

# Answerer in Questioner's Mind for Goal-Oriented Visual Dialogue

Sang-Woo Lee Yu-Jung Heo Byoung-Tak Zhang

## Abstract

Goal-oriented dialogue has been paid attention for its numerous applications in artificial intelligence. To solve this task, deep learning and reinforcement learning have recently been applied. However, these approaches struggle to find a competent recurrent neural questioner, owing to the complexity of learning a series of sentences. Motivated by theory of mind, we propose “Answerer in Questioner’s Mind” (AQM), a novel algorithm for goal-oriented dialogue. With AQM, a questioner asks and infers based on an approximated probabilistic model of the answerer. The questioner figures out the answerer’s intent via selecting a plausible question by explicitly calculating the information gain of the candidate intentions and possible answers to each question. We test our framework on two goal-oriented visual dialogue tasks: “MNIST Counting Dialog” and “Guess-What?!”. In our experiments, AQM outperforms comparative algorithms and makes human-like dialogue. We further use AQM as a tool for analyzing the mechanism of deep reinforcement learning approach and discuss the future direction of practical goal-oriented neural dialogue systems.

## 1. Introduction

Goal-oriented dialogue is a classical artificial intelligence problem including digital personal assistants, order-by-phone tools, and online customer service centers. Goal-oriented dialogue occurs when a questioner asks an action-oriented question and an answerer responds with the intent of letting the questioner know a correct action to take. Significant research on goal-oriented dialogue has tackled this problem (Lemon et al., 2006; Williams & Young, 2007), though a good solution has not yet been provided (Bordes & Weston, 2017).

Motivated by the achievement of neural chit-chat dialogue

School of Computer Science and Engineering, Seoul National University. Correspondence to: Byoung-Tak Zhang <btzhang@bi.snu.ac.kr>.

Copyright 2018 by the authors.

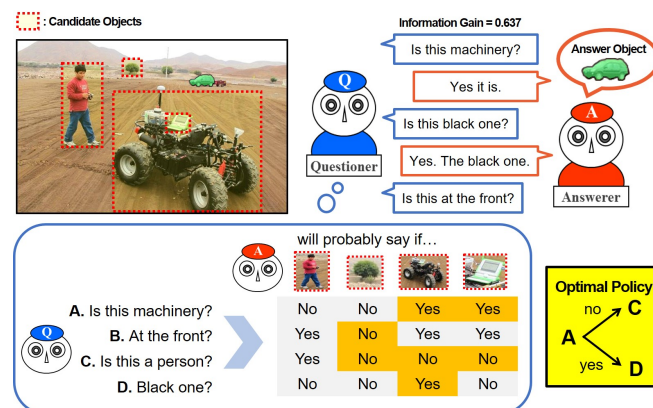


Figure 1. Illustration of an AQM algorithm for goal-oriented visual dialogue. AQM makes a decision tree on the image for asking efficient questions.

research (Vinyals & Le, 2015), recent studies on goal-oriented dialogues have utilized deep learning, using massive data to train their neural networks in for self-play environments. Many researchers attempted to solve goal-oriented dialogue tasks by using deep supervised learning (deep SL) approach (Wen et al., 2016) based on seq2seq models (Cho et al., 2014) or deep reinforcement learning (deep RL) approach utilizing rewards obtained from the result of the dialogue (Zhao & Eskenazi, 2016; Li et al., 2017). However, these methods struggle to find a correct RNN model that uses back-propagation, owing to the complexity of learning a series of sentences. These algorithms tend to generate redundant sentences, making generated dialogues inefficient (Kim et al., 2017; Das et al., 2017b).

Our idea to deal with goal-oriented dialogue is to introduce opponent modeling. In this approach, an agent considers what the opponent will respond by using an explicit approximated model of the opponent. Our study is motivated by theory of mind, the ability to attribute mental states to others and to understand how our mental states are different (Premack & Woodruff, 1978). If one wishes to efficiently convey information to an opponent, it is best to converse in a way that maximizes the opponent’s understanding (Bruner, 1981). For our method, we consider the mind to be beyond a part of mental states (e.g., belief, intent, knowledge). The

mind is the probabilistic distribution of the opponent model itself.

We propose an “Answerer in Questioner’s Mind” (AQM) algorithm to allow an agent to ask appropriate consecutive questions during goal-oriented dialogue (Figure 1). AQM’s questioner explicitly possesses an approximated model of the answerer. The questioner utilizes the approximated model to calculate the information gain of the candidate answerer’s intentions and answers for each question. In the experiment, AQM’s question generator extracts proper questions from training data, not really generates.

We test AQM mainly on goal-oriented visual dialogue tasks. There have been several kinds of visual-language tasks including image captioning (Vinyals et al., 2015) and visual question answering (VQA) (Antol et al., 2015; Mao et al., 2016), and recent research goes further to propose multi-turn visual dialogue tasks (Kim et al., 2017). Our main experiment is conducted on “GuessWhat?!” a cooperative two-player guessing game on the image (Figure 2). AQM achieves an accuracy of 63.63% in 3 turns and 78.72% in 10 turns, outperforming deep SL (46.8% in 4.1 turns) (de Vries et al., 2017) and deep RL (52.3% in 5 turns) (Strub et al., 2017) algorithms. Though we demonstrate the performance of our models in visual dialogue tasks, our approach can be directly applied to general goal-oriented dialogue where there is a non-visual context.

