Interactive Reinforcement Learning for Object Grounding via Self-Talking

Yan Zhu[1*] **Shaoting Zhang**[2] **Dimitris Metaxas**[1]
[1]Rutgers University [2]Baidu Research
[1]{yz328, dnm}@cs.rutgers.edu [2]zhangshaoting@baidu.com

Abstract

Humans are able to identify a referred visual object in a complex scene via a few rounds of natural language communications. Success communication requires both parties to engage and learn to adapt for each other.

In this paper, we introduce an interactive training method to improve the natural language conversation system for a visual grounding task. During interactive training, both agents are reinforced by the guidance from a common reward function. The parametrized reward function also cooperatively updates itself via interactions, and contribute to accomplishing the task. We evaluate the method on GuessWhat?! visual grounding task, and significantly improve the task success rate. However, we observe language drifting problem during training and propose to use reward engineering to improve the interpretability for the generated conversations. Our result also indicates evaluating goal-ended visual conversation tasks require semantic relevant metrics beyond task success rate.

1 Introduction

Natural language interaction is probably the most efficient and natural way for humans to acquire knowledge and exchange information. In artificial intelligence, developing intelligent agents that can correspond the linguistic concepts with the visual sensor inputs, and communicate via goal-ended dialogue is also a fundamental problem.

Recently, two visual grounded conversation tasks along with datasets/environments are proposed: VisDial [2] and GuessWhat?! [4]. VisDial [2] collects a large dataset containing free-form conversations about natural images. Based on VisDial dataset, Das *et al*[3] further proposed an *image level* grounding task and applied reinforcement learning to train a goal-ended conversation system. The generated conversation was mainly evaluated on an image retrieval task.

Instead, GuessWhat?! proposed an environment focusing on *object instance level* grounding. The object grounding task implicitly requires agents detect and recognize object instances, understand the spatial layout, and use positional / attribute word to reason and distinguish among candidate regions. In this paper, we focus on GuessWhat?! task, since object level grounding is more practically relevant to real-world applications, for instance, interactive navigation robots.

Beyond supervised training using human dialog, reinforcement learning has recently been adopted in visual conversation [3, 10]. Particularly, on GuessWhat?! task, Strub *et al*[10] used RL to tune the question generator while keeping answer agent and guesser unchanged. However, human collaboration normally requires both parties to engage and adapt for the other. In this paper, we propose to interactive train all three models in a dynamic environment. The question generator and the answer models are collectively tuned by a common reward function using reinforcement learning.

[*] Yan Zhu works at Facebook Applied Machine Learning now. The majority of the work was done at Rutgers.

NIPS 2017 Workshop on Visually-Grounded Interaction and Language (ViGIL), Long Beach, CA, USA.

The reward function, which is parameterized by the guesser model, is also dynamically updated to cooperate with other two models. Our result significantly outperforms the previous best result on GuessWhat?! task and achieves near human performance. Despite improved task success rate, we observe the generated conversations suffer from language drifting problem. We also propose a reward engineering technique to help improve interpretability of the generated conversations.

Our main result shows that agents are able to achieve near human level performance on visual object grounding task, by drifting from natural language towards a contrived language. It also indicates current goal-ended visual conversation task requires more semantic related metrics for evaluation, other than task completion rate. In summary, the contribution of this paper is two-fold:

- We introduce an interactive training method for object instance grounding task. The proposed training method significantly outperforms previous best results on GuessWhat?! task.
- To balance between interpretability and task success rate, we propose a reward engineering technique to interfere training, and improves the readability of the generated conversations.

2 Related Work

Visual Conversation: Existing visual conversation datasets can be divided into task-oriented dialogue [4] and free-form (chit-chat) dialogue [2]. Recently, Das *et al*[3] also introduced a goal-ended conversation task based on VisDial. One salient difference is that GuessWhat?! focuses on *object instance level* grounding while the task in [3] focus on *image level* grounding. Beyond supervised training baselines, Das *et al*. [3] showed cooperative RL improves supervised trained baselines in VisDial image retrieval task. Also on GuessWhat?!, Strub *et al*. [10] also showed that RL could improve task success rate by only tuning the question generator and keeping other two models static. Notably, the reward function in GuessWhat?! task depends on the *subjective* parametric guesser model rather than an objective metric. In this work, we focus on the object grounding task in GuessWhat?!, and extend reinforcement training towards a more interactive setting: both conversation bots are collectively trained using RL, and the parameterized reward function (the guesser model) is also actively involved in the updating dynamics.

Artificial Language vs Natural Language: Recently, [3, 6, 8] found that multiple agents can develop their own communication protocols (artificial languages) during cooperative training. The emerged language is very effective between AI agents, but not interpretable for humans [1]. To narrow the semantic gap between artificial language and natural language, [7, 9] explored different techniques to retain interpretability, including constrain vocabulary size [7] and iteratively updating different agents [9]. These findings are based on synthetic environments, rather than natural image based tasks. To echo these findings, we observe similar language drifting problems during interactive training. To improve the interpretability, we choose to explicitly enforce the desired dialogue properties in the reward function, thus balancing the trade-off between task success rate and interpretability.

