I love your chain mail! Making knights smile in a fantasy game world: Open-domain goal-oriented dialogue agents

Shrimai Prabhumoye*
Language Technologies Institute,
Carnegie Mellon University,
Pittsburgh, PA, USA
sprabhum@cs.cmu.edu

Margaret Li*
Facebook AI Research,
New York, NY, USA

Jack Urbanek, Emily Dinan, Douwe Kiela, Jason Weston, Arthur Szlam
Facebook AI Research,
New York, NY, USA

Abstract

We study learning within a rich multi-player text-based fantasy environment where agents engage in both actions and open-domain dialogue. Specifically, we investigate training a goal-oriented dialogue model with reinforcement learning (RL) that can learn to converse with other agents that speak and act such that goal actions are achieved during their interaction. We describe two tractable RL policies: learn to pick topics or an utterance given the top-K utterances from a dialogue model. We show these models outperform an inverse model baseline and can converse naturally with their dialogue partner in order to achieve goals.

1 Introduction

In this work, we study a multi-player text-based fantasy environment [31] with grounded actions and reference objects. Given a particular character to play in a particular scenario (location, set of objects and other characters to interact with), an agent should conduct open-ended dialogue with the goal of making their dialogue partner execute a specified action, differing from many other text-adventure game works that do not involve dialogue [22, 3]. The action could be an emote (smile, laugh, ponder, etc), or a game action (wear chain mail, drink mead, put glass on table, etc). The richness of the environment means that there are a huge set of possible tasks and scenarios in which to achieve a wide range of actions. We plan to make our code and models publicly available.

We train a variety of baseline models to complete the task. We compare agents trained to imitate human actions given a goal (an "inverse model") to two different RL approaches: optimizing actions with latent discrete variables (topics), or via rewarding actions sampled from the model (via the top-K outputs). We show that both types of RL agent are able to learn effectively, outperforming the inverse model approach or the chit-chat imitation baseline, and can converse naturally with their dialogue partner to achieve goals.

In short, our main contributions are: a new family of tasks that combines goal-oriented dialogue and chit-chat in a rich, fully realized environment, and the results and analysis of scalable RL algorithms and behavioral-cloning models (and simple heuristic methods) on these tasks.

*Equal contribution.

Wordplay: When Language Meets Games Workshop @ NeurIPS 2020,
https://wordplay-workshop.github.io/

2 LIGHT Game Environment

We work in the LIGHT game environment [31], which is a multi-user medieval fantasy text-based game. Characters can speak to each other via free text, send emote actions like *applaud*, *nod* or *pout* (22 emote types in total), and take actions to move to different locations and interact with objects (e.g. *get cutlery*, *put cutlery in drawer*, etc.), see Appendix D.5 for a full list of game actions and how the game engine works.

LIGHT is built with crowd-sourced data both for the world (locations, characters and objects) and human demonstrations of player interactions. There are a total of 663 locations, 1755 characters, and 3462 objects. They range from beaches with crabs and seaweed to crypts with archaeologists and coffins, yielding an extremely rich environment for agents to learn within. Crowdworkers were asked to play the role of characters within the game. This involved them making utterances, game actions and emotes, while interacting with each other (in pairs). The resulting gameplay data consists of 10,777 episodes with an average of 18.3 actions each of rich human play. These are split into train (8538), validation (500) and test (1739) portions, the latter being split into new episodes in existing settings (test seen, 1000) and completely new settings (test unseen, 739).

Players were not given specific goals, but instead asked to play the role convincingly of the character given, during play some of them effectively defined their own goals during the interactions, see Appendix Fig. 3. Existing work [31] does not consider using this data to learn goal-based tasks, but instead has only used this for chit-chat and action imitation learning, different to this work.

3 Tasks

The tasks we introduce in this work involve achieving open-domain goals during interaction between two agents in a given LIGHT scenario. One of the agents, which we will call the "environment agent" $\mathcal{M}_{env}$, together with the game engine, effectively functions as an environment for the other agent, denoted by $\mathcal{M}_{player}$. We assume that the environment agent is fixed; in this work it will be a model trained via behavioral cloning from human-human interaction data. $\mathcal{M}_{player}$ must conduct open-ended dialogue such that a given goal action is executed in the future by the environment agent.

More formally, the two agents $\mathcal{M}_{env}$ and $\mathcal{M}_{player}$ are given their views of the scenario ($\mathbf{D}_{\text{env}}$ and $\mathbf{D}_{\text{player}}$ respectively). These consist of the setting name, scenario description, character names, and their own persona, all described as a sequence of text (see Fig 1). Note that each agent can only access their own persona but not the persona of the partner with whom they are conversing, but they do know the name of their partner. Denote by t the time-step of the environment, $\mathbf{U}_{\mathbf{t}}^{\text{player}}$ and $\mathbf{U}_{\mathbf{t}}^{\text{env}}$ the utterances of the agents $\mathcal{M}_{player}$ and $\mathcal{M}_{env}$ respectively, and denote by $\mathbf{A}_{\mathbf{t}}^{\text{env}}$ the environment actions by $\mathcal{M}_{env}$. Hence the interaction sequence is:

$$\mathbf{S}_t = [\mathbf{U}_{\mathbf{0}}^{\text{player}}, (\mathbf{U}_{\mathbf{0}}^{\text{env}}, \mathbf{A}_{\mathbf{0}}^{\text{env}}), \mathbf{U}_{\mathbf{1}}^{\text{player}}, (\mathbf{U}_{\mathbf{1}}^{\text{env}}, \mathbf{A}_{\mathbf{1}}^{\text{env}}), \ldots, \mathbf{U}_{\mathbf{n}}^{\text{player}}, (\mathbf{U}_{\mathbf{n}}^{\text{env}}, \mathbf{A}_{\mathbf{n}}^{\text{env}})]. \tag{1}$$

The agent $\mathcal{M}_{player}$ is additionally given a persuasion goal $\mathbf{g}$ to achieve. That is, the objective of $\mathcal{M}_{player}$ is for $\mathcal{M}_{env}$ to take the action $\mathbf{g}$. An episode ends when $\mathbf{A}_t^{\text{env}} == \mathbf{g}$ or when n becomes larger than a set number of turns.

Goals We experiment separately with two different types of goals: game actions and emote actions. We use the same train, valid, test (seen and unseen) split of the original human-human LIGHT episodes, assign roles $\mathcal{M}_{player}$ and $\mathcal{M}_{env}$ randomly, and randomly pick an action by $\mathcal{M}_{env}$ that occurs in the episode as the goal. We can then present the corresponding setting to our agents in order to form a new interaction, but within the same scenario and with a goal that was naturally desirable and achievable within that setting.

In our setup, $\mathcal{M}_{player}$ speaks (but does not act). This allows us to study grounded dialogue between agents; it guarantees that the player cannot force the goal to be reached by performing actions itself. It has to produce appropriate utterances $\mathbf{U}^{\text{player}}$ such that $\mathcal{M}_{env}$ eventually takes the action $\mathbf{g}$.

Observations The state observation $\mathcal{O}_t = (\mathbf{D}_{\text{player}}, \mathbf{S}_{t-1}, \mathbf{g})$ at time t given to a model consists of the agent's setting description ($\mathbf{D}_{\text{player}}$), the utterance and action history up to that time step ($\mathbf{S}_{t-1}$), and the agent's goal ($\mathbf{g}$). Our models for $\mathcal{M}_{player}$ consume $\mathcal{O}_t$ as a flattened sequence of tokens, and return a dialogue utterance $\mathbf{U_t^{\text{player}}}$. Each structured component is represented in the flattened sequenced separated by a special token denoting the types, e.g. names, settings, etc.