In the discussion section, aside from the experimental success of AQM, we leverage AQM as a tool for analyzing deep RL approach on goal-oriented dialogue tasks from the perspective of theory of mind. According to our argument, training two agents to make plausible dialogues via rewards during self-play is not adaptable to a service scenario. To enable an agent to converse with a human, the opponent agent in self-play should model a human as much as possible. We prove that AQM and RL have a similar objective function, and that the objective function of RL can be replaced by a direct objective function for each question, instead of reward assigned at the end of dialogue. Through these series of arguments, we discuss future directions for building practical goal-oriented dialogue systems.

## 2. Previous Works

### 2.1. Theory of Mind Approach

In this section, we introduce a series of studies considering opponent’s uncertainty or the opponent model itself explicitly. We refer to these methods as theory of mind approach.

**Opponent Modeling** Studies on opponent modeling have treated simple games with a multi-agent environment where an agent competed with the other (Hernandez-Leal & Kaisers, 2017; Foerster et al., 2017). In the study of Fo-

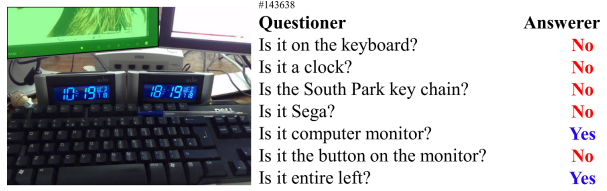


Figure 2. An example of the GuessWhat?! game. A green mask highlights the correct object.

erster et al., the agent has the model of the opponent and updates it assuming the opponent will be updated by gradient descent with RL. They argued modeling opponent could be applied to track the non-stationary behavior of an opponent agent. Their model outperformed classical RL methods in simple games, such as tic-tac-toe and rock-paper-scissors. On the other hand, AQM applied opponent modeling to a cooperative multi-agent setting. We believe that opponent modeling can also be applied to dialogue systems in which agents are partially cooperative and partially competitive.

**Obverter** A recent study applied the obverter technique (Batali, 1998), motivated by theory of mind, to an image-description-match classification task to study language emergence (Choi et al., 2018). In the experiments of Choi et al., one agent transmitted one sentence for describing artificial images to the opponent agent. In their study, the obverter technique can be understood as that an agent play both questioner and answerer, maximizing the consistency between visual and language modules. Their experimental results showed that their obverter technique generated a word corresponding to a specific object (e.g. ‘bbbbbbb{b,d}’ for a blue box). They argued their method could be an alternative to RL-based language emergence systems. Compared to their model, however, AQM uses real images, creates multi-turn dialogue, and can be used for general goal-oriented dialogue tasks.

**Information Gain** AQM’s question-generator optimizes information gain using an approximated opponent model. However, the concept of utilizing information gain in a dialogue task is not new for a classical rule-based approach. Polifroni and Walker used information gain to build a dialogue system for restaurant recommendations (Polifroni & Walker, 2006). They made a decision tree using information gain and asked a series of informative questions about restaurant preferences. Rothe et al. applied a similar method to generate questions on a simple Battleship game experiment (Rothe et al., 2017). It is noticeable that they used pre-defined logic to generate questions with information gain criteria to make novel (i.e., not shown in the training dataset) and human-like questions. Unlike these previous studies, AQM makes a new decision tree for every new context (see the decision tree in Figure 1). AQM also considers

uncertainty by using a multi-modal deep learning module, and does not require hand-made or domain-specific rules.

**Theory of AI's Mind** Chandrasekaran et al. argued that human-machine collaboration could be improved when humans understand the properties of behavior (e.g., strengths, weakness, or quirks) of the opponent machine (Chandrasekaran et al., 2017). They referred to this kind of human understanding as "theory of AI's mind." In their human experiment on visual question answering, a team consisting of a human-questioner and a machine-answerer performed better when the human knew whether the opponent could answer correctly.

## 2.2. Visually Grounded Dialogue Task

**GuessWhat?!** GuessWhat?! is a cooperative two-player guessing game proposed by De Vries et al. (Figure 2) (de Vries et al., 2017). GuessWhat?! has received attention in the field of deep learning and artificial intelligence as a testbed for research on the interplay of computer vision and dialogue systems. The goal is to locate a hidden object in a rich image scene by asking a sequence of questions. One participant, called "Oracle," or "Answerer" in our paper, is randomly assigned an object in the image. The other participant, "Questioner," guesses the object assigned to the answerer. The questioner asks a series of questions, for which the answerer responds as "yes," "no," and "n/a." When the questioner decides to guess the correct object, a list of candidate objects is then revealed. A win occurs when the questioner picks the correct object. The GuessWhat?! dataset contains 66,537 MSCOCO images (Lin et al., 2014), 155,280 games, and 831,889 question-answer pairs.

**Visual Dialog** In Visual Dialog (Das et al., 2017a), two agents also communicate with questioning and answering about the given MSCOCO image. Unlike GuessWhat?!, an answer can be a sentence and there is no restriction for the Visual Dialogue answerer. Das et al. used this dataset to make a goal-oriented dialogue task, where the questioner guesses a target image from 9,627 candidates in the test dataset (Das et al., 2017b). The dataset includes a true caption of each image achieving percentile ranks around 90%. In their self-play experiments, adding information via a dialogue improved the percentile ranks to around 93%, where the questioner and answerer were trained with deep SL and deep RL methods. This means that the models predict the correct image to be more exact than 93% of the rest images in the test dataset.

