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

Sang-Woo Lee, Yujung Heo, and Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University
Seoul 08826, Republic of Korea
{slee, yjheo, btzhang}@bi.snu.ac.kr

Abstract

We propose an "answerer in questioner's mind" (AQM) framework, a novel approach for goal-oriented dialogue. In AQM, a questioner asks and infers based on an approximated probabilistic model of the answerer in the questioner's model. The questioner figures out the answerer's intent by selecting a plausible question following the principle of information theory, which explicitly calculates the information gain of candidate intentions and candidate answers to each question. This process allows the proposed method to overcome learning difficulties, owing to comparative neural conversation systems and a deep reinforcement learning approach. We test our framework on two goal-oriented visual dialog tasks, "MNIST Counting Dialog" and "GuessWhat?!."

1 Introduction

Recent studies utilize deep learning with massive data to solve goal-oriented visual dialogue tasks. In these studies of deep learning, two neural agents are trained concurrently to make a successful dialogue. Many researchers have attempted deep supervised learning (deep SL) approach using seq2seq, which produces a corresponding sentence when one sentence is put into a recurrent neural network [1–4]. Other researchers fine-tunes a pre-trained network using the deep reinforcement learning (deep RL) approach [4–8]. However, these algorithms struggle to find a correct recurrent neural network model using back-propagation, because of the complexity in learning a series of sentences [9].

We propose the "answerer in questioner's mind'' (AQM) framework to allow an agent to ask appropriate consecutive questions during goal-oriented dialogue. We argue that it is advantageous for a questioner to ask a question that generates a different answer per the answerer's intent. In AQM, the questioner explicitly possesses an approximated probabilistic model of the answerer. The questioner utilizes the answerer's approximated model to explicitly calculate the information gain of the candidate answerer's intentions and the answers for each question. In the experiments, we select the set of candidate questions from the training data. After that, the questioner selects the question containing maximum information gain during each turn of dialogue.

2 Answerer in Questioner's Mind (AQM) Framework

Preliminary In our experimental setting, two machine players, a questioner, and an answerer communicate via natural dialog. Specifically, there exists a class c: the answerer's intention or the goal-action the questioner should perform. The answerer knows the class c whereas the questioner does not. The goal of the dialog is to let questioner know the correct class c via asking a question by a questioner and receiving an answer by an answerer. We treat C, Q_t, and A_t as random variable of class, t-th question, and t-th answer, respectively. c, q_t, and a_t becomes their single instance. In a

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

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 needs one module, the answer-generator, and the questioner needs two modules, a question-generator and a guesser. The objective function of the answer-generator and guesser is maximizing $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ and $p(c|a_{1:t}, q_{1:t})$, respectively. We set the objective function of question-generator as maximizing the following equation of information gain $I[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}]$.

$$
\begin{aligned}
&I[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}] \\
&= \sum_{a_t} \sum_{c} p(c|a_{1:t-1}, q_{1:t-1}) \cdot p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1}) \ln \frac{p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})}{p(a_t|q_t, a_{1:t-1}, q_{1:t-1})}
\end{aligned} \tag{1}
$$

For the answerer's answer-generator module, a neural network is trained by minimizing cross-entropy over the answer distribution, $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$, which is the same as the deep SL approach. Alternatively, 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 from training data, as does the answer-generator, or by distilled from an answer-generator module directly [10]. If a_t is sentence, like Visual Dialog [11, 4], the probability can be extracted from the multiplication of the word probability of the recurrent neural network.

Guesser For the questioner's guesser module, the posterior of class c given a history $(a_{1:t}, q_{1:t})$ and the prior of class c is calculated.

$$
\hat{p}(c|a_{1:t}, q_{1:t}) \propto \hat{p}'(c) \prod_{j}^{t} \tilde{p}(a_j|c, q_j, a_{1:j-1}, q_{1:j-1}) \tag{2}
$$

We refer to this prior of class c as $\hat{p}'(c)$ or simply the prior $\hat{p}'$, and this posterior of class c as $\hat{p}(c|a_{1:t}, q_{1:t})$ or simply the posterior $\hat{p}$. We use a term of likelihood as $\tilde{p}$, prior as $\hat{p}'$, and posterior as $\hat{p}$ in the perspective that a questioner classifiers 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 same as $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$.

Question-generator For the AQM question-generator module, the following question q_t^* is selected, having a maximum information gain of $\tilde{I}[C, A_t; q_t, a_{1:t-1}, q_{1:t-1}]$, or simply $\tilde{I}$. To calculate the information gain $\tilde{I}$, the question-generator module uses the likelihood $\tilde{p}$ and the posterior $\hat{p}$.