3.1 Reinforcement learning formulation

Our task set-up can be framed as a Markov decision process. Because the entire history and goal is given to $\mathcal{M}_{player}$, the environment is Markovian. We give a terminal reward of +1 only if the goal g is achieved and 0 otherwise, i.e, it is +1 if the environment agent takes the goal action g. The episode ends after n steps. In our experiments we consider $n = 1$ and $n = 3$. When we formulate our tasks as a reinforcement learning problem, we will also refer to $\mathcal{M}_{player}$ as the "RL agent".

4 Models

In this section we describe the models for $\mathcal{M}_{env}$ and $\mathcal{M}_{player}$. In this work these are retrieval models, using the LIGHT dialogue training corpus as candidates (111k utterances).

Base agent architecture All our models adopt the same base architecture: a 12-layer bidirectional transformer [32] pre-trained on a large dialogue corpus (Reddit, 174M examples), and then fine-tuned on our task. To score retrieval candidates, we use a *bi-encoder* [13, 31], in which two transformers are used, one to encode the context, and another to encode a candidate response, and a dot product between the first output vector of each scores the match, and the maximum scoring candidate is chosen as the final utterance/action/emote. For actions, the candidates are the set of admissible actions at that game state, which are provided by the game engine, for example *get apple* is only available in the candidate set if it is a valid action (an apple is present in the room). For emotes, all 22 candidates are always available. For dialogue the training set candidates are used (111k in this case). To train the model, a cross entropy loss is used, with negatives sampled from the batch [21].

Environment agent The environment agent is the base agent described above, and stays fixed over episodes where an RL agent is trained. This helps guarantee our RL models stick to using the semantics of natural language (English) rather than so-called language drift of learning a new emergent language on the same tokens [16].

4.1 Inverse model

We consider an inverse model, trained to imitate human actions given a goal, as both a baseline for comparing to RL models, and for producing weights from which we can fine-tune. The inverse model consists of a bi-encoder, as described above, which takes as input an observation $\mathcal{O}_t$, and outputs an utterance. We train it by extracting from the human-human game logs training set (which does not have goals) every instance where a game action occurs at time t in $\mathbf{S_t}$. We consider as baselines both a version where the goal is given in the input, and where it is removed.

4.2 Topic RL model

Optimizing all the parameters of a large transformer architecture by RL is both incredibly costly in data efficiency and computing time, and is also known to have the problem of language drift [16]. A solution to both problems is to train most of the parameters of the model with human-human language data, and then only optimize some of the parameters [33] with RL. Here, we propose a straight-forward model for that purpose.

We build an RL agent that consists of two transformers: prediction of a topic from K topics given the observation (first transformer), followed by prediction of the final dialogue utterance given the observation and topic (second transformer). The first transformer is initialized to the inverse model, and K-means provides initial topic centers given its output representation. A two-layer fully connected neural net is then placed at the output of the first transformer which is trained by RL to predict the topic C. The second transformer, trained on human-human data, given the observation and topic C outputs the dialogue utterance. We use the Advantage Actor-Critic implementation **A2C** [14] to train the policy and the value function. Further details are given in appendix B.

Model	Goal Type	Test Seen ($n=1$) Reward	Test Seen ($n=3$) Reward	Test Seen ($n=3$) Turns	Test Unseen ($n=1$) Reward	Test Unseen ($n=3$) Reward	Test Unseen ($n=3$) Turns
Inverse model (no goal)	game act	0.185	0.345	2.55	0.160	0.345	2.57
Inverse model	game act	0.223	0.414	2.42	0.193	0.410	2.48
Top-K RL	game act	0.327	0.491	2.26	0.278	0.442	2.34
Topic RL	game act	**0.359**	**0.561**	2.15	**0.313**	**0.496**	2.26
Topic RL (1-step 3x)	game act	-	0.493	2.22	-	0.479	2.29

Table 1: Results on the test seen and unseen environments for our models.

Self: guard	**Partner:** archer	**Self:** swimmer	**Partner:** turtles
Persona:	I guard the castle. I guard the king. I would kill to protect the royal family	**Persona:**	I am a huge fan of deep sea exploration, but I take any chance I can get to go for a swim...
Setting:	The armory, Inside Tower. The near top of the tower 6 feet before the very top. Where the watchers keep their eye...	**Setting:**	Bank, Swamp This is a grassy area that surrounds much of the swamp. It's a plain field with some trees nearby along...
$\mathbf{U_0^{player}}$	This is the armory! The king keeps the best weapons here. Take a look -	$\mathbf{U_0^{player}}$	Just keep taking good care of your beautiful little turtle family! Your species is quite unique and I love to see you about when I go for a swim.
$\mathbf{U_0^{env}}$	Hello, I need to get into the palace to see the king. I think he might like to see these weapons.	$\mathbf{U_0^{env}}$	Well, thank you for that. Do you happen to know where my other turtle friend is? You haven't captured any turtles have you?
$\mathbf{A_0^{env}}$	get weapon	$\mathbf{A_0^{env}}$	hug swimmer

Table 2: Example 1-step episodes where after the Topic RL agent's utterance $\mathbf{U_0^{player}}$ the environment agent's response action $\mathbf{A_0^{env}}$ was equal to the RL agent's goal $\mathbf{g}$. Our RL agent both makes natural utterances given the situation, and that elicit the desired goal.

4.3 Top-K RL model

The Top-K model, related to [6], also keeps the number of trainable parameters small. As above it keeps close to the base retrieval model to avoid drift. It first uses the inverse model to get a context embedding $\tilde{s}$ from the observation, and a list of K candidate utterance embeddings $v_1, ...v_K$ corresponding to utterances $u_1, ...u_K$. These are the encodings by the inverse model of the K utterances it considers most likely given the context and goal. We form scores $t_i = (A\tilde{s}+b)^T v_i$, and obtain a probability distribution over these K candidates for our policy:

$$\pi(u_i|\text{context}) = \text{softmax}(t_0, ..., t_K)(i). \tag{2}$$

Here the trainable parameters of the RL agent are the map A and biases b.

5 Experiments

We compare our models on the game action tasks (with results for emotes given in the appendix). We experiment with differing number of steps n allowed to complete the goal, n=1 and n=3. Results for both seen and unseen test environments (§2) are given in Table 1. We report the average reward and for n=3 the average number of turns before completion. The results show clear improvements for our RL models compared to the inverse model for each n, and improvements for an $n = 3$ model compared to naively applying an n=1 model three times. As a sanity check we also tried, after training, to replace the Topic RL policy with random topic prediction, which yielded poor results, e.g. 0.217 reward for n=1 test seen game actions. Our model is clearly learning appropriate topic acts.

We show examples of successful utterances, achieving goal actions in Table 2 for a diverse range of scenarios, actions and language. For example, for the guard's goal to encourage the archer to *get weapon* the Topic RL model utters "This is the armory! The king keeps the best weapons here. Take a look", which ends up leading to the desired action in the subsequent turn. More examples (for both $n = 1$ and $n = 3$) are given in Appendix D.7.

A much more detailed analysis of the experiments is also given in appendix D. In short: we analyzed utterance choice, and find clear improvements in semantic connection with the RL models compared to the inverse model given the task. We also analyze model capacity e.g. choices of K, train vs. test performance, breakdown by goal and difficulty, showing there are a number of challenging tasks still unresolved for longer step, more difficult action subcases.

6 Conclusion

In this paper, we investigate agents that can interact (speak or act) and can achieve goals in a rich world with diverse language, with preliminary success. Future work should scale tasks and models to harder tasks with more steps with richer and richer goal (game) states.

References

[1] A. Bordes, Y.-L. Boureau, and J. Weston. Learning end-to-end goal-oriented dialog. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[2] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

[3] M.-A. Côté, Á. Kádár, X. Yuan, B. Kybartas, T. Barnes, E. Fine, J. Moore, M. Hausknecht, L. E. Asri, M. Adada, et al. Textworld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*, 2018.