3 Model Architecture and Interactive Training

We generally follow the architecture design in [4] for answer model and guesser. The only salient architecture difference is that we use seq2seq model for question generator instead of vanilla LSTMs.

seq2seq Question Generator The seq2seq model was originally introduced for machine translation, and later adopted in conversation systems. Compared with vanilla LSTM, seq2seq can be extended with attention modules for long distance reasoning. We used a global dot product attention layer to combine the visual context with the language embeddings.

The seq2seq model first encodes the previous conversation history using the LSTM encoder, then the language embeddings are mixed with the image feature (we also use VGG features) in the attention module. At reinforcement training stage, we only take at most 2 round of recent conversations as the input to the seq2seq model, instead of the whole conversation history.

Interactive Reinforcement Training After the supervised pre-training, three models only obtain knowledge from the static dataset, but not yet learn to cooperate with each other via interaction.

We argue that success communication requires *all* parties learn to adapt for others. Based on this intuition, we enable three models to actively learn in a self-talking environment in an interactive manner.

In [4], the reward function for the question generator is a binary score, dependent on whether the guesser finish the task: so the reward at round t is $R_t(\mathbf{s}_t, (q_t, a_t) : \theta_g) \in \{0, 1\}$, where θ_g is guesser's parameter, $\mathbf{s}_t$ is the state of question generator and answer model. We use the same reward definition to update the answer model. For the answer model, we also augment a score branch (a single FC + ReLU layer), to estimate the reward value, in order to stabilize the policy gradient updates.

With the above extension, we want to maximize the expected reward over two models's policies, parameterized by θ_a and θ_q: $J(\theta_a, \theta_q) = \mathop{\mathbb{E}}_{\pi_a, \pi_q}[\Sigma_{t=0}^{T} R_t(\mathbf{s}_t, (q_t, a_t))]$. The policy gradient updates for the question generator and the answer model can be written as follows:

$$\mathop{\nabla}_{\theta_q} J = \mathop{\mathbb{E}}_{\pi_a, \pi_A}[\sum_{t=1}^{T} \mathop{\nabla}_{\theta_q} log\pi_q(\mathbf{q_t}|\mathbf{s}_q^t) * (r - b_q))]; \qquad \mathop{\nabla}_{\theta_a} J = \mathop{\mathbb{E}}_{\pi_q, \pi_a}[\sum_{t=1}^{T} \mathop{\nabla}_{\theta_a} log\pi_a(\mathbf{a_t}|\mathbf{s}_a^t) * (r - b_a))] \quad (1)$$

Note that the evaluation of reward R depends on the guesser model. The accuracy of the guesser model is indeed far from perfect even for human dialogue (30% errors). The mismatch between generated conversation and human conversation further enlarges the error and affects the policy gradient. Therefore, we let the guesser tune itself on the generated dialogue. Guesser's parameter θ_g is updated by optimizing cross entropy loss of guesser's prediction using generated conversations:

$$\mathop{\nabla}_{\theta_g} L_{CE}(\mathbf{O}_{gt}, Guesser((q, a)_{:T}; \theta_g)) \quad (2)$$

Reward Engineering Above interactive learning effectively improves task success rate, but the generated conversations diverge from natural language towards an effective but uninterpretable communication protocols. One explanation is that during interactive training, the guesser model manages to tolerate the gradually shifted conversation, and feedback positive reward "over-generously".

Based on the above assumption, we use several heuristics to prune unnatural generated questions before feeding the generated conversation into the guesser. The intention is to limit the guesser only read the "natural" part of the conversation, and explicitly discourage the guesser to squeeze signal from unnatural QAs. Specifically, we use two heuristics to prune the unnatural QAs: 1) Removing questions containing repetitive words/phrases (e.g. "is it in front left front left front left?"); 2) Removing near duplicate questions happened in earlier conversations (e.g. "Is it on the left? ... On the left?"). In an extreme case, if the generated QAs are mostly unreadable and pruned, the guesser won't get enough input and forced to fail, so that the reward will be zero.

This way, we explicitly inject our preference for the generated text. As a result, we effectively interfere the rewarding function, thus it can be viewed as a form of reward engineering [5].

4 Experiments

Implementation details We use similar data preprocessing as [10]. We also do supervised pre-training and get comparable error rate as in [10] (answer model: 0.213, guesser: 0.380, question generator: 0.581). At interactive learning stage, we use Adam optimizer with batch size 64 with learning rate 1e-4. To generate questions, we use multinomial sampling in both training and testing.

