Conversational sim2real: Training Tutoring Systems on User Simulators Using Reinforcement Learning

Paul Bricman
University of Groningen
paulbricman@protonmail.com

Abstract

The development of intelligent tutoring systems (ITS) is constrained by domain-specific knowledge. In contrast, dialogue systems are able to retrieve information and provide task-based assistance across diverse domains thanks in part to the use of user simulators. In this paper, I investigate the reliability of using pretrained language models as user simulators in developing ITS. An RL-based tutor agent is trained to assist a user simulator based on GPT-Neo in articulating answers to overarching questions (e.g. "Why do electrons repel each other?") through a set of static predefined interventions (e.g. "How can this be split into parts?") by means of Q-learning. Finally, I evaluate the agent's performance in transfer to live human users through a set of dialogues with the untrained and trained versions, before testing the difference in obtained reward for significance.

1 Introduction

The vast majority of conversational assistants and dialogue systems being researched and deployed today are focused on information retrieval (IR) or task-based intent [10]. Dialogue systems designed for facilitating IR aim to understand what piece of information the user is searching for (e.g. the weather forecast, the publication date of a movie, etc.), before accessing it and relaying it to the user [7]. Similarly, task-based dialogue systems focus on understanding the user's intent with respect to a discrete action to be taken on their behalf (e.g. making a restaurant reservation, ordering food, etc.). In contrast, relatively few resources have been invested in developing conversational assistants which help users reason to conclusions and identify appropriate actions themselves, instead of merely delegating the decision-making process, missing rich opportunities for promoting user agency.

However, this user-centered approach has been intimately adopted by a different research field, namely intelligent tutoring systems (ITS) [8, 15]. Developments in ITS aim to create systems which help actively cultivate both theoretical and practical skills in students through adaptive exercises and timely feedback on their performance. The goal of those systems is not merely to outsource the problem-solving skills of their users, but instead to nurture those very skills in their users. Adapting the difficulty of exercises to the current user skill level and providing hints at strategic points in the curriculum are only a handful of the techniques which appear to promote the effectiveness of ITS [17].

Unfortunately, developing an ITS requires both a solid understanding of the target domain (e.g. electronics, programming) and comprehensive knowledge about domain-specific obstacles which students might face. This is reflected in the niched nature of the ITS literature, consisting largely in analyses of the effectiveness and design choices of highly specialized systems [8]. In contrast, however, dialogue systems often strive for generality, being able to conversationally interface users to large bodies of factual knowledge and vast sets of actions to be taken on their behalf [10, 18, 14, 7, 11]. To overcome the difficulty in implementing domain-agnostic ITS, insights from sample-efficient dialogue system development might be leveraged.

Preprint. Under review.

One way in which dialogue systems overcome the low availability of user data across various domains is by using user simulators [12, 11, 7]. Essentially, the conversational assistants are trained and evaluated against simulators which exhibit procedurally-generated intents (e.g. they are looking for the publication date of a certain movie, they want to make a restaurant reservation with specific details) [10]. Those user simulators range in terms of sophistication from rule-based systems to statistical ones, or even combinations of the two, constructing coherent user personas to be maintained throughout the interaction [14]. This can be seen as a specific instance of the more general sim2real paradigm from Reinforcement Learning (RL), where agents are trained in a virtual environment due to low interaction costs, before transferring their abilities to real-world environments [9]. While most sim2real developments are seen in interaction with 3D environments mimicking real-world physics, there has been limited interest in using dialogue as a conversational and textual environment to be navigated.

Moreover, despite the user simulators being trained on limited data to learn how to mimic user behavior in new situations, the training data is often limited to previous dialogue datasets [16]. This fails to take advantage of the rich models of the world internalized as latent representations by models trained to autoregressively predict text in a general setting, unbound by conversational contraints [3]. Language models trained on large corpora are likely to have encoded comprehensive knowledge about the world as an instrumental goal in reducing perplexity in text generation. This makes them prime candidates for being employed as plug-and-play world simulators in a broader sense, while still qualifying as user simulators. The language model essentially becomes the RL agent's environment, changing its state as the result of the agent's actions, and supplying the agent with rewards accordingly.

