison with Core et al. [7], we used the same experimental design. In particular, Core et al. focused on two scenarios (i.e., simulated conversations) called Being Heard and Bearing Down. In Being Heard, a Soldier asks for a transfer but the root cause is sexual harassment, and in Bearing Down, a Soldier has grabbed and threatened another Soldier. Being Heard is used as a practice scenario; the coach may give hints and feedback, and participants play through the scenario twice to measure improvement. Bearing Down is used as an in-game assessment; the coach is deactivated including the correctness indicating icons. The full experimental procedure is: (1) pre-survey, followed by pre-test, (2) introductory video on leadership counseling skills, (3) first try of Being Heard followed by AAR, (4) second try of Being Heard followed by AAR, (5) complete Bearing Down with coach deactivated followed by AAR, (6) post-survey, followed by post-test.

Both studies used undergraduate participants from the University of Southern California. The first study was conducted during the fall 2016 and spring 2017 semesters, and the second study was conducted during the fall 2017 semester. Because the goal of the first study was data collection, the RL service made decisions randomly for both the hint/feedback decision and the AAR question-type decision. Because a random policy was used during data collection, during training, RL could explore different possible policies and not be limited by assumptions (e.g., the original ELITE Lite Counseling would never give feedback after a correct choice). Unfortunately due to software error, data from 9 participants was completely lost and for some participants no AAR information was recorded. The data from 93 participants was used as training data for the hint/feedback decision-making policy. The data from 72 participants contained AAR information and was used for training the AAR question-type policy. In the larger data set, there were 43 male participants and 50 female participants; the ratio for the smaller set was 35:37. The self-reported ethnicity of the 93 person sample was 47 Asian/Pacific Islander, 23 White, 13 Hispanic/Latino, 6 Black/African American, 4 Other/Unreported, and 0 Native American/American Indian with a similar pattern in the 72 person sample.

The second study compared the trained RL policy against a baseline (control) condition in which the heuristics from the deployed system were used to give hints and feedback, and task choices in the AAR were made randomly (i.e., identify-the-error versus choose-again). In the RL condition, the hints and feedback, and choices in the AAR were controlled by RL policies learned from the data collected in the first study. We alternated between these two conditions, assigning participants to one based on the order in which they arrived. Due to lost data and software crashes, we excluded 8 participants leaving 72 participants (35 in the baseline condition and 37 in the RL condition). There were 22 male participants and 50 female participants. The self-reported ethnicity was 40 Asian/Pacific Islander, 14 White, 6 Other/Unreported, 5 Hispanic/Latino, 5 Black/African American, and 2 Native American/American Indian.

5 REINFORCEMENT LEARNING FOR ELITE LITE

As discussed in the background and related work section, this work uses an MDP framework for RL. An MDP is defined as a tuple (S, A, P, R, γ) where S is the set of states that the agent may be in, A is the set of actions of the agent, $P : S \times A \rightarrow P(S, A)$ is the set of transition probabilities between states after taking an action, $R : S \times A \rightarrow \Re$ is the reward function, and $\gamma \in [0,1]$ a discount factor weighting long-term rewards. In our experiments, we set γ to 0.9 because of the importance of long-term rewards such as gains in learning and confidence.

RL seeks to learn a policy which given the state of the agent specifies the action to take to maximize rewards. At any given time step i the agent is in a state $s_i \in S$. When the agent performs an action $\alpha_i \in A$ following a policy $\pi : S \rightarrow A$, it receives a reward $r_i(s_i, \alpha_i) \in \Re$ and transitions to state $s_i^{'}$ according to $P(s_i^{'}|s_i, \alpha_i) \in P$. In tutoring applications, actions might include doing nothing, asking a question, or giving a hint, and states might include information such as the number of correctly answered questions.

For an RL-based agent the objective is to maximize the reward it receives during an interaction. Because it is very difficult for the agent, at any point in the interaction, to know what will happen in the rest of the interaction, the agent must select an action based on the average reward it has previously received after having performed that action in similar contexts. This average reward is called *expected future reward*, also called the Q-function, $Q^{\pi} : S \times A \rightarrow \Re$. The quality of the policy π being followed by the agent can always be measured by the Q-function.