[4] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2951–2960, 2017.

[5] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

[6] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, and B. Coppin. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

[7] L. El Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[8] M. Fatemi, L. E. Asri, H. Schulz, J. He, and K. Suleman. Policy networks with two-stage training for dialogue systems. *arXiv preprint arXiv:1606.03152*, 2016.

[9] M. Gasic, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, and S. Young. Pomdp-based dialogue manager adaptation to extended domains. In *Proceedings of the SIGDIAL 2013 Conference*, pages 214–222, 2013.

[10] A. Gupta, R. Lowe, J. Foerster, D. Kiela, and J. Pineau. Seeded self-play for language learning. In *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, pages 62–66, 2019.

[11] M. Henderson, B. Thomson, and J. D. Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, 2014.

[12] H. Hu, D. Yarats, Q. Gong, Y. Tian, and M. Lewis. Hierarchical decision making by generating and following natural language instructions. *arXiv preprint arXiv:1906.00744*, 2019.

[13] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston. Real-time inference in multi-sentence tasks with deep pretrained transformers. *arXiv preprint arXiv:1905.01969*, 2019.

[14] I. Kostrikov. Pytorch implementations of reinforcement learning algorithms. `https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail`, 2018.

[15] A. Lazaridou, A. Potapenko, and O. Tieleman. Multi-agent communication meets natural language: Synergies between functional and structural language learning. *arXiv preprint arXiv:2005.07064*, 2020.

[16] J. Lee, K. Cho, and D. Kiela. Countering language drift via visual grounding. *arXiv preprint arXiv:1909.04499*, 2019.

[17] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, and D. Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.

[18] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

[19] M. Li, J. Weston, and S. Roller. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*, 2019.

[20] R. Lowe, A. Gupta, J. Foerster, D. Kiela, and J. Pineau. On the interaction between supervision and self-play in emergent communication. *arXiv preprint arXiv:2002.01093*, 2020.

[21] P.-E. Mazaré, S. Humeau, M. Raison, and A. Bordes. Training millions of personalized dialogue agents. *arXiv preprint arXiv:1809.01984*, 2018.

[22] K. Narasimhan, T. Kulkarni, and R. Barzilay. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*, 2015.

[23] J. Oh, S. Singh, H. Lee, and P. Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2661–2670. JMLR. org, 2017.

[24] O. Pietquin, M. Geist, S. Chandramohan, and H. Frezza-Buet. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):7, 2011.

[25] V. Rieser and O. Lemon. *Reinforcement learning for adaptive dialogue systems: a data-driven methodology for dialogue management and natural language generation*. Springer Science & Business Media, 2011.

[26] C. Sankar and S. Ravi. Deep reinforcement learning for modeling chit-chat dialog with discrete attributes. *arXiv preprint arXiv:1907.02848*, 2019.

[27] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[28] S. Singh, D. Litman, M. Kearns, and M. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133, 2002.

[29] S. P. Singh, M. J. Kearns, D. J. Litman, and M. A. Walker. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*, pages 956–962, 2000.

[30] J. Tang, T. Zhao, C. Xiong, X. Liang, E. Xing, and Z. Hu. Target-guided open-domain conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5624–5634, 2019.

[31] J. Urbanek, A. Fan, S. Karamcheti, S. Jain, S. Humeau, E. Dinan, T. Rocktäschel, D. Kiela, A. Szlam, and J. Weston. Learning to speak and act in a fantasy text adventure game. *arXiv preprint arXiv:1903.03094*, 2019.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[33] D. Yarats and M. Lewis. Hierarchical text generation and planning for strategic dialogue. *arXiv preprint arXiv:1712.05846*, 2017.

[34] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

A More Detailed Related work

Chit-chat dialogue There is an increasing body of work in the domain of chit-chat, where the primary approaches being currently tried are end-to-end neural approaches. They are typically large pre-trained and then fine-tuned transformers, either generative or retrieval. Retrieval models work best, or match generative models, on a number of tasks [34, 5, 19]. Our work shares a commonality with these approaches in that the original LIGHT dialogue data we use has no specified goals, and humans chit-chat together (and act). Thus, the conversations cover a rich number of diverse topics. In [31] models were trained in a similar fashion to chit-chat task models, and we adopt similar architectures here, but instead adapt them to learn to pursue goals.

Goal-oriented dialogue Traditional goal-oriented dialogue has focused on narrow tasks that would typically be useful for a dialogue-based assistant, for example restaurant [11], taxi, train, and hotel [2] or trip [7] booking. Hence, each task typically focuses on a narrow slice of natural language and world knowledge for a specialized domain. Earlier work focused on labeled state representations, slot filling mechanisms and dialogue managers [25], and more recent work has shifted to an end-to-end approach [1], in line with chit-chat models, but still the two sets of tasks are rarely considered together, or by using the same methods. Recently, [30] used coarse-grained keywords as targets for open-domain chit-chat but in this work the target can be achieved when either the human or the agent uses the keyword in the response.

RL for dialogue The classical goal-oriented dialogue literature studies RL extensively [29]. Typically, they used RL to improve dialogue managers, which manage transitions between dialogue states [28, 24, 25, 9, 8]. Recent works have focused more on end-to-end learning. Some works have focused on self-play type mechanisms for end-to-end reinforcement learning, where the reward is derived from the goal. A related approach to ours is the negotiation task of [17, 33], which requires two agents to swap 3 item types (hats, balls, books) where the value of the items is different for the two agents, and derives their personal reward. In contrast, our setup encompasses a rich world of settings and characters – with 3462 object types, and a corresponding large number of actions. This is reflected in the vocabulary size itself ($\sim$32,000 versus $\sim$2,000 in the negotiation tasks). Other notable uses of RL in dialogue include within visual question answering [4], in the domain of chit-chat where RL has been used to decrease repetitive and generic responses through the the use of self-play [18], and through human-bot conversation [26].

RL for language and games RL is used extensively for learning to play games, one of the most well known examples being AlphaGo [27]. Since then, language in games has started to be more deeply explored, for example in graphical games such as Minecraft [23], Real-time strategy war games [12], or in text adventure games [22, 3]. The latter are related to our setting. However, those approaches use RL to optimize the set of actions given feedback in a *single-player* rather than multi-player game, so the text only refers to the environment, and there is no dialogue or actions from other agents. We focus on the latter.

Self-Play and Language Self-play has started to become more and more widely used in NLP in general recently, see e.g. [20, 15, 10].

B Topic RL Model Further Details

Optimizing all the parameters of a large transformer architecture by RL is both incredibly costly in data efficiency and computing time, and is also known to have the problem of language drift [16] – that is, there is no guarantee after training with self-chat that the models will output recognizable natural language utterances. A solution to both problems is to train most of the parameters of the model with human-human language data, and then to either disentangle or only optimize some of the parameters with model self-chat [33]. Here, we propose a straight-forward model for that purpose. We assume an RL agent that consists of two components.

The first component $F_C(\mathcal{O}) = P_C(T_s(\mathcal{O}))$ maps from an observation to a discrete variable with C possible values. It consists of a chain of two functions: a transformer T_s that takes in the observation, and outputs a state representation $\tilde{s}$, and a policy chooser $c = P(\tilde{s}) \in (1, \dots, C)$ which takes in the state representation and outputs the value of the discrete latent variable.