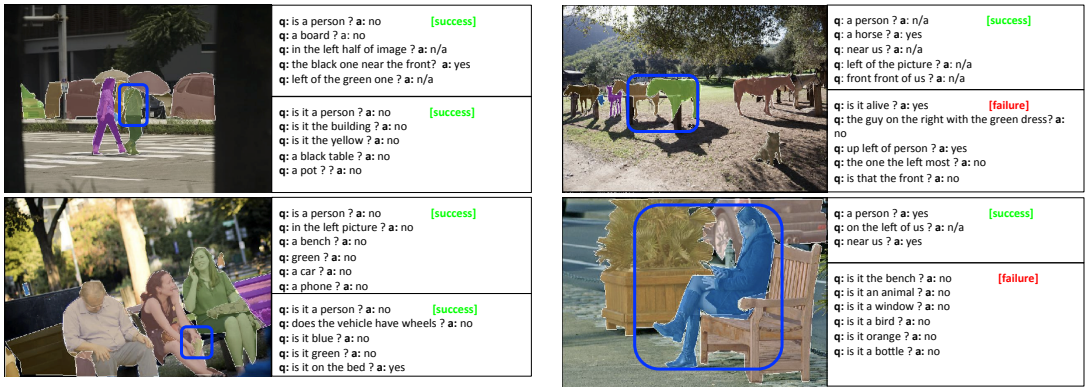
Task Success Rate: The task success rates for baseline models and different variants of interactive trained models are shown in Table 1a. In table 1a, we use IRL to denote interactive reinforcement learning and use the superscript to denote which models are actively tuned during training. All interactively trained models consistently outperform the baseline RL model [10]. [2] As expected, the most successful model is IRL^{QAG}, which is very close to the human score as measured in [4].

Quantifying Semantic Gap: Although effectively improve task success rate, IRL^{QAG} tends to generate uninterpretable conversations. Typical example are 1) repetitive word/phrase in questions; 2) the answer model tends to use "n/a" more frequently, even if the rational answers are "yes" or "no". (top left example in Figure 1b). We setup human studies to quantify this semantic gap: First, we asked independent human subjects to evaluate the generated *answer quality*. We randomly generated 100 cases and asked 20 human subjects to decide whether the generated answers agree with their judgments. Each QA pair is evaluated by 3 subjects with binary scores (3rd row in Table 1a). When guesser and answer model are jointly updated (IRL^{AG}), generated answers tend to disagree with humans. However, when the question generator is jointly updated, the answer quality improves, probably because generator adapts its questions to make it easy for answer model.

[2] The best score for baseline [10] is copied from the author's Github page

	SL	RL[10]	Human	RL^{Q}	IRL^{QA}	IRL^{QG}	IRL^{AG}	IRL^{QAG}	IRL-$prune^{QAG}$
SR	.417	.603	.844	.582	.651	.631	.777	**.829**	.813
Q-que	-	-	-	**1.71**	1.97	2.71	-	4.98	3.53
Q-ans	.780		-	**.915**	.801	.909	.408	.590	.681

(a) Evaluation: task success rate (SR) and quality of questions and answers on test set.



(b) Examples of generated conversation from IRL-$prune^{QAG}$ (top right in each cell) and SL baseline model (bottom right). The ground truth target region is highlighted by blue bounding boxes.

Second, we asked human subjects to evaluate *question quality* in terms of 1) whether the question is interpretable and 2) relevant to the image content. We asked the subjects to rank the quality of the generated questions from different models (best rank is 1). The averaged ranking is shown in the 2nd row of Table 1a. IRL^{QAG}'s semantic gap is enlarged most, despite improving success rate.

The reward engineering version IRL-$prune^{QAG}$ improves interpretability compared with IRL^{QAG}, but the task success rate is also slightly degraded. Some qualitative examples shown in Figure 1b.

5 Conclusion

We proposed an interactive training method on object instance level visual grounding conversation task and significantly improve task success rate. Observing the language drifting problem during the interactive learning, we proposed a reward engineering technique during training and improved interpretability. The major problem of our method is still language drifting. Our result also suggests visual goal-ended conversation need semantic evaluation metric other than task success rate.

References

[1] P Chattopadhyay, D Yadav, V Prabhu, A Chandrasekaran, A Das, S Lee, D Batra, and D Parikh. Evaluating visual conversational agents via cooperative human-ai games. *HCOMP*, 2017.

[2] A Das, S Kottur, K Gupta, A Singh, D Yadav, J Moura, D Parikh, and D Batra. Visual dialog. *CVPR*, 2017.

[3] A Das, S Kottur, J Moura, S Lee, and D Batra. Learning cooperative visual dialog agents with deep rl. *ICCV*, 2017.

[4] H de Vries, F Strub, S Chandar, O Pietquin, H Larochelle, and A Courville. Guesswhat?! visual object discovery through multi-modal dialogue. *CVPR*, 2017.

[5] Daniel Dewey. Reinforcement learning and the reward engineering principle. *2014 AAAI Spring Symposium Series*, 2014.

[6] K Evtimova, A Drozdov, D Kiela, and K Cho. Emergent language in a multi-modal, multi-step referential game. *arXiv:1705.10369*, 2017.

[7] S Kottur, J. Moura, S Lee, and D Batra. Natural language does not emerge naturally in multi-agent dialog. *EMNLP*, 2017.

[8] I Mordatch and P Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv:1703.04908*, 2017.

[9] I Mordatch S Milli, P Abbeel. Interpretable and pedagogical examples. *arXiv preprint arXiv:1711.00694*, 2017.

[10] F Strub, H de Vries, J Mary, B Piot, A Courville, and O Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. *IJCAI*, 2017.