There are several algorithms for estimating the Q-function. This work uses Least-Squares Policy Iteration (LSPI) [12]. LSPI can learn directly from a corpus of interactions and is sample efficient, i.e., it makes maximal use of data. It is also a model-free method, which does not require a model of the environment. In order to reduce the search space and make learning more efficient we use linear function approximation of the Q-function. Thus $Q(s, \alpha) = \sum_{i=1}^{k} w_i \phi_i(s, \alpha)$ where s is the state that the agent is in and α the action that it performs in this state, and $\tilde{w}$ is a vector of weights where w_i is the weight for the feature function $\phi_i(s, \alpha)$. The magnitude of a weight w_i is an indicator of the contribution of the feature $\phi_i(s, \alpha)$ to the $Q(s, \alpha)$ value. These feature functions can be specified manually or through feature selection algorithms [13]. We manually selected the features but also experimented with automatic feature selection.

LSPI is an iterative procedure starting with an arbitrary initial weight vector $\tilde{w}_1$ of dimension k which is iteratively improved until it converges to a near-optimal policy. It takes as input a set of m samples $D = \{(s_1, \alpha_1, r_1, s_1^{'}),...,(s_m, \alpha_m, r_m, s_m^{'})\}$ and works as follows:

At iteration $j = 1,2,...,$

- First step: Let the current linear Q-function be $Q_j(s, \alpha) = \tilde{w}_j^T \tilde{\phi}(s, \alpha)$ and the corresponding greedy policy be $\pi_j(s) = argmax_a Q_j(s, \alpha)$.
- Second step: Calculate $Q_{j+1}(s, \alpha) = \tilde{w}_{j+1}^T \tilde{\phi}(s, \alpha)$ where $\tilde{w}_{j+1}$ can be estimated from $\tilde{A}\tilde{w} = \tilde{c}$. $\tilde{A}$ is a $k \times k$ matrix and $\tilde{c}$ a k-vector and are computed as follows: $\tilde{A} = \sum_{l=1}^{m} \tilde{\phi}(s_l, \alpha_l) (\tilde{\phi}(s_l, \alpha_l) - \gamma\ \tilde{\phi}(s_l^{'}, \pi_j(s_l^{'})))^T$ and $\tilde{c} = \sum_{l=1}^{m} \tilde{\phi}(s_l, \alpha_l) r_l$.

The algorithm stops when the distance between the current feature weights and the feature weights computed in the previous iteration is less than a manually selected value ϵ. Note that LSPI does not require setting a learning rate like other RL algorithms such as Q-learning or SARSA.

We learned two RL policies, one hint/feedback decision-making policy and one AAR question-type policy. Tables 1 and 2 show the state variables that we track for the hint/feedback decision-making and AAR question-type policies respectively. The hint/feedback decision-making RL policy can choose from 3 actions: do nothing, provide hint, and provide feedback. The state variable "after user response" in Table 1 models whether the user has just made a choice: "no" = 0, and means a hint could be given but not feedback, while "yes" = 1, and means that feedback could be given but not a hint. Since the training corpus followed this pattern, RL picked up this relationship without the need for us to encode it explicitly. The AAR question-type RL policy can choose between 2 actions: choose-again and identify-the-error, and has no implicit constraints.

The features used in LSPI are combinations of all the possible values of the state variables and the actions. For each state variable one of the possible values is always the "null" (empty) value. The hint/feedback decision-making RL policy is used only for the Being Heard scenario (first and second try) whereas the AAR question-type RL policy is used for both the Being Heard scenario (first and second try) and the Bearing Down scenario. The state variables that are related to numbers of responses and scores, time, and question difficulty have been clustered into 6 classes including the null class in order to keep the number of features manageable. The 5 non-null classes for the numbers of responses and scores were defined as [0,1], (1,3], (3,6], (6,11] and (11,∞). The 5 non-null classes for time were defined as [0,3], (3,5], (5,10], (10,17] and (17,∞) (in seconds). We used the data from Core et al. [7] as a development set to calculate question difficulty. Based on this data we calculated the probability that a question would be correct, and defined 5 non-null classes of difficulty given the following probability ranges: [0,0.3], (0.3,0.6], (0.6,0.8], (0.8,0.9] and (0.9,1]. We decided not to use equal-size classes because the data was not uniformly distributed. For example, for time there were more data points between 0 and 5