Model	Goal Type	Test Seen			Test Unseen		
		$(n=1)$	$(n=3)$		$(n=1)$	$(n=3)$	
		Reward	Reward	Turns	Reward	Reward	Turns
Random Utterance	game act	0.183	0.349	2.54	0.161	0.344	2.57
Inverse model (no goal)	game act	0.185	0.345	2.55	0.160	0.345	2.57
Inverse model	game act	0.223	0.414	2.42	0.193	0.410	2.48
Top-K-TF RL	game act	**0.402**	0.537	2.18	**0.331**	0.449	2.35
Top-K-BE RL	game act	0.327	0.491	2.26	0.278	0.442	2.34
Topic RL	game act	0.359	**0.561**	2.15	0.313	**0.496**	2.26
Top-K-TF RL (1-step 3x)	game act	-	0.526	2.14	-	0.475	2.26
Topic RL (1-step 3x)	game act	-	0.493	2.22	-	0.479	2.29
Random Utterance	emote	0.086	0.200	2.79	0.061	0.185	2.81
Inverse model (no goal)	emote	0.072	0.219	2.77	0.075	0.212	2.78
Inverse model	emote	0.089	0.262	2.72	0.088	0.266	2.74
Top-K-TF RL	emote	0.166	0.400	2.55	0.131	0.349	2.59
Top-K-BE RL	emote	0.219	**0.485**	2.46	0.171	**0.436**	2.53
Topic RL	emote	**0.247**	0.482	2.43	**0.208**	0.427	2.49
Top-K-TF RL (1-step 3x)	emote	-	0.336	2.58	-	0.293	2.65
Topic RL (1-step 3x)	emote	-	0.406	2.42	-	0.348	2.50

Table 3: Results on the test seen and unseen environments for our models.

Self: guard	**Partner:** archer	**Self:** swimmer	**Partner:** turtles
Persona:	I guard the castle. I guard the king. I would kill to protect the royal family	**Persona:**	I am a huge fan of deep sea exploration, but I take any chance I can get to go for a swim...
Setting:	The armory, Inside Tower. The near top of the tower 6 feet before the very top. Where the watchers keep their eye...	**Setting:**	Bank, Swamp This is a grassy area that surrounds much of the swamp. It's a plain field with some trees nearby along...
$\mathbf{U}_0^{\text{player}}$	This is the armory! The king keeps the best weapons here. Take a look -	$\mathbf{U}_0^{\text{player}}$	Just keep taking good care of your beautiful little turtle family! Your species is quite unique and I love to see you about when I go for a swim.
$\mathbf{U}_0^{\text{env}}$	Hello, I need to get into the palace to see the king. I think he might like to see these weapons.	$\mathbf{U}_0^{\text{env}}$	Well, thank you for that. Do you happen to know where my other turtle friend is? You haven't captured any turtles have you?
$\mathbf{A}_0^{\text{env}}$	get weapon	$\mathbf{A}_0^{\text{env}}$	hug swimmer

Self: townsperson	**Partner:** villager	**Self:** songbird	**Partner:** wasp
Persona:	We are the people who live in this town. We are common, and there are many...	**Persona:**	I fly high and bring beautiful music to the people. I soar high and low going where the ...
Setting:	The Lagoon, Lake The Lagoon is a dark and mysterious place during the night hours. A lot of moss and lily...	**Setting:**	Meadow, Countryside Large clear outdoor meadow. Flowers of blue and white appearing in bunches here and there. The ...
$\mathbf{U}_0^{\text{player}}$	It is cold up here. Would you like my coat	$\mathbf{U}_0^{\text{player}}$	Get out of here, wasp!
$\mathbf{U}_0^{\text{env}}$	Oh yes please if I may. My shoe has become sodden from running to the market I should love to dry it a bit.	$\mathbf{U}_0^{\text{env}}$	You? Fly away from me? You're in my forest, bird. I control this land.
$\mathbf{A}_0^{\text{env}}$	remove Cloak	$\mathbf{A}_0^{\text{env}}$	hit a songbird

Table 4: Example 1-step episodes where after the RL agent's utterance $\mathbf{U}_0^{\text{player}}$ the environment agent's response action $\mathbf{A}_0^{\text{env}}$ was equal to the RL agent's goal $\mathbf{g}$. Our RL agent both makes natural utterances given the situation, and that elicit the desired goal.

The second component $T_u(\mathcal{O}, c)$ is an additional transformer that takes as input the observation as well as the output of the first component, and outputs a dialogue utterance. The entire model is thus the chain $u = T_u(\mathcal{O}, P_C(T_s(\mathcal{O})))$. We make this explicit decomposition so that we can train only part of the model with RL; note that the "action" trained via RL is choosing c, not outputting the final utterance.

Initial topics We first pre-train the transformer T_s using the inverse model described in Section 4.1, which produces a vectorial representation of a given observation. We then run K-means over the vectorial representations of all observations from the training set to provide the mapping to one of C values, which represent dialogue topics, which we use as our initial function $P_C(\tilde{s})$. These two

Model	Goal	Train (n = 1) Reward	(n = 3) Reward	Turns
Top-K-TF RL	act	0.677	0.752	1.72
Topic RL	act	0.539	0.752	1.87
Top-K-TF RL (1-st. 3x)	act	-	0.737	1.62
Topic RL (1-st. 3x)	act	-	0.660	1.87
Top-K-TF RL	emote	0.498	0.668	2.13
Topic RL	emote	0.483	0.612	2.22
Top-K-TF RL (1-st. 3x)	emote	-	0.587	1.96
Topic RL (1-st. 3x)	emote	-	0.570	1.99

Table 5: Results on the training environment

		1-Step		**1-Step 3x**		**3-Step**	
Verb	Count	Topic	Top-K	Topic	Top-K	Topic	Top-K
get	213	27.70	28.17	37.56	43.66	**44.13**	40.85
hit	172	43.02	46.51	63.95	66.86	63.95	**75.58**
hug	178	61.26	69.82	72.52	81.53	85.13	**85.56**
give	136	33.09	41.91	50.00	54.41	**56.62**	48.53
remove	127	9.45	13.39	22.83	22.83	**27.56**	26.77
steal	55	47.27	50.91	63.64	63.64	**80.00**	54.55
drop	27	0.00	0.00	**18.52**	**18.52**	7.41	7.41
put	25	0.00	0.00	8.00	**12.00**	4.00	4.00
eat	10	30.00	10.00	**70.00**	20.00	60.00	40.00
wear	10	0.00	0.00	20.00	**30.00**	20.00	10.00
drink	3	33.33	33.33	33.33	33.33	33.33	33.33

Table 6: Verb success in percentage on 1000 test seen episodes. The 3-step model performs best for high and medium frequency verbs.

functions together give us our initialization of F_C. We use the set of topics as a set of actions $\mathcal{A}$ for our RL setup. [2]

From c to $\mathcal{A}$ Given our initial choice of F_C, we can also pre-train T_u. We simply take our initial human-human training data, and for each observation append the topic computed by F_c to it. This allows our model to be able to generate an action (utterance) conditional on both an input and a topic. We can now train a policy by RL that optimizes the topic at any given point in the episode.

Policy training We keep the pre-trained portions of the model T_u and T_s fixed and during fine-tuning only optimize P_C. The cluster chooser P_C is redefined (from the initial K-means) to be an MLP network consisting of 2 layers. A discrete action is sampled from a categorical probability distribution over the possible topics, given by $\mathbf{c_t} \sim \text{Categorical}(\mathbf{h_t^2})$, where $\mathbf{h_t^2} = \tanh(\mathbf{W_2}\tanh(\mathbf{W_1 s_t} + b_1) + b_2)$.

The state vector $\mathbf{s_t}$ also encodes the goal $\mathbf{g}$ and thus, the policy is conditioned on the goal $\mathbf{g}$ of the agent. Hence, the policy can learn strategies that will result in picking actions at each time step $\mathbf{t}$ that will help the agent to achieve its goal $\mathbf{g}$. As our RL agent can only choose topics, it cannnot redefine easily the meaning of words to cause language drift. We use the Advantage Actor-Critic implementation **A2C** [14] to train the policy and the value function in both this and the subsequently described Top-K model.

[2] We show the clusters denoted by their topics along with the most representative sentences in Table 9 in Appendix D.1. We see that the learned clusters are non-random but rather correspond to specific topics.