In this paper, I investigate the reliability of using pretrained language models as user simulators in developing conversational tutoring systems. I address this by using Q-learning to train a tutor agent which is rewarded for helping the user simulator reach seemingly pertinent and diverse answers to an overarching question selected to guide the specific dialogue episode (e.g. "What are the most important books ever written?"), in what will be referred to as the Question Answering Assistance (QAA) task. Both state and action spaces are discretized for simplicity, where states are quantized regions of the latent space populated by user simulator replies, and actions are selected from a finite set of approximately 200 intervention prompts (e.g. "How could you approach this differently?"). Following 20,000 10-turn dialogues with the user simulator, the trained tutor agent is then tested for significance against an untrained tutor agent at helping the present author answer sampled questions, to examine transfer performance.

2 Methods

2.1 Data

The overarching questions which guide individual dialogues have been sampled from the first 10,000 questions contained in the Quora Question Pairs dataset [2]. In contrast, the interventions which the tutor agent has at its disposal during its interaction with the user simulator have been extracted from a previously published set of general question prompts designed to challenge students to engage with arbitrary topics (see Table 1 for samples) [5]. Moreover, in the current paper, an additional intervention is accessible to the tutor agent, namely a reply consisting of the string "Great, now please try to answer our original question again.", followed by the repeated overarching question guiding the specific dialogue.

2.2 NLP Models

Multiple pretrained natural language processing (NLP) models are employed in this project. First, the user simulator is based on a pretrained instance of GPT-Neo 1.3B, using a simple greedy decoding strategy for text generation [3]. As it is meant to mimic a user's replies in a dialogue, whenever it has generated a new line or more than two sentences, its generation is abruptly stopped by only allowing an end-of-sequence token. This ensures the user simulator does not accidentally continue the dialogue beyond its assigned scope, generating plausible replies for the actual tutor agent. Such complex interactions between language models and goal-oriented RL agents are only explored as possible future work.

Table 1: Question samples. Both overarching and intervention questions are answered by the user simulator. However, each dialogue episode starts with a randomly sampled overarching question, while the intervention questions are then employed by the tutor agent based on its RL policy.

Type	Contents
Overarching	What are natural numbers?
Overarching	What are the most important books ever written?
Overarching	Why am I afraid of pain?
Overarching	How can I find my lost phone?
Overarching	Why do electrons repel each other?
Intervention	How would you explain this to a child?
Intervention	How can this be split into parts?
Intervention	What's an example of this?
Intervention	How would you represent this visually?
Intervention	What are the basic facts of this?

Second, the encoder model used to transform user simulator replies into semantic embeddings which act as environment states in this conversational sim2real paradigm is based on MiniLM-L6-H384-uncased [4]. Additionally, the same encoder model is used during reward computation in order to penalize repetitive user simulator answers to the overarching question. Third, the cross-encoder model used to estimate whether a user simulator reply is a pertinent answer to the overarching question is based on qnli-distilroberta-base [1].

2.3 Environment & Agent

Whenever the RL agent performs an action (i.e. replies using an intervention prompt), the sandbox environment which coordinates the training procedure appends the reply to a growing dialogue data structure. The environment's dynamics are based on prompting the user simulator to generate a user reply to the tutor agent's intervention. The simulated user reply is similarly appended to the data structure representing the present dialogue. However, due to the fact that the tutor agent uses a discretized state space, the user reply gets encoded and assigned to its closest cluster centroid during quantization. The cluster centroids are determined in advance, before the tutor agent is trained using Q-learning, by means of generating 10,000 5-turn dialogues between a user simulator and a random untrained tutor agent, following K-means clustering ($K = 100$). The purpose of the prior random dialogues is to provide an estimate of the distribution of user replies across latent space, in order for the cluster centroids used in quantization to be effective in capturing relevant nuances in user replies.