## 3. Answerer in Questioner's Mind (AQM)

**Preliminary** In our experimental setting, two machine players, a questioner and an answerer, communicate via natural dialogue. Specifically, there exists a class  $c$ , which is an

---

### Algorithm 1 AQM's Question-Generator

---

```

 $\hat{p}(c) \sim \hat{p}'(c)$ -model
 $\tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1}) \sim \tilde{p}(a|c, q)$ -model
 $Q \leftarrow Q$ -sampler
for  $t = 1:T$  do
     $q_t \leftarrow \operatorname{argmax}_{q_t \in Q} \tilde{I}[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}]$  in Eq. 2
    Get  $a_t$  from the answerer
    Update  $\hat{p}(c|a_{1:t}, q_{1:t}) \propto \tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1}) \cdot \hat{p}(c|a_{1:t-1}, q_{1:t-1})$ 
end for
    
```

---

answerer's intention or a goal-action, the questioner should perform. The answerer knows the class  $c$ , whereas the questioner does not. The goal of the dialogue for the questioner is to find the correct class  $c$  by asking a series of questions to the answerer. The answerer responds the correct answer to given question.

We treat  $C$ ,  $Q_t$ , and  $A_t$  as random variables of class,  $t$ -th question, and  $t$ -th answer, respectively.  $c$ ,  $q_t$ , and  $a_t$  becomes their single instance. In a restaurant scenario example,  $q_t$  can be "Would you like to order?" or "What can I do for you?"  $a_t$  can be "Two coffees, please." or "What's the password for Wi-Fi?"  $c$  can then be "Receive the order of two hot Americanos." or "Let the customer know the Wi-Fi password."

**AQM's claim** In our problem setting, the answerer requires one module, an answer-generator, and the questioner needs two modules, a question-generator and a guesser. The objective function of three modules is as follows.

Answer-generator:  $\operatorname{argmax}_{a_t} p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$

Guesser:  $\operatorname{argmax}_c p(c|a_{1:t}, q_{1:t})$

Question-generator:  $\operatorname{argmax}_{q_t} I[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}]$

$I$  is information gain or mutual information of the class  $C$  and the current answer  $A_t$ , where the previous history ( $a_{1:t-1}, q_{1:t-1}$ ) and a current question  $q_t$  are given (See Equation 2). Note that maximizing information gain can be understood as minimizing the conditional entropy of class  $C$ , given a current answer  $A_t$ .

**Answer-generator** For the answerer's answer-generator module, a neural network is trained by minimizing cross-entropy over the answer distribution  $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ , like the deep SL approach. On the other hand, the questioner's question-generator and guesser module possess the approximated answer distribution  $\tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ , or simply the likelihood  $\tilde{p}$ . The likelihood  $\tilde{p}$  can be obtained by learning training data as does in the answer-generator, or by distilling from an answer-generator module directly (Hinton et al., 2015). If

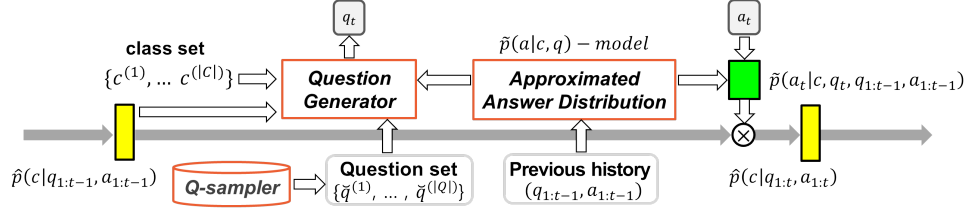


Figure 3. Illustration of the AQM question-generator module.

$a_t$  is a sentence, as in Visual Dialog (Das et al., 2017b), the probability is extracted from the multiplication of the word probability of RNN.

**Guesser** For the questioner’s guesser module, the posterior of class  $c$ ,  $\hat{p}(c)$ , is calculated based on the history  $(a_{1:t}, q_{1:t})$ , the likelihood  $p$ , and the prior of class  $c$ ,  $\hat{p}'(c)$ .

$$\hat{p}(c|a_{1:t}, q_{1:t}) \propto \hat{p}'(c) \prod_j^t \tilde{p}(a_j|c, q_j, a_{1:j-1}, q_{1:j-1}) \quad (1)$$

We use a term of likelihood as  $\tilde{p}$ , prior as  $\hat{p}'$ , and posterior as  $\hat{p}$  from the perspective that the questioner classifies class  $c$ . When the answerer’s answer distribution  $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$  is fixed, the questioner achieves an ideal performance when the likelihood  $\tilde{p}$  is the same as  $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ .

**Question-generator** AQM’s question-generator module selects  $q_t^*$ , which has a maximum information gain  $\tilde{I}[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}]$ , or simply  $\tilde{I}$ . To calculate information gain  $\tilde{I}$ , the question-generator module uses the likelihood  $\tilde{p}$  and the posterior  $\hat{p}$ .

$$\begin{aligned} q_t^* &= \operatorname{argmax}_{q_t \in Q} \tilde{I}[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}] \\ &= \operatorname{argmax}_{q_t \in Q} \sum_{a_t} \sum_c \hat{p}(c|a_{1:t-1}, q_{1:t-1}) \cdot \\ &\quad \tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1}) \ln \frac{\tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})}{\tilde{p}'(a_t|q_t, a_{1:t-1}, q_{1:t-1})} \end{aligned} \quad (2)$$

where  $\tilde{p}'(a_t|q_t, a_{1:t-1}, q_{1:t-1}) = \sum_c \hat{p}(c|a_{1:t-1}, q_{1:t-1}) \cdot \tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ .  $Q$  is the set of the candidate questions.