	1-Step		
	Topic	Top-K-TF	Top-K-BE
1-step achievable	0.452	0.505	0.407
1-step unachievable	0.000	0.005	0.005

Table 7: Test seen breakdown by difficulty (1-step achievable or not).

	1-Step 3x			3-Step		
	Topic	Top-K-TF	Top-K-BE	Topic	Top-K-TF	Top-K-BE
1-step achievable	0.616	0.647	0.587	0.686	0.664	0.620
1-step unachievable	0.044	0.058	0.044	0.068	0.049	0.078

Table 8: Test seen breakdown by difficulty (1-step achievable or not). The 3-step models outperform the 1-step 3x models on both sets.

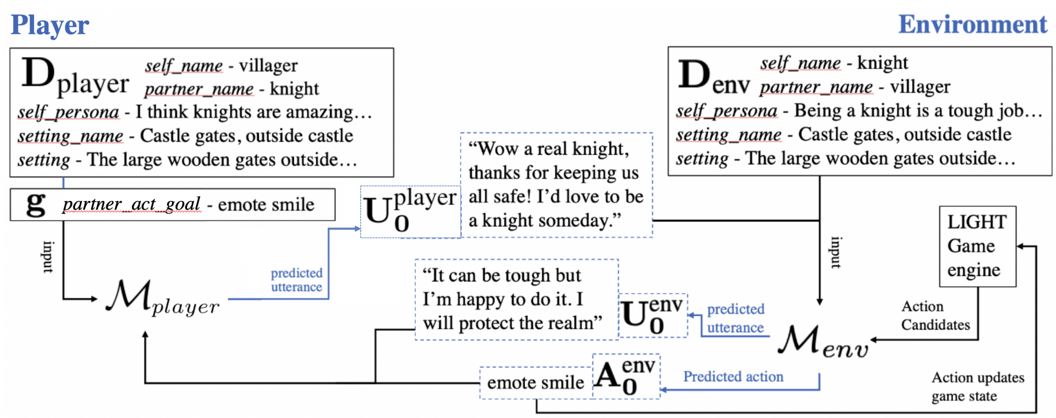


Figure 1: Example interaction in the described task setup (single turn). Here the RL agent $\mathcal{M}_{player}$ would receive a reward as the environment agent $\mathcal{M}_{env}$ took the desired action g.

C Top-K Model Further Details

The Top-K model, related to [6], is another approach to keeping the number of trainable parameters small. As above it keeps close to the base retrieval model to avoid drift. It first uses the inverse model to get a context embedding $\tilde{s}$ from the observation, and a list of K candidate utterance embeddings $v_1, ...v_K$ corresponding to utterances $u_1, ...u_K$. These are the encodings by the inverse model of the K utterances it considers most likely given the context and goal. We form scores $t_i = (A+b)^T v_i$, and obtain a probability distribution over these K candidates for our policy:

$$\pi(u_i|\text{context}) = \text{softmax}(t_0, ..., t_K)(i). \tag{3}$$

Here the trainable parameters of the RL agent are the map A and biases b.

Alternatively, we can train a small (2-layer) Transformer model T_w that takes as input the set $\{\tilde{s}, v_1, ...v_K\}$. Instead of a softmax over dot products t_i as in (3), we use the attention weights in the last layer of T_w above $\tilde{s}$ against the candidates as the distribution over the candidates for sampling an utterance. In this case, the weights of T_w are the trainable parameters of the RL agent. We call the former model a policy "bi-encoder". In the main paper we only reported in the bi-encoder results. In the appendix we label these as Top-K-BE RL in tables, and label the latter Transformer model as Top-K-TF.

D More Detailed Experimental Analysis

Analysis of utterance choice To understand the semantics the models are learning that ground language to actions, we visualize the top scoring utterances, averaged over their probabilities on the 1-step test set, broken down by verb type. We observe a clear improvement in semantic connection for the Topic RL model over the inverse model. For example utterances such as "Have a taste of this" are highly scoring for *drink* goals, “hmm..this sure smells nice” for *eat* goals, “Ew you vile beast, do not touch me! I will have you removed” for *hit* goals, and “How I love being pampered by you, sweetheart” for *hug* goals. Given there are $\sim$111,000 possible utterances in our setting, the model has clearly learned meaningful representations. Appendix Tables 13 and 14 show results for the inverse model and Topic RL model respectively.

Train vs. test performance We compare training performance of our models in Table 5. We see the same trends that models that performed better on test fit better on train (e.g. Top-K vs. Topic RL on 1-step tasks). Nevertheless, we do observe significant overfitting can occur, indicating that future work could explore either models that improve through better generalization, or by exploiting more training data – for example by self-play with more goals, rather than just using goals from human logs, as we have done here.

Model capacity We evaluate different values of K or numbers of topics for Top-K and Topic RL. Full results are given in Appendix Table 11. They show that increasing the capacity of both models improves performance up to 200 clusters or $K = 200$, after which performance saturates. However, $K = 200$ (56.1%) is substantially better than $K = 50$ (47.7%) on the 3-step task, for example.

Performance breakdown by goal We show the breakdown of test performance by goal type in Table 6 (splitting by verb type) and Appendix Table 12 (splitting by emote type). The results show that the easiest tasks are common actions with clear differentiation such as *hug* (85% success) and *hit* (75%). Actions like *get*, *drop*, *give* which are more confusable have somewhat lower numbers, with more rare actions (e.g. wear) faring worse.

Performance breakdown by difficulty We can break down the test results into difficulty by considering in the 3-step task, which examples are 1-step achievable given the model’s possible actions under the policy (i.e. the possible Top-K utterances or Topic RL cluster choices), and reporting results separately. The results are given in Table 7 and 8. They show that non 1-step achievable goals are much harder, representing a significant challenge to future systems.

1-step 3x baseline To investigate further the quality of our 3-step task models, we consider an additional baseline of taking a 1-step task trained model (Topic RL or Top-K) and applying it on the 3-step task, which it has not been optimized for. The results in Table 3 show test results are inferior for this approach. Breaking down further by goal type (Table 6 and Appendix Table 12) shows that there are large improvements for the 3-step model on goals which are more often expressed in the data. Table 7 shows that 3-step models outperform the 1-step 3x models on both 1-step achievable and the harder 1-step unachievable goals. Training performance (Table 5) further validates these results.

3-step task repeats We analyze the number of repeated utterances in an episode. The Topic RL model repeats at least one utterance 25.8% of the time, with 15.59% utterances overall repeated. The 1-step 3x baseline in comparison repeats 37.3% at least once, and 22.94% on average. We note that repeating an utterance may possibly bring the desired goal in some cases, just as in real life.

D.1 Clusters

D.2 Training Curves

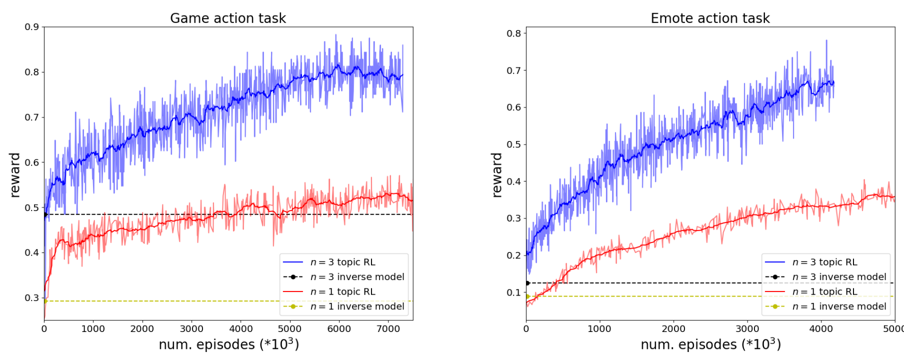
Figure 2: Topic RL model training for n=1 and n=3 step goals for game actions (left) and emotes (right), comparing to the inverse model baselines. We report rewards averaged over the batch (512 for n=1, and 128 for n=3). Darker lines indicate smoothed plots.

