Large-Scale Answerer in Questioner's Mind for Visual Dialog Question Generation

Sang-Woo Lee, Tong Gao, Sohee Yang, Jaejun Yoo, & Jung-Woo Ha
Clova AI Research, NAVER Corp.
{sang.woo.lee,tong.gao,sh.yang,jaejun.yoo,jungwoo.ha}@navercorp.com

Abstract

Answerer in Questioner's Mind (AQM) is an information-theoretic framework that has been recently proposed for task-oriented dialog systems. AQM benefits from asking a question that would maximize the information gain when it is asked. However, due to its intrinsic nature of explicitly calculating the information gain, AQM has a limitation when the solution space is very large. To address this, we propose AQM+ that can deal with a large-scale problem and ask a question that is more coherent to the current context of the dialog. We evaluate our method on GuessWhich, a challenging task-oriented visual dialog problem, where the number of candidate classes is approximately 10K. Our experimental results and ablation studies show that AQM+ outperforms the state-of-the-art models by a remarkable margin with a reasonable approximation. Based on our results, we argue that AQM+ is a general task-oriented dialog algorithm that can be applied for non-yes-or-no responses.

1 Introduction

Recent advances in deep learning have led an end-to-end neural approach [1–5] to task-oriented dialog problems that can reduce a laborious labeling task on states and intents [6, 7]. Lee et al. have recently proposed "Answerer in Questioner's Mind" (AQM) algorithm that does not depend on a limited capacity of RNN models to cover an entire dialog [8]. AQM treats the task-oriented dialog problem as twenty question games and selects the question that gives a maximum information gain. Unlike the other approaches, AQM benefits from explicitly calculating the posterior distribution and finding a solution analytically. The authors showed promising results in the task-oriented dialog problem, such as GuessWhat [9], where a questioner tries to find an object that is in answerer's mind via a series of Yes/No questions. The candidates are confined to the objects that are presented in the given image (less than ten on average). However, this simplified task may not be general enough to practical problems where the numbers of objects, questions and answers are typically unrestricted. For example, GuessWhich is a generalized version of GuessWhat that has a greater number of class candidates (9,628 images) and the dialogs consists of sentences beyond yes or no [10].

To address this, we propose a more generalized version of AQM, dubbed AQM+. Compared to the original AQM, the proposed AQM+ can easily handle the increased number of questions, answers, and candidate classes by employing an approximation based on subset sampling. Because our algorithm considers the previous history of the dialog, AQM+ can generate a more contextual question. Our main contributions are summarized as follows: 1) We propose AQM+ that extends the AQM framework toward more general and complicated tasks. AQM+ can handle a more complicated problem where the number of candidate classes is extremely large. 2) At every turn, AQM+ generates a question considering the context of the previous dialog, which is desirable in practice. 3) AQM+ outperforms comparative deep learning models by a large margin in Guesswhich, a challenging task-oriented visual dialog task.

NeurIPS 2018 Workshop on Visually Grounded Interaction and Language (ViGIL), Montreal, Canada

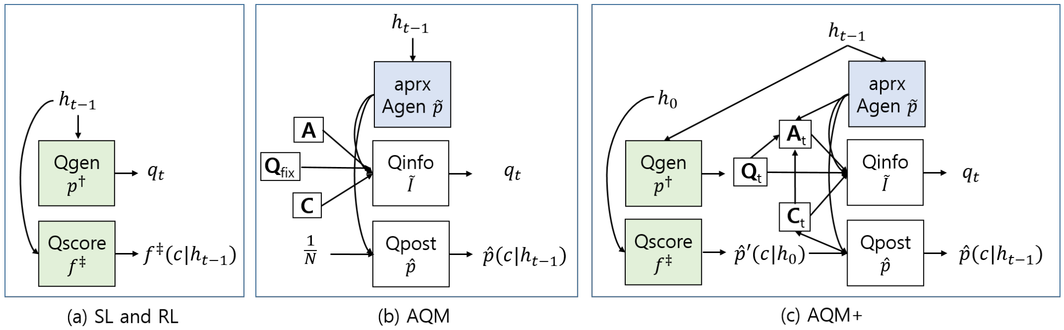

Figure 1: Modules in AQM+ and comparative models. SL and RL have their main neural modules as Qgen $p^\dagger$ and Qscore $f^\ddagger$, while AQM has aprxAgen $\tilde{p}$ used for Qpost $\tilde{I}$ and Qinfo $\hat{p}$. AQM+ contains all five modules and uses these to make subsets $\mathbf{Q}_t$, $\mathbf{A}_t$, and $\mathbf{C}_t$, thus achieving approximated estimation on information gain for large-scale inference, along with efficient contextual question generation.