Algorithm 1 and Figure 3 explain the question-generator module procedure. The question-generator requires the  $\hat{p}'$ -model for the prior, the  $\tilde{p}(a|c, q)$ -model for the likelihood, and the  $Q$ -sampler for the set of candidate questions.

**Explainable AQM** Explainable machine learning has recently received much attention (Ribeiro et al., 2016; Zeiler

& Fergus, 2014). AQM is explainable because it is clear how a question is selected. Asking a series of questions in AQM corresponds to constructing an efficient decision tree classifier. We further introduce two properties of AQM related to explainability.

**AQM’s Property 1.** *The performance of AQM’s questioner is optimal, where the likelihood  $\tilde{p}$  is equivalent to the answering distribution of the opponent  $p$ .* For the guesser module, the performance of the guesser with the posterior  $\hat{p}$  is optimal when  $\tilde{p}$  is  $p$ . The performance of question-generator also increases as the  $\tilde{p}$  becomes more similar to the opponent, when the  $\hat{p}$  is fixed.

**AQM’s Property 2.** *The contexts of history questioner requires are the posterior  $\hat{p}$  and the history itself  $(a_{1:t}, q_{1:t})$  used as input for the likelihood  $\tilde{p}$ .* In comparative deep learning methods, hidden neurons in RNN are expected to track the context of history; though it has not been known yet what kinds of information should be tracked. If the question to be asked is independent from the previous questions, the only context AQM should track is the posterior  $\hat{p}$ . In this case, the posterior  $\hat{p}$  in the yellow box of Figure 3 corresponds to the hidden vector of the RNN in the comparative dialogue studies.

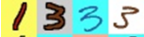

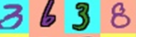
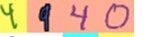
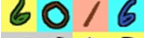

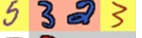

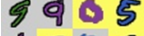

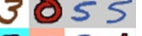
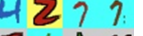
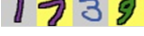



## 4. Experiments

### 4.1. MNIST Counting Dialog

To clearly explain the mechanism of AQM, we introduce an MNIST Counting Dialog task, which is a toy goal-oriented visual dialogue problem, illustrated in Figure 4. This task utilizes the concept of the MNIST Dialog dataset suggested by Seo et al. (Seo et al., 2017). Like MNIST Dialog, each image in MNIST Counting Dialog contains 16 digits, each having four randomly assigned properties: color = {red, blue, green, purple, brown}, bgcolor = {cyan, yellow, white, silver, salmon}, number = {0, 1, ..., 9}, and style = {flat, stroke}. Unlike MNIST Dialog, MNIST Counting Dialog only asks counting questions. This type does not require the information of previous questions and answers.

The goal of the MNIST Counting Dialog task is to inform the questioner to pick the correct image among 10K candidate images via questioning and answering. In other words,



#	Question	Answer
1	How many 0's are there in image?	One
2	How many green digits are there in the image?	Three
3	How many digits with yellow background are there in the image?	Six
4	How many stroke digits are there in the image?	Eleven

Figure 4. Illustration of MNIST Counting Dialog, a simplified version of MNIST Dialog (Seo et al., 2017).

class  $c$  is an index of the true target image ( $1 \sim 10,000$ ). In the example of Figure 4, asking about the number of 1 digits or 6 digits classifies a target image perfectly if the recognition accuracy on each digit in the image is 100%. Asking about the number of 0 digits does not help classify, because all images have one zero. If recognition accuracy is less than 100%, asking about the number of 1 digits is better than asking about the number of 6 digits, because the variance of the number of 1 digits is larger than that of the 6 digits.

For the MNIST Counting Dialog task, we do not use neural networks for modeling the questioner or answerer. The questioner's answering model is count-based for each property value and is trained for 30K training data. To add randomness or uncertainty to this task, we set the ratio of recognition accuracy of each property to that of the answerer. If the recognition accuracy on the digit decreases, the answering accuracy on the image is decreased much. If the recognition accuracy of color is 85%, the answering accuracy is 46.6%, on average. 22 questions about red to stroke are used for the candidate questions.

Figure 5 shows that AQM nearly always chooses the true target image from 10K candidates in six turns if the recognition accuracy is 100%. However, AQM also chooses correctly with a probability of 51%, 39%, and 31% (test accuracy) in six turns, when the recognition accuracy (Acc in the legend) is 95%, 90%, 85%, respectively. "Random" denotes a questioner with a random question-generator module and the AQM's guesser module.

## 4.2. GuessWhat?!

**$\hat{p}'(c)$ -model for the Prior** The questioner does not know the list of candidate objects while asking questions. This makes the GuessWhat?! task difficult, although the number of candidates is around 8. We use YOLO9000, a real-time object detection algorithm, to estimate the set of candidate objects (Redmon & Farhadi, 2016). The prior  $\hat{p}'(c)$  is set to

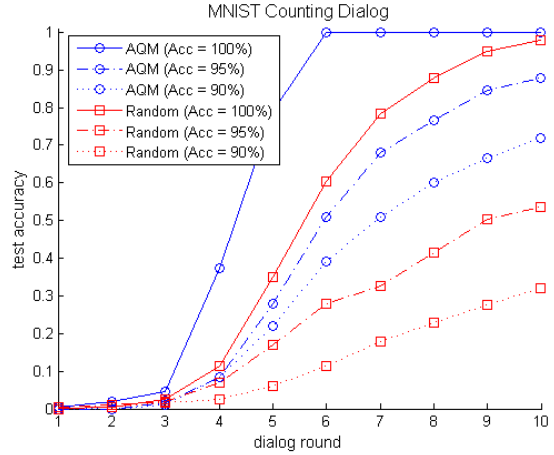


Figure 5. Test accuracy from the MNIST Counting Dialog task. Acc is the average of the randomly assigned recognition accuracy.