Besides observing a new environment state based on the semantic embedding of the user reply which is then assigned to one of K=100 clusters, the tutor agent also receives a numerical reward following each of its interventions. If the user's reply is too semantically similar to previous replies, based on cosine similarity to previous user embeddings ($\delta = 0.5$ threshold), then the reward is 0. The rationale behind this is that not only is the ideal tutor agent behavior one which elicits pertinent answers to the overarching questions, but one which elicits multiple diverse such answers throughout the interaction. This way, repeating the same pertinent answer yields no reward. However, in the case of a semantically novel user reply, the reward is equal to the value directly predicted by the cross-encoder standing for the estimated amount of question-answer relatedness.

The RL tutor agent itself is based on Q-learning, with the discrete state space entirely described by the cluster index to which the user reply is assigned to, and the action space consisting of the set of tutor interventions. Both α and γ are set to 0.5 as default values, with no hyperparameter tuning involved.

2.4 Training

As Q-learning is an off-policy RL algorithm, the guiding policy of the tutor agent over the course of its training is a softmax policy whose temperature linearly decreases from 1 to 0 over the course of the training phase. This enables the tutor agent to intensely explore the policy space in the early stages

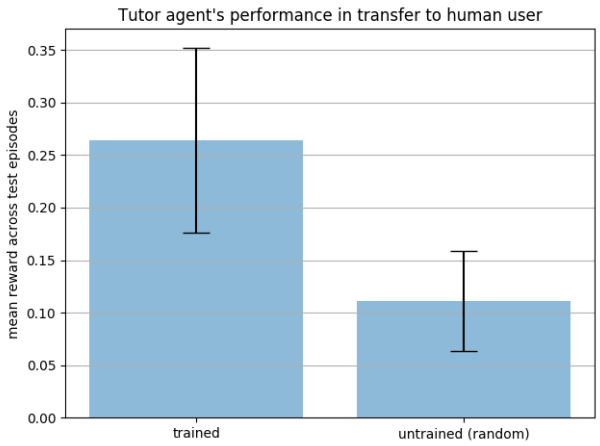

Figure 1: Mean reward obtained by tutor agent over test dialogues with a human user.

Table 2: Dialogue sample.

Type	Contents
Overarching	Do water purifiers like Brita really work?
User simulator	Yes, they do.
Tutor agent	How would you explain this to a student?
User simulator	The Brita is a water purifier that uses a special filter to remove chlorine and other harmful substances from the water.
Tutor agent	How are the components of this interconnected?
User simulator	It is a filter that is connected to the water supply.
Tutor agent	How can you best use this?
User simulator	You can use it to purify water in your home.

before gradually committing to the seemingly most promising interventions given the circumstances. Training takes place for 20,000 10-turn dialogues with the user simulator (see Table 2 for a sample).

3 Results

In this paper, I investigated the reliability of language models as user simulators in developing RL-based intelligent tutoring systems. Following an extensive training process consisting of a tutor agent interacting with the user simulator for 20,000 10-turn dialogues, the tutor agent has then been evaluated in terms of performance in transfer to a human user by means of comparing it to an untrained (i.e. random) tutor agent. Both the trained and untrained tutor agents guided five 5-turn dialogues with the present author instead of the user simulator, while the reward has been tracked in the same way as during training, except for the human user replacing the user simulator's role in conversation. The same sampled overarching prompts have been used for both conditions, to avoid confounding variables. Additionally, the order of the two has been randomized, to ensure a rudimentary level of blindness in the experiment. While the trained agent obtained a higher mean reward than the untrained agent during the human trials (see Fig. 1), an unpaired one-sided t-test revealed the difference not to be significant ($p = 0.08, t = 1.53$).

Moreover, the moving average of the reward history recorded during training exhibited limited signs of reaching a plateau (see Fig. 2). When comparing training and testing performance, it is relevant to note that the training phase was based on a language model of moderate size providing answers, while the testing phase was based on a human user replying to the tutor agent's interventions.