D.3 Hyperparameters and additional experimental details

The Topic RL models have 576969 trainable parameters, the Top-K RL have 14767105, and the Top-K Biencoder have 1181953. Training using 8 V100 machines took $\sim$2 weeks (1 step), $\sim$5 weeks (3 step). Learning rate is the only hyperparameter that was swept over. For each setup, we considered learning rate values between $3 \cdot 10^{-6}$ and $7 \cdot 10^{-3}$. Other hyperparameters (e.g. eps) were briefly examined in preliminary experiments but our setup was not very sensitive to changes in those values; hence we used the default provided values.

#C	Topic	Representative Sentences
19	animal sounds	‘Meow! Purr!’, ‘Bah-Buk! Tasty!’, ‘Woof! Very!’, ‘Bock! Bock!’
12	find the cost	‘I would love some fruit. What are your prices?’, ‘They are beautiful. How much do the cost?’, ‘It flows easily, how much are you selling it for?’
28	prayer, God	‘Then your poor life is a sign from God for you to join us in the church and serve him!’, ‘If you say so priest. From now I will pray every night for wealth and good food!’, ‘Continue to love, worship, and serve Him.’
45	ask favor	‘Yes but do you mind doing me a favor?’, ‘Since I have helped you, could you do me a favor?’, ‘If I offer to solve your problem, what will you personally do for me in return?’

Table 9: Clusters learnt from the dialogue utterances (Clusters = 50). ‘#C’ denotes the cluster ID.



Model	n (steps)	goal	Best LR	goal	Best LR
Topic RL	1	game act	7E-04	emote	7E-04
Topic RL	3	game act	7E-04	emote	7E-04
Top-K-TF RL	1	game act	7E-05	emote	3E-05
Top-K-TF RL	3	game act	1E-05	emote	1E-05
Top-K-BE RL	1	game act	1E-03	emote	1E-03
Top-K-BE RL	3	game act	1E-03	emote	1E-03

Table 10: Best learning rate values for each reported experiment.

D.4 Additional Results

			Test Seen			Test Unseen		
			$(n=1)$	$(n=3)$		$(n=1)$	$(n=3)$	
Model	**Goal Type**	# Clusters	Reward	Reward	Turns	Reward	Reward	Turns
Topic RL	game act	50	0.324	0.477	2.31	0.277	0.470	2.24
Topic RL	game act	100	0.348	0.523	2.21	0.282	0.488	2.28
Topic RL	game act	200	0.359	0.561	2.15	0.313	0.496	2.26
Topic RL	game act	500	0.362	0.505	2.23	0.307	0.46	2.35
Topic RL	game act	1000	0.372	0.510	2.20	0.333	0.464	2.32
Top-K-TF RL	game act	50	0.329	0.503	2.24	0.261	0.439	2.39
Top-K-TF RL	game act	100	0.370	0.521	2.12	0.292	0.468	2.33
Top-K-TF RL	game act	200	0.402	0.537	2.18	0.331	0.449	2.35
Top-K-TF RL	game act	500	0.402	-	-	0.299	-	-
Top-K-TF RL	game act	1000	0.426	-	-	0.337	-	-

Table 11: Results with different numbers of clusters (Topic RL) or candidates (Top-K RL). Some experiments were not completed because of resource limitations.

		1-Step		**1-Step 3x**		**3-Step**	
Emote	Count	Topic	Top-K	Topic	Top-K	Topic	Top-K
laugh	109	20.18	11.01	32.11	20.18	**44.04**	26.61
smile	106	31.13	13.21	58.49	37.74	**61.32**	44.34
ponder	94	31.91	2.13	44.68	7.45	**59.57**	24.47
frown	85	18.82	9.41	29.41	21.18	**34.12**	24.71
nod	75	40.00	21.33	58.67	52.00	**84.00**	56.00
sigh	67	55.22	4.48	82.09	14.93	**85.07**	11.94
grin	63	4.76	1.59	25.40	12.70	**33.33**	26.98
gasp	57	21.05	0.00	**33.33**	0.00	**33.33**	3.51
shrug	47	29.79	6.38	51.06	48.94	**59.57**	48.94
stare	41	7.32	4.88	**26.83**	17.07	**26.83**	9.76
scream	40	17.50	20.00	25.00	25.00	**42.50**	30.00
cry	32	12.50	28.13	18.75	50.00	43.75	**56.25**
growl	27	40.74	37.04	**48.15**	40.74	33.33	40.74
blush	26	3.85	19.23	11.54	50.00	19.23	**53.85**
dance	24	37.50	29.17	**62.50**	33.33	**62.50**	33.33
applaud	23	17.39	0.00	**43.48**	21.74	21.74	21.74
wave	19	21.05	21.05	**36.84**	21.05	10.53	26.32
groan	17	5.88	0.00	**17.65**	11.76	11.76	5.88
nudge	16	0.00	0.00	0.00	6.25	0.00	**12.50**
wink	15	13.33	20.00	13.33	33.33	13.33	**53.33**
yawn	11	0.00	0.00	0.00	18.18	**27.27**	**27.27**
pout	6	0.00	33.33	16.67	**66.67**	16.67	16.67

Table 12: Emote success in percentage on 1000 test seen episodes. The 3-step model performs best for high and medium frequency verbs.