State Variable	Number of Possible Values	State Variable	Number of Possible Values
scenario number	3	gender	3
# correct responses in prev scenario	6	# correct responses so far	6
# mixed responses in prev scenario	6	# mixed responses so far	6
# incorrect responses in prev scenario	6	# incorrect responses so far	6
# responses in prev scenario	6	# responses so far	6
score in prev scenario	6	score so far	6
avg correct response time in prev scenario	6	avg correct response time so far	6
avg mixed response time in prev scenario	6	avg mixed response time so far	6
avg incorrect response time in prev scenario	6	avg incorrect response time so far	6
avg response time in prev scenario	6	avg response time so far	6
response quality of prev question	4	response time of prev question	6
response quality of one before prev question	4	response time of one before prev question	6
response quality of two before prev question	4	response time of two before prev question	6
response quality if question has appeared before	4	question difficulty	6
response quality of prev question combined with after user response	8	has question appeared in prev scenario	2
after user response	2	is this the final state	2

Table 1: State variables considered for the hint/feedback decision-making RL policy.

State Variable	Number of Possible Values	State Variable	Number of Possible Values
scenario number	4	gender	3
# correct responses in prev scenario	6	# correct responses so far	6
# mixed responses in prev scenario	6	# mixed responses so far	6
# incorrect responses in prev scenario	6	# incorrect responses so far	6
# responses in prev scenario	6	# responses so far	6
score in prev scenario	6	score so far	6
avg correct response time in prev scenario	6	avg correct response time so far	6
avg mixed response time in prev scenario	6	avg mixed response time so far	6
avg incorrect response time in prev scenario	6	avg incorrect response time so far	6
avg response time in prev scenario	6	avg response time so far	6
response quality of prev question	4	response time of prev question	6
response quality if question has appeared before	4	question difficulty	6
# choose-again so far	6	has question appeared in prev scenario	2
# identify-the-error so far	6	is this the final state	2

Table 2: State variables considered for the AAR question-type RL policy.

seconds than between 5 and 10 seconds, and for question difficulty it was more likely to have a correctness probability over 0.6 than below 0.6. For the hint/feedback decision-making RL policy we ended up with 168 states × 3 actions = 504 features, and for the AAR question-type RL policy with 151 states × 2 actions = 302 features.

For both policies a reward is applied at the end of each scenario and is a combination of the unnormalized learning gain (see the results section), the final score of the previous scenario (if we are in the second try of Being Heard or in Bearing Down), the final score of the current scenario, and the confidence gain. We used equal weights for each of the above 4 factors. We experimented with a variety of different weights but these settings resulted in policies with a uniform decision (do nothing) and/or highly fluctuating LSPI feature weight values with no signs of convergence. Although the final reward weights were equal, the values of the reward factors were not normalized so in practice more weight was placed on the final scores of the previous and current scenarios [0,20] than on the learning gain [-1,1] or the confidence gain [-7,7]. We also experimented with immediate rewards based on the correctness of the learner's choices but ended up omitting them from the final model. In the training data there were many cases of correct answers with no associated hint or feedback; thus, these immediate rewards gave little information.

Gain	RL	Control	p-value
Unnormalized Knowledge	0.0068	0.0161	0.7224
Unnormalized SJT	0.1192	0.0740	0.1360
Unnormalized Combined	0.0718	0.0496	0.3034
Normalized Knowledge	0.0513	0.0613	0.8320
Normalized SJT	0.2683	0.1895	0.1950
Normalized Combined	0.1813	0.1352	0.3039
Confidence	0.5270	0.0762	0.0248

Table 3: Comparison of RL and control conditions in terms of unnormalized and normalized learning gain, and confidence gain (two-tailed t-test).

We experimented with LSPI with fast feature selection [13] to see if we could reduce the number of features used. As expected, the selected features would change from iteration to iteration. However, there were no signs that the algorithm was actually converging, which indicates that all of our features were relevant. This was also evident from the values of the weights resulting from standard LSPI; the absolute values of all feature weights were relatively high.