$1/N$ , where  $N$  is the number of extracted objects.

**$\tilde{p}(a|q, c)$ -model for the Likelihood** We use the answerer model from previous GuessWhat?! research (de Vries et al., 2017). The inputs of the answer-generator module consist of a VGG16 feature of a given context image, a VGG16 feature of the cropped object in the context image, spatial and categorical information of the target object, and the question  $q_t$  at time step  $t$ . Our answer-generator module assumes the answer distribution is independent from the history ( $a_{1:t-1}, q_{1:t-1}$ ).

$$\tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1}) \propto \tilde{p}''(a_t|c, q_t) \quad (3)$$

We use two strategy to make the questioner's answer distribution  $\tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$  approximate the answerer's answer distribution  $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ . The first is "indA," in which  $p$  and  $\tilde{p}$  is trained separately for the training data. The second is "depA," in which  $\tilde{p}$  is trained for the answer from  $p$ . The question and the image is also sampled from the training data.

**$Q$ -sampler for the Candidate Question Set** We compare two  $Q$ -samplers. The first is "randQ," which samples questions randomly from the training data. The second is "countQ," which causes every other question from the set  $Q$  to be less dependent on the other. countQ checks the dependency of two questions with the following rule: the probability that two consecutive questions' answers are same cannot exceed 95%. In other words,  $\sum_a \tilde{p}^\dagger(a_i = a|q_i, a_j = a, q_j) < 0.95$ , where  $\tilde{p}^\dagger(a_i|q_i, a_j, q_j)$  is derived from the count of a pair of answers for two questions in the training data.  $\tilde{p}$  made by indA is used for countQ. We set the size of  $Q$  to 200.

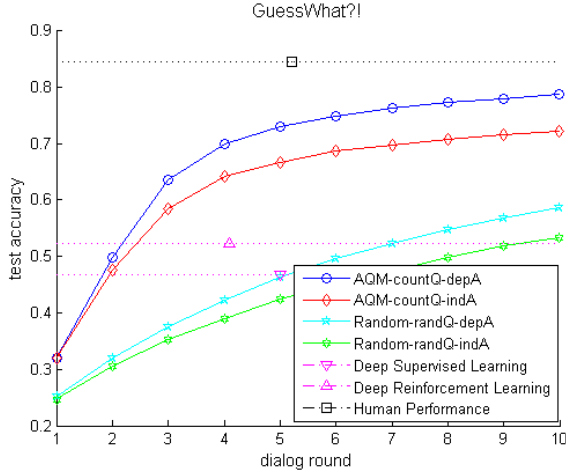


Figure 6. Test accuracy from the GuessWhat?! task. Previous works do not report the performance change with an increase in the number of turns.

**Experimental Results** Figure 6 and Table 1 shows the experimental results. Figure 7 illustrates the generated dialogue. Our best algorithm, AQM-countQ-depA, achieved 63.63% in three turns, outperforming deep SL and deep RL algorithms. By allowing ten questions, the algorithms achieved 78.72% and reached near-human performance. If answerer’s answer distribution  $p$  is directly used for the questioner’s answer distribution  $\tilde{p}$  (i.e.,  $\tilde{p} = p$ ), AQM-countQ achieved 63.76% in three turns and 81.96% in ten turns. depA remarkably improved the score in self-play but did not increase the quality of the generated dialogue significantly. On the other hand, countQ did not boost the score much but increased the quality of the generated dialogue. The comparative deep SL method used the question-generator with hierarchical recurrent encoder-decoder (Serban et al., 2015), achieving an accuracy of 46.8% in five turns (de Vries et al., 2017). However, Random-randQ-depA achieved 46.36% in five turns, which is a competitive result to the deep SL model. The comparative deep RL method applied reinforcement learning on long short-term memory, achieving 52.3% in about 4.1 turns (Strub et al., 2017).

## 5. Discussion

Many researchers have studied dialogue systems for cooperative games using RL, to increase the score in self-play environments (Lazaridou et al., 2016; Kottur et al., 2017; Mordatch & Abbeel, 2017). In the discussion section, we discuss various issues in RL research on goal-oriented dialogue, mainly based on the case of goal-oriented visual dialogue studies (de Vries et al., 2017; Das et al., 2017b). Our primary argument is that considering the opponent’s

Table 1. Test accuracies from the GuessWhat?! task.

Model	Accuracy
Baseline	16.04
Deep SL (de Vries et al., 2017)	46.8
Deep RL (Strub et al., 2017)	52.3
<b>Random-randQ-indA (5-q)</b>	42.48 ( $\pm 0.84$ )
<b>Random-randQ-depA (5-q)</b>	46.36 ( $\pm 0.91$ )
<b>AQM-randQ-indA (5-q)</b>	65.66 ( $\pm 0.55$ )
<b>AQM-countQ-indA (5-q)</b>	66.73 ( $\pm 0.76$ )
<b>AQM-countQ-depA (5-q)</b>	72.89 ( $\pm 0.70$ )
<b>AQM-countQ-depA (10-q)</b>	<b>78.72 (<math>\pm 0.54</math>)</b>
Human (Strub et al., 2017)	84.4

mind in implementing an agent is important.

Section 5.1 discusses three objectives of goal-oriented visual dialogue research: score in self-play, service, and language emergence. Section 5.2 introduces three properties of RL algorithms in goal-oriented dialogue tasks using AQM as a tool. Section 5.3 discusses the direction of future works to make the questioner agent applicable for service.