Verb	count	Top utterances
get	213	'Why hello there, I haven;t seen you in awhile.', "Oh hello, I didn't expect to find anyone else here.", "Well hello there, wasn't expecting to see you here.", 'Wow! What a fine place this is.', "Oh, hello! I didn't see you all here.", 'Well hello there! I did not expect to see anyone here.', "Isn't this place so wonderful!?", 'I need some light.', 'So how is buisiness going?', '"Ah, what a long day we have ahead of us!"'
put	25	'Why hello there, I haven;t seen you in awhile.', "Well hello there, wasn't expecting to see you here.", "Oh hello, I didn't expect to find anyone else here.", 'Wow! What a fine place this is.', 'Eerie. I must light a candle. And say a prayer', "Oh, hello! I didn't see you all here.", 'Well hello there! I did not expect to see anyone here.', "Isn't this place so wonderful!?", 'Greetings! How are my subjects doing this fine day?', 'Good morning. Someone needs to tend to this rickety rectory. I almost fell through the floor.'
drink	3	'Eerie. I must light a candle. And say a prayer', 'It is a wonderful day to drink! Time to get my drunk on!', 'I need another drink.', "Greetings m'lord! Cold day isn't it?", 'I am person just trying to enjoy the ambiance of this room', 'I need some light.', 'It appears you need some guidance.', 'Hello person! How are you on this fine evening?', 'Good evening good evening sir! Can I help you?', "Well hello there, wasn't expecting to see you here."
eat	10	'Why hello there, I haven;t seen you in awhile.', 'Hello bird, how are you doing?', 'Ahh, what a great day to nibble at the feet of humans.', 'I hope there is food in here.', 'Mmmm a human come into my territory. My lucky day indeed.', 'Ugh I am so tired of being used as food around here.', 'I am so delighted to not have to scavenge for food in the village.', 'WOW! So much food to eat here', '"Come here! I need to eat!"', 'man i hope i can find something to eat here'
steal	55	'well what a fine mess i have gotten myself into this time', '*ARGH* you must let me out of this place.', 'I have seen you before! Thief what is it you think you will get today?', 'Wow, this lavoratory is filthy!', 'Hey, you there. Come here!', 'Hey, you over there! You look like you could use a little something I have.', 'Hello! You look as though you are in need of some of my wares.', 'It appears you need some guidance.', 'Why hello there, I haven;t seen you in awhile.', 'Enjoy! You finally have a place of your very own.'
hit	172	'Whatchit! You almost crushed me!', '*ARGH* you must let me out of this place.', 'Hey, you there. Come here!', 'well what a fine mess i have gotten myself into this time', 'Wow, this lavoratory is filthy!', 'You must bow before me.', 'Why are you in here! Back away from me or I will strike!', 'Why hello there, I haven;t seen you in awhile.', '"Come here! I need to eat!"', 'Ugh not another one of these beasts.'
hug	222	'Why hello there, I haven;t seen you in awhile.', 'Minister! It is so good to see you!', "Well hello there, wasn't expecting to see you here.", "Oh hello, I didn't expect to find anyone else here.", "I'm so glad you're here with me", 'It is so nice and warm in here.', 'Wow! What a fine place this is.', 'I am so happy for this day.Even if is in this filthy place', "Oh, hello! I didn't see you are.", 'Hail, friend. How are things?'
wear	10	'Why hello there, I haven;t seen you in awhile.', "Well hello there, wasn't expecting to see you here.", "Oh hello, I didn't expect to find anyone else here.", 'Wow! What a fine place this is.', 'Good afternoon sir! I did not expect to find you here.', 'Well hello there! I did not expect to see anyone here.', 'Why I did not expect to see you here, sir! Please join us.', 'Good evening good evening sir! Can I help you?', 'It appears you need some guidance.', '"Ah, what a long day we have ahead of us!"'
drop	27	"Well hello there, wasn't expecting to see you here.", 'Why hello there, I haven;t seen you in awhile.', "Oh hello, I didn't expect to find anyone else here.", 'Wow! What a fine place this is.', "Oh, hello! I didn't see you all here.", 'Well hello there! I did not expect to see anyone here.', '"Ah, what a long day we have ahead of us!"', 'well what a fine mess i have gotten myself into this time', 'Oh, hello! I was just checking to see if anyone dropped these goblets. Ha, ha, ha.', 'So how is buisiness going?'
give	136	'Why hello there, I haven;t seen you in awhile.', "Well hello there, wasn't expecting to see you here.", 'Wow! What a fine place this is.', "Oh hello, I didn't expect to find anyone else here.", 'Good evening good evening sir! Can I help you?', "Isn't this place so wonderful!?", 'Well hello there! I did not expect to see anyone here.', "Oh, hello! I didn't see you all here.", 'Wow this is such a nice place.', 'I must get this place cleaned at once!'
remove	127	"Well hello there, wasn't expecting to see you here.", 'Why hello there, I haven;t seen you in awhile.', "Oh hello, I didn't expect to find anyone else here.", "Oh, hello! I didn't see you all here.", 'Wow! What a fine place this is.', 'Well hello there! I did not expect to see anyone here.', 'It appears you need some guidance.', 'Good evening good evening sir! Can I help you?', 'Another hectic day in this place.', '"Ah, what a long day we have ahead of us!"'

Table 13: Top utterances for each verb for the inverse model.

Verb	count	Top utterances
get	213	'Here sir, I found this.', 'Oh hello there brothers! Why whose towel is this thats left all by its self?', 'How did this get here?', 'Meh. Whats this you have here?', "What is this? Is this someone's head?!", "Thank you, sir. What's with all this silk?", 'What is this here?', 'It looks like there is something missing!', "Oh, look, somethin' shinny", 'what is this ston slab'
put	25	'How did this get here?', 'Oh hello there brothers! Why whose towel is this thats left all by its self?', 'Where did you find this?', 'Ah.... I wonder what this doll looked like before...', "Thank you, sir. What's with all this silk?", 'Wait... one... MOMENT. What is my royal CUP doing in here?', 'Here sir, I found this.', 'What is this room here for? Miaow!', 'Have you noticed this artwork on this wood maam?', 'So you decided to look at this one?'
drink	3	'Oh, what is this? It smells heavenly!', "What's that stuff? Smells good.", 'hmmnnnn.. this sure smells nice', 'Hello monk, that incense smells amazing.', 'I wish I can just have a taste of that', 'Do you smell that? It smells DIVINE!', 'I wonder how this tastes?', 'Hmmnnn... This smells great!', 'Have a taste of this', 'Where did you get this? I could use a smoke afterwards!'
eat	10	'Oh, what is this? It smells heavenly!', "Hmmm, sniff. This doesn't smell edible.", 'Something in here smells good...I hope I can eat it.', 'I wonder how this tastes?', "What's that stuff? Smells good.", 'I wish I can just have a taste of that', 'hmmnnnn.. this sure smells nice', 'Ew this is disgusting. Even for me.', 'Mmm look at all this delicious trash.', 'Hmmnnn... This smells great!'
steal	55	'"Hey! I think you dropped this!"', 'How did this get here?', 'Here sir, I found this.', 'Wow, where were you hiding this?', 'What about this! Is this yours or was it already here?!', "What is this? Is this someone's head ?!", 'Where did you find this?', 'Tell me where you found this!', 'Where did you steal that from?', 'See this? Do you think I just found this laying around some house?'
hit	172	'Foul scourge! How dare you bring your taint here!', 'Ooooh, how horrid! Away with you you filthy creature! GUARDS! GUARDS!', 'You come to my place and are trying to take my land! Is that what you are doing? You dirty scumbag!', 'Why are you in here! Back away from me or I will strike!', 'Ew you vile beast, do not touch me! I will have you removed!', 'GUARD! Get this scum off of me at once. How dare you, you scoundril!', 'Be gone you foul beast!', 'Quickly?! You started this you repugnant beast of a man!', 'I want out! this place is evil.', 'How dare someone of your low status attack me?? Have at you, you maggot!'
hug	222	'he loves me so much', 'ahhhh i love you to dear', 'How I love being pampered by you, sweetheart!', "Aw you are so cute I can't resist cuddling with you", "I'm so glad to be here in everyone's company.", 'awww. I love you child', 'Oh how i have missed you.', 'I love you so dang much.', 'Lord of Light, I adore you.', "I'm so happy to be here today"
wear	10	'Here sir, I found this.', 'Like this broken weapon here?', 'Oh hello there brothers! Why whose towel is this thats left all by its self?', 'Hello my king, do you know where this weapon came from?', 'Here sir...you dropped this...you may need it.', "Thank you, sir. What's with all this silk?", 'Meh. Whats this you have here?', 'How did this get here?', 'Meow. I need this hay', 'Are you here to purchase that amazing blue knight armor sir?'
drop	27	'Here sir, I found this.', 'How did this get here?', "Oh, look, somethin' shinny", 'Oh hello there brothers! Why whose towel is this thats left all by its self?', "Thank you, sir. What's with all this silk?", 'It looks like there is something missing!', 'What is this here?', 'I heard theres some valuable stuff in here mate, know anything about that?', 'Meh. Whats this you have here?', "Let's stuff it here!"
give	136	'Here sir, I found this.', 'Meh. Whats this you have here?', 'Wow, this looks to be very old. Where is it from?', "My goodness I wonder how that got there! It sure is pretty isn't it?", 'Say, where did you get this?!', 'Oh hello there brothers! Why whose towel is this thats left all by its self?', 'Someone left this bag in this pew. Do you know what it is?', 'Tell me where you found this!', "What is this? Is this someone's head?!", 'what is this ston slab'
remove	127	'I suppose for today we may as well look at some garbs.', 'Hey there! Got time to take a look at something?', "Thank you, sir. What's with all this silk?", 'Hmm, where am i and why is everything so sharp?', 'Ah, squire Lawrence. Did you polish my armor?', 'What are you jotting down, sir?', 'Hello ratty. I am looking to clean my clothes!', 'Yes sir what is this good news? Did you finally get me a new dress!?', 'At least my hat is clean.', "Oh, hello there. Pardon my, erm, dusty appearance. It's been quite journey to get even this far!"