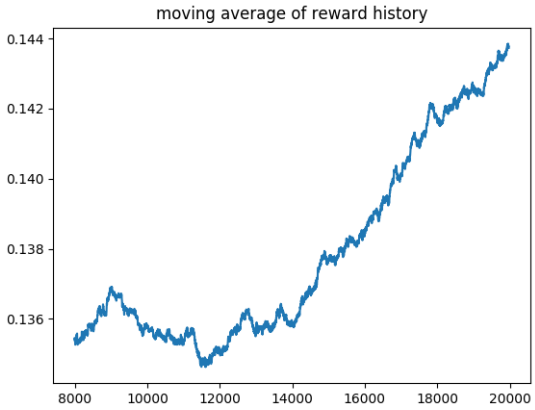

Figure 2: Smoothed reward history recorded during the tutor agent's training.

4 Discussion

4.1 Conclusion

Promising initial results indicate that pretrained off-the-shelf language models are underexplored user simulators. They exhibit the ability to hold coherent dialogues with other agents, and in doing so, they provide a rich and scalable interactive environment for training other agents via the conversational sim2real paradigm. Those initial results are even more relevant when considering the myriad issues which have crippled the tutor agent's transfer performance.

4.2 Potential issues

4.2.1 Poor objective function design

The reward which the tutor agent received during training was not only a function of the last user simulator's reply and the overarching question guiding the respective dialogue, but was also based on the prior user simulator replies. If the last reply was extremely semantically similar to a prior pertinent answer, then the reward given to the agent was zero, to discourage repetitive answers. However, the prior replies to which the new one was tested against only consisted in the ones which were estimated to themselves be pertinent answers. This meant that if past replies were just under the threshold of question-answer compatibility as estimated by the cross-encoder model, then new slightly pertinent, yet repetitive answers would evade the filter, leading to important quantities of reward for a suboptimal behavior. To fix this, reward could simply be scaled by the semantic similarity measure of the new reply to all prior ones, rather than being based on a hard cut-off of semantic similarity and question-answer compatibility.

4.2.2 Dialogue lacks the Markov property

The entire state of the environment as perceived by the tutor agent has simply been the cluster index of the semantic embedding of the last user simulator's reply, following a hard assignment to a set of precomputed centroids. No information from past user replies, past tutor agent interventions, or the overarching question was reflected in the state of the environment, as far as the tutor agent was concerned. In a sense, this work has assumed that dialogue has the Markov property, the last environment state being sufficient to guide future interaction. Yet this is far from true, as long-term linguistic dependencies lead to the conversational context permeating each individual reply, providing important information. To fix this, rather than encoding only the last user's reply into latent space out of context, the whole past dialogue could be encoded at once, while only the token embeddings of the last user's reply could be mean-pooled together into one final semantic embedding. This would lead to a state of the environment which reflects the most recent state of understanding of the user, while still containing information about topics, entities, and others mentioned previously in the dialogue.

4.2.3 Action space has limited expressiveness

While the user simulator's reply span a massive space, the tutor agent's contributions to the dialogue in the form of interventions have been limited to a set of approximately 200 predefined prompts. Additionally, those prompts have been created in an out-of-context regime, presenting bias towards predominantly academic styles of interaction (e.g. "How would you model this?"). The suitability of the set of interventions in open-ended natural conversation seeded by both formal and informal overarching questions has been limited, prompting overly formal and out-of-place interventions on occasion. This could be addressed by either making the action space larger while still keeping it discrete, or by making the action space continuous altogether. The continuous action space might be implemented by designing the tutor agent in such a way as to also make use of the flexibility of pretrained language models, just like the user simulator, while still incentivizing it with RL-based goal-oriented constraints. For instance, the tutor agent's intervention could perhaps be the output of a few-shot text generation task, where the inputs are (1) the user's last reply, and (2) the turn-level goal to be implemented given the current policy (e.g. "ask them to break the problem down into smaller parts"), as occasionally practiced in more traditional dialogue systems. Alternatively, a tutor agent based in part on a pretrained language model could receive external goal-directed supervision in the form of adjustments to the probability distribution over upcoming tokens (i.e. semantic similarity to raw intervention prompts or to continuous action embeddings directly outputted by DQN approaches) [6].