3 Experiments on MNIST Counting Dialog

To clearly explain the mechanism of AQM, we introduce an MNIST Counting Dialog task, a toy goal-oriented visual dialogue problem, as illustrated in Figure 1 (Left). This task utilize the concept of the MNIST Dialog dataset suggested by Seo et al. [12]. Like MNIST Dialog, each image in MNIST Counting Dialog contains 16 digits, each of which digits has four randomly assigned properties: color = {red, blue, green, purple, brown}, bgcolor = {cyan, yellow, white, silver, salmon}, number = {0, 1, $\cdots$, 9}, and style = {flat,stroke}. Unlike MNIST Dialog, MNIST Counting Dialog only ask counting questions. This type of questions does not require the information of previous questions and answers.

The goal of the MNIST Counting Dialog task is informing the question to pick the correct image from 10K candidate images by questioning and answering. For this task, we do not use neural networks or machine learning algorithms for modeling questioner or answerer. To put randomness or uncertainty to this task, we set the ratio of the recognition accuracy of each property to the answerer. The recognition error of the digit is amplified to the counting error of the image. If the recognition accuracy of counting color is 85%, the counting accuracy is 46.6% in average. Answering model

#	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

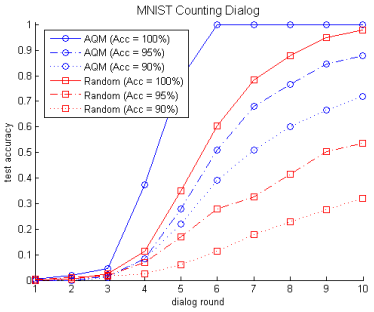

Figure 1: (Left) An illustration of MNIST Counting Dialog. It is simplified version of MNIST Dialog [12]. (Right) The test accuracy from the MNIST Counting Dialog Task. Acc is the average of the randomly assigned recognition accuracy. AQM rapidly picks the correct image from 10K candidates, even though there are uncertainty in recognition on the property of digits.

in questioner is count-based for each value of property, and is trained for 30K training data. 22 questions (for from red to stroke) and corresponding 22 answerer's answers are used for learning each instance of the training data.

Figure 1 (Right) shows that AQM almost always picks out the true target image from 10K candidates in six turns if the recognition accuracy is 100%. However, AQM also picks out correctly with the probability of 51%, 39%, 31% in six turns, if the recognition accuracy is 95%, 90%, 85%, respectively. This property of AQM can be extended to general goal-oriented visual dialogues. 'Random' denotes the questioner with random question-generator module and AQM's guesser module.

4 Experiments on GuessWhat?!

We test the proposed AQM framework using GuessWhat?!, a goal-oriented visual dialogue task proposed by de Vries et al. [3], as illustrated in Figure 2 (Left). In GuessWhat?!, two machine players (i.e. questioner and answerer) ask and answer 'yes,' 'no,' and 'N/A' questions about two objects hidden in an image, leading the questioner to select the correct object. The image containing one correct object, known by the answerer but not known by the questioner. The questioner asks a series of questions to gain enough evidence to locate the correct object. When the questioner decides to guess the correct object, the list of candidate objects is revealed. A win occurs when the questioner picks the correct object.

$\hat{p}'(c)$-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 10. We use YOLO2, a state-of-the-art object detection algorithm, to estimate the set of candidate objects [13]. The prior $\hat{p}'(c)$ is set to $1/N$, where N is the number of extracted objects.

$\tilde{p}(a|q,c)$-model for the Likelihood We use a visual question answering model used in the previous research of GuessWhat?! [3], for the both the answerer's answer distribution $p(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$ and the questioner's answer distribution (or likelihood) $\tilde{p}(a_t|c, q_t, a_{1:t-1}, q_{1:t-1})$. In our experiment, we directly utilize answerer's p as questioner's $\tilde{p}$. This setting assumes the ideal training of $\tilde{p}$ is achieved though there still exists an uncertainty from the probability distribution itself. The inputs of the answer-generator module we used 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 a_t at time step t. In our experiment, the answerer answers deterministically by using the argmax of the softmax distribution from the answer-generator module, whereas the questioner's answer-generator considers the uncertainty of the softmax distribution. Our answer-generator module assumes the answer distribution is independent to its previous history $(a_{1:t-1}, q_{1:t-1})$.