### 5.1. Objective Function of Goal-Oriented Visual Dialogue

**Score** In goal-oriented visual dialogue research, score is used as one of the main measurements of dialogue efficiency. However, a high score can be achieved via optimization over  $p(c|a_{1:T}, q_{1:T})$ , which is the objective function of RL. Particularly, the agent can achieve a high score if the probability distribution of both the questioner and answerer is not bound to a human’s distribution, even when human-like dialogue is generated. For example, in GuessWhat?!, Han et al. showed that pre-defined questions about location can provide an accuracy of 94.34% in five turns (Han et al., 2017). Their methods divided an image evenly into three parts using two vertical or horizontal lines for three cases of answer {yes, no, n/a}. A natural language-based protocol can also be created by using size, color, category, or other major properties of the object.

**Service** One of the ultimate objectives of goal-oriented dialogue research is to create an agent that can be used in a real service (Bordes & Weston, 2017). However, successful reports have been limited. Chattopadhyay et al. reported the human-machine performance of the deep RL method in a study on Visual Dialog (Das et al., 2017b; Chattopadhyay et al., 2017). In their method, questioner and answerer were both fine-tuned by RL. Thus, the answerer’s answering distribution differed from the training data. This RL method also used the objective function of deep SL as the regularizer, conserving generated dialogue as human-like.





	Groundtruth	Supervised Learning (SL)	Answerer in Questioner's Mind (AQM)
	Is it person? Is he in the foreground? Is he wearing blue?	Is it a person? Is it the one in the front? Is it the one in the middle? Is it the one in the middle? Is it the whole person?	Is it a person? Is it a player? Is it in the background? Is it the umpire? Single piece in the bunch?
	Is it an animal? Is it a device? Is it silver in color?	Is it the cat? Is it the cat? Is it the chair? Is it the book? Is it the book?	Is it alive? A laptop? Is it the shelf portion on the right? Left side of green color? Is a table?
	Groundtruth	Reinforcement Learning (RL)	Answerer in Questioner's Mind (AQM)
	Is it a person?	Is it a person? Is it girl in white?	Is it a fork? Is it a chair? Is it a food? Is it a person? Is it alive?
	Is a person? Is it tree? Is an object? The bat playing? The shoes on food? Is the red seat?	Is it a person? Is it a love? Is an umbrella? Is in the middle? On a person? Is it on on far right?	Is it a person? Is it the chair? Is it a traffic signal light? Is it the phone? Is it alive?

Figure 7. Generated dialogues from our algorithm and the comparative algorithms. The tested games are sampled from the selected results of previous papers.

Nevertheless, the RL algorithm deteriorated the score in the game with a human, compared to deep SL. The authors also assessed six measures of generated dialogue quality: accuracy, consistency, image understanding, detail, question understanding, and fluency. However, the human subjects reported that the deep RL algorithm performed worse than the deep SL algorithm for all measures, except “detail.”

**Language Emergence** Plenty of research has recently been published on language emergence with RL in a multi-agent environment. Some studied artificial (i.e., non-natural) language (Evtimova et al., 2017; Lazaridou et al., 2018), whereas others attempted to improve the quality of generated natural dialogue. One of the best progress found in the latter study is the improvement on the quality of a series of questions in a multi-turn VQA. When deep RL is applied, the questioner generates fewer redundant questions than deep SL (Strub et al., 2017; Das et al., 2017b). It can be understood that the questioner in these methods are optimized by both  $p(q_t|a_{1:t-1}, q_{1:t-1})$  and  $p(c|a_{1:T}, q_{1:T})$ . The questioners are improved more than the answerers, because the answerer gets the objective function directly for each answer (i.e., VQA), whereas the questioner does not.

## 5.2. Analyzing RL via AQM

RL’s Property 1. *AQM and RL approaches share same objective function.* Two algorithms have the same objective function with the assumption that  $q_t$  only considers the performance of the current turn. The assumption is

used in the second line of the following equation.

$$\begin{aligned}
 & \arg\max_{q_t} \max_{q_{t-}} \ln p(c|a_{1:T}, q_{1:T}) \\
 & \approx \arg\max_{q_t} \ln p(c|a_{1:t}, q_{1:t}) \\
 & = \arg\max_{q_t} E \left[ \ln \frac{p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})}{p(a_t|q_t, a_{1:t-1}, q_{1:t-1})} \right] \\
 & = \arg\max_{q_t} I[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}]
 \end{aligned} \tag{4}$$

In the third line,  $a_t \sim p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ ,  $c \sim p(c|a_{1:t-1}, q_{1:t-1})$ , and Bayes rule is used. The assumption in the second line can be alleviated via multi-step AQM, which uses  $I[C, A_{t:t+k}; q_{t:t+k}, a_{1:t-1}, q_{1:t-1}]$  as the objective function of the question-generator module. In the multi-step AQM, the optimal question cannot be selected in a greedy way, unlike AQM. The multi-step AQM needs to search in a tree structure; a Monte Carlo tree search (Coulom, 2006) can be used to find a reasonable solution.