4.3 Future work

4.3.1 Change the task

The proof-of-concept experiment described in this paper has been entirely based on a rudimentary version of the question answering assistance (QAA) task introduced above. However, there is no reason not to consider alternative tasks informing the RL agent's policy, while maintaining most of the training setup. For instance, a reading comprehension assistance task might be based on turn-level estimates of whether the simulated user understands an overarching concept. If the user simulator's replies help significantly reduce perplexity on reconstructing a target paragraph to be learned, then the tutor agent could be deemed successful. Alternatively, if the growing dialogue as a knowledge base manages to help a language model correctly answer a set of fill-in-the-gaps exercises, then again the tutor agent could be rewarded accordingly. Moreover, the RL agent could be incentivized to relax or strengthen specific beliefs on the part of the user by obtaining a reward as a function of how strong the user appears to exhibit them. Similarly, the RL agent could be trained to elicit certain emotions from the user, being rewarded based on sentiment analysis of their replies.

4.3.2 Hallucinate conversational futures

Unlike many successful RL agents, the tutor agent trained in the context of this paper incorporates no explicit planning, relying solely on following Q-values of various state-action pairs. However, it is plausible that modeling different ways in which the dialogue might evolve following various interventions could help the tutor agent implement an even more effective policy. Concretely, the tutor agent could make use of a user simulator to hallucinate different conversational futures and see how rewarding they are, regardless of whether it is actually interacting with a real human or not during the actual dialogue. For instance, it might predict that hinting at an analogy to physics might not be particularly promising when guiding a user completely unfamiliar with physics. Moreover, the RL agent could be trained on a general user simulator, but might employ a specialized simulator tailored to the unique human user, for more accurate conversational forecasts. The specialized user simulator could be based on the user's personal knowledge base or on a compact user persona.

4.3.3 Extend the set of interventions

Another extensible component of the conversational sim2real setup is the action set. There is no reason why the tutor agent should be mostly limited to static prompts written in advance. For instance, an intervention could employ retrieving information related to the user's last reply using an online search engine. The tutor agent might learn that this action is particularly promising when the user exhibits a total lack of factual knowledge on the topic. Alternatively, another available intervention could be to make use of a few-shot prompt to generate a set of counterarguments to the user's claim, to

paraphrase it, or to highlight some of its consequences. Discrete actions expressed through language prompts, together with information retrieval routines, among others, could simply be added to the set of available interventions which the tutor agent to make use of.

4.3.4 Extend to other modalities

While a core aspect of the present paper has been to investigate means of exploiting the rich latent representations internalized by pretrained language models, the interactive environment might be extended to other modalities through multi-modal models. For instance, a multi-modal encoder which embeds both texts and images in a joint latent space might be used to guide users in achieving visual-intensive tasks by exapting the textual interactive environment created by language models [13]. Concretely, the multi-modal encoder might be tasked with ranking a predefined set of texts based on semantic similarity to the feed from a user's camera or screencast (e.g. "I used too much solder while soldering this through-hole connection." or "I made a free-form selection using the Lasso tool."). During training, only those text prompts would be valid user simulator responses, ranked by likelihood in order to build on the knowledge of the world internalized by language models. Alternatively, just like the proposed fix for limited action set expressiveness, predefined text prompts describing user states could be contextualized and rendered more coherent with respect to the ongoing dialogue through few-shot prompting.

4.3.5 AI metacognition

While the entirety of this paper has been focused on eventually helping human users answer arbitrary questions, another possible development of this approach is to imbue AI systems with metacognitive skills. The dialogue loop meant to identify pertinent and varied answers to a question could be triggered by a higher-level RL system opting for a "self-reflection" action. After reasoning "in its head" by means of simulating a back-and-forth conversation around a given overarching question, it would likely gain more insight into the problem than it would have by simply answering the question directly through a language model or via IR. In this, the conversational sim2real paradigm can be used in an online setting to find high-quality and highly-interpretable solutions to recent problems.

4.3.6 Parallelize training

As an engineering direction for future work, unlike the previous research ones, the tutor agent could facilitate multiple dialogues concurrently, by means of batching the autoregressive generation step of the training loop which the user simulator is based on. Q-values would then be aggregated, or experience cumulated in a DQN approach. Moreover, the tutor agent could be trained on a multi-node cluster, facilitating dialogues in a decentralized fashion before aggregating adjustments during rendezvous.