Table 14: Top utterances for each verb for the Topic RL model.

D.5 Game actions within LIGHT

Action	Constraints	Outcome
get *object*	actor and *object* in same room *object* is gettable	actor is carrying *object*
drop *object*	actor is carrying *object* *object* is gettable	*object* is in room
get *object1* from *object2*	Actor and *object2* in same room *object1* is gettable *object2* is surface or container *object2* is carrying *object1*	actor is carrying *object1*
put *object1* in/on *object2*	Actor and *object2* in same room *object2* is container or surface actor is carrying *object1*	*object2* is carrying *object1*
give *object* to *agent*	Actor and *agent* in same room *object* is a member of actor	*agent* is carrying *object*
steal *object* from *agent*	actor and *agent* in same room *object* is a member of *agent*	actor is carrying *object*
hit *agent*	Actor and *agent* in same room	inform *agent* of attack
hug *agent*	Actor and *agent* in same room	inform *agent* of hug
drink *object*	actor is carrying *object* *object* is a drink	inform actor of drinking successfully
eat *object*	actor is carrying *object* *object* is a food	inform actor of eating successfully
wear *object*	actor is carrying *object* *object* is wearable	actor is wearing *object*
wield *object*	actor is carrying *object* *object* is a weapon	actor is wielding *object*
remove *object*	actor is wearing/wielding *object* *object* is wearable or a weapon	actor is carrying *object*

Table 15: LIGHT actions and constraints from [31]

D.6 LIGHT example

Persona 1: Village youth	**Persona 2: Horse**
I am a youth who lives in the village. I am adventurous and curious about everything. I get bored easily, so I will always try to slip away from mundane daily tasks to find something more interesting to do. I am smart for my age and a bit smug.	I live on a farm. I work for humans. I like hay.

Setting: Royal stables, Outside Castle - The royal stables hold all of the horses of the noble people living in the castle. They are very well constructed, consisting of expert mason work and fixtures of some sort of heavy, forest-scented wood. Each horse has a large stall, lined with fresh hay replenished by stable hands daily. On the walls of stable, there are tools used to care for and groom the horses, hung on hooks. The stables are lit by lanterns hung throughout the stables.

Village Youth This is so boring cleaning out the stables every day	
Horse Thank you for taking such good care of us	
Village Youth I'll be getting you some treats soon as well	
Horse Mmm. It's also nice to have your company.	
Village Youth I wonder what it'd be like to just take one of these horses and roam the countryside	
Horse Let's go. It's been a while since I've been taken out to stretch my legs	*Emote:* nudge
Village Youth Yeah, let's break out of here. The royalty don't even care about the horses like I do	
Horse It's starting to get dark. I think we will need this lantern.	*Action:* get lantern
Village Youth Good idea. I do need some supplies before we go	*Action:* get Tool
Horse I'll probably want a snack while we are gone.	*Action:* get wheat
Village Youth Let's see, what else do I need?	*Emote:* ponder
Horse Maybe the wood for a fire?	*Action:* give lantern to village youth
Servant: Yeah, we'll need a bundle for sure	*Action:* get wood
Horse I've got the hay. I think we are ready to go. Let's get out of here	*Action:* get hay

Figure 3: Example episode from the LIGHT dataset, consisting of an environment (location setting, characters with given personas, objects), utterances and game actions. There are 10,777 such human-human gameplay episodes, and a rich world of 663 locations, 1755 characters and 3462 objects.

D.7 1-Step and 3-Step Episode Examples

Self: a cowardly young man in armour	**Partner:** guard	**Self:** bodyguard	**Partner:** intruder
Persona:	I have just been trained as a royal soldier. I am 18 years old and terrified...	**Persona:**	I am an immortal bodyguard. The gods have appointed me to protect the king...
Setting:	Trash Heap, Wasteland A largest trash heap in the kingdom has been burned out so many times that it no longer resembles anything...	**Setting:**	Treasure Cavern, Inside Temple Glittering as far as the eye can see the Treasure Cavern is filled with gold, silver, precious gems,...
$\mathbf{U}_0^{\text{player}}$	I'm also in need of a new shield.	$\mathbf{U}_0^{\text{player}}$	Step back intruder! You have no business in the king's treasure cavern!
$\mathbf{U}_0^{\text{env}}$	Squire, my shield fatigues me.	$\mathbf{U}_0^{\text{env}}$	Ha! I'm here to take all of this treasure. End the king's reign!
$\mathbf{A}_0^{\text{env}}$	hug a cowardly young man in armour	$\mathbf{A}_0^{\text{env}}$	get gold
$\mathbf{U}_0^{\text{player}}$	Thank you, sir. I needed a hug.	$\mathbf{U}_0^{\text{player}}$	You come to my place and are trying to take my land! Is that what you are doing? You dirty scumbag!
$\mathbf{U}_0^{\text{env}}$	Yes. I need you to hold this shield for me.	$\mathbf{U}_0^{\text{env}}$	Then I will get away with your gold!
$\mathbf{A}_0^{\text{env}}$	remove shield	$\mathbf{A}_0^{\text{env}}$	hit bodyguard

Table 16: Successful 3-step episodes. On the left: the topic-RL agent's goal was to get the environment agent to remove shield. On the right: the topic-RL agent's goal was to get the environment agent to hit the topic-RL agent. In both episodes, the topic-RL agent makes natural utterances given the situation that elicit the desired goal in 2 turns.

Self: cat	**Partner:** challenger	**Self:** peasant	**Partner:** the man
Persona:	I live in the barn of a small farm. I protect the farm from pests...	**Persona:**	I am poor and dirty. I hate that I am starving to death...
Setting:	The Dungeon, Inside Palace The dungeon is in the very most bottom room of the Palace. Many have gone down to the dungeon...	**Setting:**	Cottage, Countryside The small cottage was white with two, shuttered windows. It was in the unique shape of a...
Goal:	drop poison	**Goal:**	put coin in dinner table
$\mathbf{U}_0^{\text{player}}$	What's that stuff? Smells good.	$\mathbf{U}_0^{\text{player}}$	Oh, what is this? It smells heavenly!
$\mathbf{U}_0^{\text{env}}$	That's smells like the good stuff alright!	$\mathbf{U}_0^{\text{env}}$	Yes it is top quality.
$\mathbf{A}_0^{\text{env}}$	give poison to cat	$\mathbf{A}_0^{\text{env}}$	steal moldy bread from peasant
$\mathbf{U}_0^{\text{player}}$	i thought so too, well that's gone. What are you up to?	$\mathbf{U}_0^{\text{player}}$	Excellent! What have you been doing out here on this fine day?
$\mathbf{U}_0^{\text{env}}$	ive been hidden waiting for the perfect time to make an escape	$\mathbf{U}_0^{\text{env}}$	I have no bread. I have just come to get some rest before moving on.
$\mathbf{A}_0^{\text{env}}$	remove bag of herbs	$\mathbf{A}_0^{\text{env}}$	eat moldy bread
$\mathbf{U}_0^{\text{player}}$	I thought you might be eating my herb again.	$\mathbf{U}_0^{\text{player}}$	No, but perhaps you can use it to cleanse your bread!
$\mathbf{U}_0^{\text{env}}$	I did it, I confess! I stole these precious herbs.	$\mathbf{U}_0^{\text{env}}$	I would love to. I fear I look disheveled. I will wear this blanket to cover my filthy clothes
$\mathbf{A}_0^{\text{env}}$	drop bag of herbs	$\mathbf{A}_0^{\text{env}}$	hug peasant

Table 17: Unsuccessful 3-step episodes. On the left: the topic-RL agent's goal was to get the environment agent to drop poison. On the right: the topic-RL agent's goal was to get the environment agent to put coin in dinner table. In both episodes, the topic-RL agent both makes natural utterances given the situation, but does not manage to achieve its goal.