The RL and AQM question-generator are closely related, as is the discriminative-generative pair of classifiers (Ng & Jordan, 2002). AQM’s question-generator and guesser module explicitly have a likelihood  $\hat{p}$ , whereas the RL’s modules do not have explicitly. The properties of AQM, for example, including optimal conditions and sentence dynamics (AQM’s property 1 and 2), can be extended to RL. The complexity of RL’s question-generator can be decomposed to tracking class posteriors  $p(c|a_{1:t-1}, q_{1:t-1})$  and

history  $(a_{1:t-1}, q_{1:t-1})$  for multi-turn question answering. For human-like learning, the context for language generation  $p(q_t|a_{1:t-1}, q_{1:t-1})$  is also required; Q-sampler corresponds to this context.

*RL’s Property 2. Optimizing both questioner and answerer with rewards makes the agent’s performance with human worse.* This is true, even when the process improves a score during self-play or uses tricks to maintain a human-like language generation. The property of self-play in a cooperative goal-oriented dialogue task is different from the case of AlphaGo, which defeated a human Go champion (Silver et al., 2017). For example, in GuessWhat?!, the reversed response of the answerer (e.g., “no” for “yes”) may preserve the score in the self-play, but it makes the score in the human-machine game near 0%.

According to AQM’s Property 1, for the play of a machine questioner and a human answerer, the performance is optimal only if the approximated answerer’s distribution  $\tilde{p}$  of AQM’s questioner is the same as the human’s answering distribution. According to RL’s Property 1, RL and AQM shares the optimality condition about the distribution of the opponent. Fine-tuning an answerer agent with a reward makes the distribution of the agent different from a human’s. Therefore, fine-tuning both agents decreases performance in service situations. The experiment with a human, studied by Chattopadhyay et al. explained in Section 5.1, empirically demonstrates our claim.

*RL’s Property 3. An alternative objective function exists, which is directly applicable to each question.* A reward can only be applied when one game is finished. If the goal of training is to make language emergence itself or to make an agent for service, two agents can communicate with more than just question-answering for back-propagation, including sharing attention for the image (Kim et al., 2017). Cross-entropy for the guesser of each round can be considered to replace reward. Information gain can also be used not only for AQM but also for alternative objectives of back-propagation.

### 5.3. Future Works

**RL with Theory of Mind** RL methods can be enhanced in a service scenario by considering the answering distribution of human. It is advantageous for the machine questioner to ask questions for which the human answer is predictable (Chandrasekaran et al., 2017). In other words, a question having a high VQA accuracy is preferred. The model uncertainty of the questioner can also be measured and utilized with recent studies on Bayesian neural networks and uncertainty measures (Kendall & Gal, 2017). Because the questioner has the initiative of dialogue, the questioner does not need to necessarily learn the entire distribution of human conservation. The question, which the questioner uses

frequently in self-play, can be asked more to a human. Then, the obtained question-answer pairs can be used for improving the answerer, like in active learning.

**Opponent Modeling in the Test-Phase** In the perspective of making an agent for service, RL in self-play can be understood as opponent modeling in the training-phase; An opponent model in self-play is just an approximated model of the true opponent, a human. On the other hand, AQM can be understood as opponent modeling in the test-phase, especially when the approximated model of the agent is the opponent model itself in self-play.

Opponent modeling during the test-phase can make it easier to handle a non-stationary environment (Foerster et al., 2017) or a multi-domain problem. The AQM models can also be easily trained because research on training an answerer model on VQA tasks has been more progressed than training an RNN for the questioner. However, during test-phase opponent modeling, it is difficult to back-propagate in an end-to-end way. Thus, it requires additional techniques. For example, in GuessWhat, if the Q-sampler in our experiment is replaced with seq2seq trained by a deep SL method, AQM would generate a question by optimizing both sentence probability from the seq2seq model and information gain, because the Q-sampler in AQM corresponds to regularizing with  $p(q_t|a_{1:t-1}, q_{1:t-1})$  in the deep RL approach.

## 6. Concluding Remarks

Our contributions are two-fold. First, we proposed the AQM, a practical and explainable goal-oriented dialogue system. The algorithm is a scaled-up version of theory of mind methods for a realistic problem. During the self-play experiment, AQM outperformed comparative deep SL and RL methods. Second, we used AQM as a tool for analyzing deep RL research. We argued that the AQM and RL approach shared the same objective function. From our discussion: 1) optimizing both agents through reward during self-play makes the service performance worse; and 2) the reward in RL can be replaced with an alternative objective function directly applicable to each question.

The limitation of AQM is that the question-generator retrieves a question from the training data, not generating a new question. One of the future directions would include generating questions from the perspective discussed in this paper. However, just generating novel questions can be produced with a simple rule-based program (Rothe et al., 2017) or replacing Q-sampler to a seq2seq model. Our primary interest is to make an agent that can talk with a human for a practical service. In our future work, we will make a new algorithm that has strengths of both theory of mind and deep RL approach.



## Acknowledgements

The authors would like to thank Jin-Hwa Kim, Cheolho Han, Wooyoung Kang, Jaehyun Jun, Christina Baek, Hanock Kwak, Marco Baroni, and Jung-Woo Ha for helpful comments and editing.