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

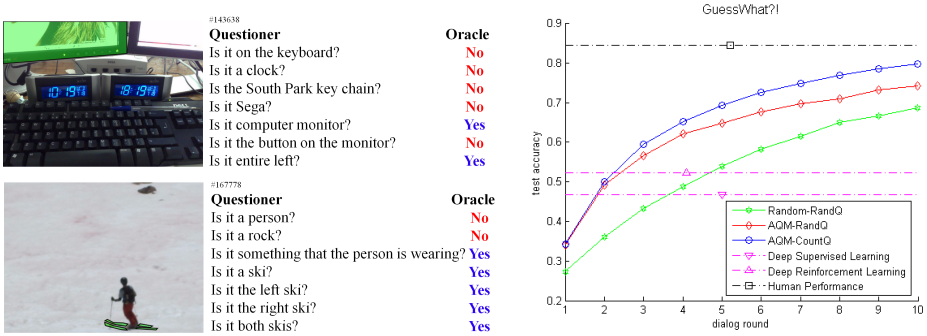

Figure 2: (Left) Examples of the GuessWhat?! game. A green mask highlights the correct object. (Right) Test accuracy from the GuessWhat?! task. When more questions are allowed, our main method reaches near-human performance. Previous works do not report the performance change with an increase in the number of turns.

Q-sampler for the Candidate Question Set We compare two Q-samplers in our paper. The first is 'randQ,' which samples a question randomly from the training data. The second is 'countQ,' which makes every other question from the set Q to be not too dependent each other. countQ checks the dependency of two questions by 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. The output of the answer-generator module for the training instance is used for the count. The procedure is not computationally expensive, because the size of Q and the answer set {yes, no, n/a} is small. We set the size of Q to 100.

Experimental Results Figure 2 (Right) shows the experimental results of the GuessWhat?! task. Our best algorithm, AQM-countQ, achieves 59.40% in three turns, outperforming deep SL and deep RL algorithms. By allowing more questions, our proposed algorithms remarkably improves accuracy and reaches near-human performance. AQM-countQ achieves 69.20% in five turns, and 79.81% in ten turns. The comparative deep SL method used the question-generator with hierarchical recurrent encoder-decoder (HRED) [14], achieving an accuracy of 46.8% in five turns [3]. The comparative deep RL method applies reinforcement learning on LSTM, achieving 52.3%[1] in around 4.1 turns [7]. 'Random' utilizes random question-generator module and AQM's guesser module.

5 Conclusion

We introduced an answerer in questioner's mind (AQM) framework, a novel approach for goal-oriented dialogue. Our framework is simple from the optimization perspective. AQM decreases the complexity in generating a series of questions, by substituting the hidden vector of the recurrent neural networks in the classical neural question-generator with the posterior of the class c. The mechanism of our framework is not a black-box but is driven from the perspective of information theory. We believe that the AQM approach can be utilized further for practical goal-oriented dialogue systems, including digital personal assistants and intelligent servers in restaurant and cafe. We think that AQM is also potentially advantageous when target answer distribution is non-stationary or multi-domain, because of AQM's simplicity and interpretability.

Acknowledgements

The authors would like to thank Jin-Hwa Kim, Cheolho Han, Wooyoung Kang, Jaehyun Jun, Christina Baek, and Hanock Kwak for helpful comments and editing. This work was supported by the Institute for Information & Communications Technology Promotion (R0126-16-1072-SW.StarLab, 2017-0-01772-VTT) and Korea Evaluation Institute of Industrial Technology (10060086-RISF) grant funded by the Korea government (MSIP, DAPA).

[1] 58.4%, the updated score of the refactored code is reported at Github (the averaged turn is unknown). https://github.com/GuessWhatGame/guesswhat

References

[1] Oriol Vinyals and Quoc Le. A neural conversational model. In *ICML deep learning workshop*, 2015.

[2] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784, 2016.

[3] Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585*, 2017.

[5] Tiancheng Zhao and Maxine Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*, 2016.

[6] Xuijun Li, Yun-Nung Chen, Lihong Li, and Jianfeng Gao. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*, 2017.

[7] Florian Strub, Harm de Vries, Jeremie Mary, Bilal Piot, Aaron Courville, and Olivier Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. *arXiv preprint arXiv:1703.05423*, 2017.

[8] Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. Evaluating visual conversational agents via cooperative human-ai games. *arXiv preprint arXiv:1708.05122*, 2017.

[9] Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. In *ICLR*, 2017.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[11] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. Visual reference resolution using attention memory for visual dialog. In *Advances in neural information processing systems*, 2017.

[13] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[14] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 2015.