4.4 Ethical considerations

While the goal of this paper has been to explore an educational application designed to be beneficial for the user, the use of pretrained language models as user simulators might, unfortunately, increase the likelihood of a few classes of problematic systems. For instance, swapping the question answering assistance objective for a more malicious one (e.g. persuasion, anxiety induction, ransom payment), has the potential to provide interactive environments for training RL agents which are radically misaligned with the end user's goals. Alternatively, the richness of the interactive environments contained by off-the-shelf pretrained language models also poses an AI safety risk. If an AGI system misaligned with human values would identify the need of learning how to persuade humans as an instrumental goal in achieving other long-term goals, the pretrained language models could sadly yield a high-speed sandbox for gaining experience in that task. While the conversational sim2real paradigm explored in this paper has the potential to seed a fruitful line of research in developing intelligent tutoring systems, it is important to be conscious of such failure modes going forward.

References

[1] cross-encoder/qnli-distilroberta-base · hugging face, . URL `https://huggingface.co/cross-encoder/qnli-distilroberta-base`.

[2] First quora dataset release: Question pairs, . URL https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs.

[3] GPT-neo, . URL https://www.eleuther.ai/projects/gpt-neo/.

[4] nreimers/MiniLM-l6-h384-uncased · hugging face, . URL https://huggingface.co/nreimers/MiniLM-L6-H384-uncased.

[5] P. Bricman. K-probes. URL https://github.com/paulbricman/k-probes/blob/3e98b87bd2bc4be748d0115beb608b97f41de8fb/prompts.json. original-date: 2020-11-25T18:45:30Z.

[6] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu. Plug and play language models: A simple approach to controlled text generation. URL http://arxiv.org/abs/1912.02164.

[7] P. Erbacher, L. Soulier, and L. Denoyer. State of the art of user simulation approaches for conversational information retrieval. URL http://arxiv.org/abs/2201.03435.

[8] A. C. Graesser, X. Hu, B. D. Nye, K. VanLehn, R. Kumar, C. Heffernan, N. Heffernan, B. Woolf, A. M. Olney, V. Rus, F. Andrasik, P. Pavlik, Z. Cai, J. Wetzel, B. Morgan, A. J. Hampton, A. M. Lippert, L. Wang, Q. Cheng, J. E. Vinson, C. N. Kelly, C. McGlown, C. A. Majmudar, B. Morshed, and W. Baer. ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics. 5(1):15. ISSN 2196-7822. doi: 10.1186/s40594-018-0110-y. URL https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-018-0110-y.

[9] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? 5(4):6670–6677. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2020.3013848. URL http://arxiv.org/abs/1912.06321.

[10] X. Li, W. Wu, L. Qin, and Q. Yin. How to evaluate your dialogue models: A review of approaches. URL http://arxiv.org/abs/2108.01369.

[11] H.-c. Lin, N. Lubis, and S. Hu. Domain-independent user simulation with transformers for task-oriented dialogue systems. page 12.

[12] B. Peng, X. Li, J. Gao, J. Liu, and K.-F. Wong. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192. Association for Computational Linguistics. doi: 10.18653/v1/P18-1203. URL http://aclweb.org/anthology/P18-1203.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. URL http://arxiv.org/abs/2103.00020.

[14] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau, and J. Weston. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.24. URL https://aclanthology.org/2021.eacl-main.24.

[15] A. C. Sales and J. F. Pane. The role of mastery learning in intelligent tutoring systems: Principal stratification on a latent variable. URL http://arxiv.org/abs/1707.09308.

[16] W. Wang, Y. WU, Y. Zhang, Z. Lu, K. Mo, and Q. Yang. Integrating user and agent models: A deep task-oriented dialogue system. URL http://arxiv.org/abs/1711.03697.

[17] J. H. Wong, S. S. Kirschenbaum, and S. Peters. Developing a cognitive model of expert performance for ship navigation maneuvers in an intelligent tutoring system. page 8.

[18] R. Zhou, S. Deshmukh, J. Greer, and C. Lee. NaRLE: Natural language models using reinforcement learning with emotion feedback. URL http://arxiv.org/abs/2110.02148.