It is hard to measure which features contribute more to learning than others, because what matters is their combination. Below we report some observations based on the weights of the features for the hint/feedback decision-making policy. The weight for the provide hint action is larger in the Being Heard scenario (first try) compared to the second try, and when a question has not been seen before there is a large weight for the provide hint action. These weights could correspond to a teaching strategy of being more likely to hint on the first attempt and less likely on the second attempt (i.e., provide more support for the first attempt). If we have incorrect or mixed responses so far in the current scenario (even if the numbers are small) then the weights for both the provide hint and provide feedback actions have large values, and if we have incorrect or mixed responses in the previous scenario (even if the numbers are small) then the weights for both the provide hint and provide feedback actions have large values. These weights could correspond to a teaching strategy of providing more support when mistakes have been made. Similar observations apply to response times. If the average response time for incorrect or mixed responses so far in the current scenario is high then the weights for both the provide hint and provide feedback actions have large values. If the average response time for incorrect or mixed responses in the previous scenario is high then the weights for both the provide hint and provide feedback actions have large values. These weights could correspond to a teaching strategy of providing more support to students who make mistakes and take a longer time to respond. We also see that if the question difficulty is high then there are weights with large values for both the provide hint and provide feedback actions corresponding to giving more support for more difficult questions.

6 RESULTS

We use the two-tailed t-test for all statistical significance results reported below. Across both conditions the experience helped people learn as measured by the pre- and post-tests. The scores on these tests range from 0 to 1. We report both the total (combined) score on the test as well as separate scores for the knowledge and SJT questions. In all cases, the scores are the ratio of correct answers to the number of questions. We report unnormalized learning gains in each case which are simply: post-test score - pre-test score. We also report normalized learning gains using the definition in Grafsgaard et al. [9]. Thus in each case the normalized learning gain is: (post-test score - pre-test score) / (1 - pre-test score) if post-test score is greater than pre-test score, and (post-test score - pre-test score) / pre-test score otherwise. We did not have any cases where the pre-test or post-test scores were equal to 1 or 0.

Based on the means, standard deviations, and the minimum and maximum scores (knowledge, SJT, combined) and self-reported confidence values, we saw no floor or ceiling effects.

The increase in average knowledge test scores across both conditions (pre-test=0.6432 to post-test=0.6545) was not significant (p=0.3920). However, the increase in average SJT scores across both conditions (pre-test=0.5789 to post-test=0.6761) was significant (p=1.3094 x 10^{-8}). This result replicates earlier work with the ELITE system, which showed greater learning gains on SJT tasks since these are better-aligned to the scenarios [7]. The increase in combined (knowledge and SJT) test scores across both conditions (pre-test=0.6060 to post-test=0.6670) was also significant (p=2.8742 x 10^{-7}).

The unnormalized mean SJT gain was greater for the RL condition (0.1192) compared to the baseline (0.0740) but this difference was not significant (p=0.1360). The normalized mean SJT gain was greater for the RL condition (0.2683) compared to the baseline (0.1895) but this difference was not significant either (p=0.1950). Table 3 shows the knowledge, SJT, and combined learning gains (unnormalized and normalized) for the RL condition and the control condition.

No significant differences were found between the RL and the control condition on measures of scenario performance, based on a score where correct choices earn 1 point and mixed choices earn 0.5 points.

The RL condition differed from the control condition most notably in the quantity of guidance it provided to learners (mean number of messages = 43.1622 for RL versus 5.2286 for control, p=1.3861 x 10^{-44}). Although no significant difference in learning was seen between the conditions, the increased ITS guidance in the RL condition significantly affected the confidence of learners. The mean pre-survey self-evaluations of confidence (on a 7 point Likert scale) were not significantly different (5.5631 for RL and 5.5429 for control, p=0.9218); the mean post-survey self-evaluations of confidence (also on a 7 point Likert scale) were significantly different with the RL condition increasing to 6.0901 and the control condition only increasing to 5.6191 (p=0.0197). The confidence gain (raw difference in scores) was 0.5270 for the RL condition and 0.0762 for the control condition and this difference was significant (p=0.0248) (Table 3).