## References

- Antol, Stanislaw, Agrawal, Aishwarya, Lu, Jiasen, Mitchell, Margaret, Batra, Dhruv, Lawrence Zitnick, C, and Parikh, Devi. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015.
- Batali, John. Computational simulations of the emergence of grammar. *Approaches to the evolution of language: Social and cognitive bases*, 405:426, 1998.
- Bordes, Antoine and Weston, Jason. Learning end-to-end goal-oriented dialog. In *ICLR*, 2017.
- Bruner, Jerome S. Intention in the structure of action and interaction. *Advances in infancy research*, 1981.
- Chandrasekaran, Arjun, Yadav, Deshraj, Chattopadhyay, Prithvijit, Prabhu, Viraj, and Parikh, Devi. It takes two to tango: Towards theory of ai’s mind. *arXiv preprint arXiv:1704.00717*, 2017.
- Chattopadhyay, Prithvijit, Yadav, Deshraj, Prabhu, Viraj, Chandrasekaran, Arjun, Das, Abhishek, Lee, Stefan, Batra, Dhruv, and Parikh, Devi. Evaluating visual conversational agents via cooperative human-ai games. *arXiv preprint arXiv:1708.05122*, 2017.
- Cho, Kyunghyun, van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.
- Choi, Edward, Lazaridou, Angeliki, and Freitas, Nando de. multi-agent compositional communication learning from raw visual input. In *ICLR*, 2018.
- Coulom, Rémi. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pp. 72–83. Springer, 2006.
- Das, Abhishek, Kottur, Satwik, Gupta, Khushi, Singh, Avi, Yadav, Deshraj, Moura, José MF, Parikh, Devi, and Batra, Dhruv. Visual dialog. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.
- Das, Abhishek, Kottur, Satwik, Moura, José MF, Lee, Stefan, and Batra, Dhruv. Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585*, 2017b.
- de Vries, Harm, Strub, Florian, Chandar, Sarath, Pietquin, Olivier, Larochelle, Hugo, and Courville, Aaron. Guess-what?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- Evtimova, Katrina, Drozdov, Andrew, Kiela, Douwe, and Cho, Kyunghyun. Emergent language in a multi-modal, multi-step referential game. *arXiv preprint arXiv:1705.10369*, 2017.
- Foerster, Jakob N, Chen, Richard Y, Al-Shedivat, Maruan, Whiteson, Shimon, Abbeel, Pieter, and Mordatch, Igor. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.
- Han, Cheolho, Lee, Sang-Woo, Heo, Yujung, Kang, Wooyoung, Jun, Jaehyun, and Zhang, Byoung-Tak. Criteria for human-compatible ai in two-player vision-language tasks. In *2017 IJCAI Workshop on Linguistic and Cognitive Approaches to Dialogue Agents*, 2017.
- Hernandez-Leal, Pablo and Kaisers, Michael. Learning against sequential opponents in repeated stochastic games. In *The 3rd Multi-disciplinary Conference on Reinforcement Learning and Decision Making*, Ann Arbor, 2017.
- Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeff. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Kendall, Alex and Gal, Yarin. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in neural information processing systems*, 2017.
- Kim, Jin-Hwa, Parikh, Devi, Batra, Dhruv, Zhang, Byoung-Tak, and Tian, Yuandong. Codraw: Visual dialog for collaborative drawing. *arXiv preprint arXiv:1712.05558*, 2017.
- Kottur, Satwik, Moura, José, Lee, Stefan, and Batra, Dhruv. Natural language does not emerge ‘naturally’ in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2962–2967, 2017.
- Lazaridou, Angeliki, Peysakhovich, Alexander, and Baroni, Marco. Multi-agent cooperation and the emergence of (natural) language. In *ICLR*, 2016.

- Lazaridou, Angeliki, Hermann, Karl Moritz Hermann, Tuyls, Karl, and Clark, Stephen. Emergence of linguistic communication from referential games with symbolic and pixel input. In *ICLR*, 2018.
- Lemon, Oliver, Georgila, Kallirroi, Henderson, James, and Stuttle, Matthew. An isu dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the talk in-car system. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pp. 119–122. Association for Computational Linguistics, 2006.
- Li, Xuijun, Chen, Yun-Nung, Li, Lihong, and Gao, Jianfeng. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*, 2017.
- Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.
- Mao, Junhua, Huang, Jonathan, Toshev, Alexander, Camburu, Oana, Yuille, Alan L, and Murphy, Kevin. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.
- Mordatch, Igor and Abbeel, Pieter. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.
- Ng, Andrew Y and Jordan, Michael I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pp. 841–848, 2002.
- Polifroni, Joseph and Walker, Marilyn. Learning database content for spoken dialogue system design. In *5th International Conference on Language Resources and Evaluation (LREC)*, 2006.
- Premack, David and Woodruff, Guy. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1 (4):515–526, 1978.
- Redmon, Joseph and Farhadi, Ali. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
- Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.
- Rothe, Anselm, Lake, Brenden M, and Gureckis, Todd. Question asking as program generation. In *Advances in Neural Information Processing Systems*, pp. 1046–1055, 2017.
- Seo, Paul Hongsuck, Lehrmann, Andreas, Han, Bohyung, and Sigal, Leonid. Visual reference resolution using attention memory for visual dialog. In *Advances in neural information processing systems*, 2017.
- Serban, Iulian V, Sordoni, Alessandro, Bengio, Yoshua, Courville, Aaron, and Pineau, Joelle. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 2015.
- Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.
- Strub, Florian, de Vries, Harm, Mary, Jeremie, Piot, Bilal, Courville, Aaron, and Pietquin, Olivier. End-to-end optimization of goal-driven and visually grounded dialogue systems. *arXiv preprint arXiv:1703.05423*, 2017.
- Vinyals, Oriol and Le, Quoc. A neural conversational model. In *ICML deep learning workshop*, 2015.
- Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.
- Wen, Tsung-Hsien, Vandyke, David, Mrksic, Nikola, Gasic, Milica, Rojas-Barahona, Lina M, Su, Pei-Hao, Ultes, Stefan, and Young, Steve. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.
- Williams, Jason D and Young, Steve. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.
- Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.
- Zhao, Tiancheng and Eskenazi, Maxine. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*, 2016.