There was also a borderline-significant trend in which the coach was more highly rated on a 6 point Likert scale (mean=4.7143) in the RL condition than the control condition (mean=4.2980); p=0.0510.

7 CONCLUSIONS AND FUTURE DIRECTIONS

From a technical perspective, this work found that Least-Squares Policy Iteration (LSPI) [12] offers some distinct benefits for applying Reinforcement Learning (RL) to Intelligent Tutoring Systems (ITSs). We were able to use over 500 state-action features; no pruning was necessary and thus no information was potentially lost. There was also no need to develop a simulated learner to facilitate a large number of trial-and-error interactions. Compared to prior applications of RL to ITS, these advantages enabled training policies with fewer assumptions about the domain and users.

From a social perspective, this work shows the potential for machine intelligence to improve how we teach interpersonal skills. Unlike earlier ITSs, which demonstrated the effectiveness of RL exclusively on STEM topics, this work showed that policies could converge and show positive benefits even for a domain where human decisions are based on subtle personal cues that are not explicitly available to the RL policy (e.g., voice affect for a virtual human). This is particularly important, since modern workplaces require teamwork that combines both social and STEM-related skills [8].

The evaluation of the RL-trained policy versus the heuristics-based (control) condition was positive, but not fully conclusive. Replicating earlier results [7], we found significant pre-post learning gains, indicating that both conditions were effective. The RL-policy also showed higher learning gains than the control condition for SJT questions, but this difference was not statistically significant for this sample size. With that said, the ability for RL to match or exceed the expert human heuristics is promising: hand-tailoring coaching rules for scenario-based training is non-trivial, so the ability to infer these rules from user data could enable ITS support to extend to new domains.

The most definitive result was that learners reported significantly higher levels of confidence after the RL condition. Confidence and test scores were both part of the reward function used to train the RL policies, so this might indicate the most room for improvement in confidence. This is notable since confidence is particularly important for interpersonal skills: a significant number of errors are omissions, where individuals know the skill but lack the confidence to apply it, especially in the case of bystander intervention for both violence and sexual harassment/assault [6].

In terms of behavior, the RL condition differed from the control condition most notably in the much larger quantity of coach guidance it provided to learners, but the connection between this increase and the increased confidence ratings for the RL condition is unclear. Positive feedback is one potential influence on learner confidence. The control condition never gave positive feedback after correct answers, but in some cases the RL policy would. Learner confidence could also be affected by the overall quantity of coach messages; learners may feel they learn more if they see more content. The borderline-significant trend to rate the coach more highly in the RL condition may be a similar result (i.e., praise and overall quantity of communication may result in increased ratings).

In principle, it might be possible that coach guidance has a somewhat linear influence on confidence (e.g., learners feel more confident, regardless of the content). However, it is not the case that the RL policy was simply giving feedback and hints in all states. To explore such a causal relationship, a follow-up experiment would need to compare the heuristic of always giving hints and feedback to these RL policies. Such heuristics are often used in real-world settings and tuned over time. This approach is simpler than using RL but has the disadvantage of only being able to evaluate a single tutoring policy at a time whereas RL explores many different policies by estimating their expected rewards (e.g., test scores). In this case, the RL policy identified a mechanism that expert designers did not, even after many years of use.

However, there are potential areas for future improvement. Currently our RL policies do not have any information about the specific questions asked, other than general features such as question difficulty. It may be beneficial to keep track of the context of the conversation in the RL state. This could potentially lead to more interesting and perhaps more successful policies. It is also the case that RL is using pre-authored content whose quality will impact the ability of the resulting policies to influence later test scores. It could be the case that pedagogical content that more strongly stressed underlying principles could help learners improve their test scores. Especially in interpersonal skills training, novices may fixate on surface-level features (e.g., the specific problems in the scenarios and on the test) and not be able to learn and apply general skills in interactive scenarios that help improve test results. By changing the pre-authored content for ELITE Lite Counseling and re-running these experiments, we may see increases in both learning gain and confidence in the RL condition.

Although the version of the system that we used is designed for teaching counseling skills to U.S. officers, the same infrastructure can be used for teaching other types of interpersonal skills. Indeed, later versions of the system deal with the issue of addressing sexual harassment in the workplace. Neither the RL state variables nor the RL actions that we have used depend on the specific topic. Thus we could potentially utilize the same RL setup for many different topics provided that we have access to topic-specific content and data.

However, while this work shows an additional domain where RL offers improvements over hand-crafted policies, substantial work remains to scale up RL to a broad range of systems. This results from a lack of data pipelines from classrooms to data repositories: transmitting data from systems to repositories, and transmitting new policies back to the systems. Ideally, the entire process would be automated such that RL would continually train improved policies as new data arrives. At the moment, much work is done by researchers in setting up the RL learning (e.g., identifying features, discretizing their values, designing the rewards). Although the setup used in this paper could be reused for similar interpersonal skills training, a more general approach is needed to automate the process of going from raw data to tutoring policies.

ACKNOWLEDGMENTS

The work depicted here was sponsored by the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

[1] David H. Autor. 2015. Why are there still so many jobs? The history and future of workplace automation. *Journal of Economic Perspectives* 29 (2015), 3–30.

[2] Joseph E. Beck, Beverly Park Woolf, and Carole R. Beal. 2000. ADVISOR: A machine learning architecture for intelligent tutor construction. In *Proc. of the National Conference on Artificial Intelligence (AAAI).*

[3] Alan S. Carlin, Chris Nucci, Evan Oster, Diane Kramer, and Keith Brawner. 2018. Data mining for adaptive instruction. In *Proc. of the 31st International Florida Artificial Intelligence Research Society Conference (FLAIRS).*

[4] Min Chi, Pamela Jordan, and Kurt VanLehn. 2014. When is tutorial dialogue more effective than step-based tutoring?. In *Proc. of the 12th International Conference on Intelligent Tutoring Systems (ITS).*

[5] Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction (UMUAI)* 21, 1–2 (2011), 137–180.

[6] Ann L. Coker, Patricia G. Cook-Craig, Corrine M. Williams, Bonnie S. Fisher, Emily R. Clear, Lisandra S. Garcia, and Lea M. Hegge. 2011. Evaluation of Green Dot: An active bystander intervention to reduce sexual violence on college campuses. *Violence Against Women* 17, 6 (2011), 777–796.

[7] Mark G. Core, Kallirroi Georgila, Benjamin D. Nye, Daniel Auerbach, Zhi Fei Liu, and Richard DiNinni. 2016. Learning, adaptive support, student traits, and engagement in scenario-based learning. In *Proc. of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).*

[8] David J. Deming. 2017. The growing importance of social skills in the labor market. *The Quarterly Journal of Economics* 132, 4 (2017), 1593–1640.

[9] Joseph F. Grafsgaard, Joseph B. Wiggins, Kristy Elizabeth Boyer, Eric N. Wiebe, and James C. Lester. 2013. Automatically recognizing facial expression: Predicting engagement and frustration. In *Proc. of the International Conference on Educational Data Mining (EDM).*

[10] Matthew Jensen Hays, Julia C. Campbell, Matthew A. Trimmer, Joshua C. Poore, Andrea K. Webb, and Teresa K. King. 2012. Can role-play with virtual humans teach interpersonal skills?. In *Proc. of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).*

[11] Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. 2009. Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems* 22, 4 (2009), 266–270.

[12] Michail G. Lagoudakis and Ronald Parr. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4 (2003), 1107–1149.

[13] Lihong Li, Jason D. Williams, and Suhrid Balakrishnan. 2009. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *Proc. of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH).*

[14] Collin F. Lynch, Kevin D. Ashley, Vincent Aleven, and Niels Pinkwart. 2006. Defining "Ill-Defined Domains"; A literature survey. In *Proc. of the ITS 2006 Workshop on Intelligent Tutoring Systems for Ill-Defined Domains.*

[15] Jonathan P. Rowe and James C. Lester. 2015. Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In *Proc. of the 17th International Conference on Artificial Intelligence in Education (AIED).*

[16] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning: An introduction.* MIT Press.