2 Algorithm: AQM+

Problem Setting In our experiments, a questioner bot (Qbot) and an answerer bot (Abot) cooperatively communicate to achieve the goal via natural language. Under the AQM framework, at each turn t, Qbot generates an appropriate question q_t and guesses the target class c given a previous history of the dialog $h_{t-1} = (q_{1:t-1}, a_{1:t-1}, h_0)$. Here, a_t is the t-th answer and h_0 is an initial context that can be obtained before the start of the dialog. We refer to the random variables of target class and the t-th answer as C and A_t, respectively. Note that the t-th question is not a random variable in our information gain calculation. To distinguish from the random variables, we use a bold face for a set notation of target class, question, and answers; i.e. $\mathbf{C}$, $\mathbf{Q}$, and $\mathbf{A}$.

Preliminary: Supervised Learning, Reinforcement Learning, and AQM Approaches In supervised learning (SL) and reinforcement learning (RL) approaches [10–12], Qbot consists of two RNN modules. One is “Qgen", a question generator finding the solution that maximizes its distribution $p^\dagger$; i.e. $q_t^* = \operatorname{argmax} p^\dagger(q_t|h_{t-1})$. The other is a “Qscore”, a class guesser using the score function for each class $f^\ddagger(c|h_t)$. Two RNN modules can be fully separated into two RNNs [13], or can share some recurrent layers but have different output layers from each other [10].

On the other hand, in the previous AQM approach [8], these two RNN-based models are substituted to the calculation that explicitly finds an analytic solution. It finds a question that maximizes information gain or mutual information $\tilde{I}$, i.e. $q_t^* = \operatorname{argmax}_{q_t \in \mathbf{Q}_{fix}} \tilde{I}[C, A_t; q_t, h_{t-1}]$, where

$$\tilde{I}[C, A_t; q_t, h_{t-1}] = \sum_{c \in \mathbf{C}} \sum_{a_t \in \mathbf{A}} \hat{p}(c|h_{t-1}) \tilde{p}(a_t|c, q_t, h_{t-1}) \ln \frac{\tilde{p}(a_t|c, q_t, h_{t-1})}{\tilde{p}'(a_t|q_t, h_{t-1})}, \tag{1}$$

$$\hat{p}(c|h_t) \propto \hat{p}'(c|h_0) \prod_{j=1}^{t} \tilde{p}(a_j|c, q_j, h_{j-1}) = \hat{p}(c|h_{t-1}) \tilde{p}(a_t|c, q_t, h_{t-1}). \tag{2}$$

Here, a posterior function $\hat{p}$ can be calculated with the following equation in a sequential way, where $\hat{p}'$ is a prior function given h_0. In AQM, Equation 1 and Equation 2 can be explicitly calculated from the model. For ease of reference, let us name every component one by one. A module that calculates an information gain $\tilde{I}$ is referred to as “Qinfo" and a module that finds an approximated answer distribution $\tilde{p}(a_t|c, q_t, h_{t-1})$ is referred to as “aprxAgen". In AQM, aprxAgen is a model distribution that Qbot has in mind where the target is the true distribution of an answer generator $\bar{p}(a_t|c, q_t, h_{t-1})$, which is referred to as “Agen”. Finally, “Qpost" denotes a posterior $\hat{p}$ calculation module for guessing a target class.

As AQM uses the full set of $\mathbf{C}$ and $\mathbf{A}$, the complexity depends on the size of $\mathbf{C}$ and $\mathbf{A}$. For the question selection, AQM uses a predefined set of candidate questions ($\mathbf{Q}_{fix}$), which is not changed for different turn.

Table 1: Test percentile mean rank (PMR) in the 10-th round. Baseline refers the 0-th round PMR of SL-Q. The results of comparative deep models in the non-delta setting are from [10]. Baseline is the 0-th turn performance of SL-Q.

	Baseline	SL-Q	RL-QA	AQM+ w/ indA	AQM+ w/ depA	AQM+ w/ trueA
non-delta	88.5	90.9	93.3	94.64	97.45	99.87
delta	95.45	95.72	95.69	97.17	98.25	99.22

(a) Non-Delta Hyperparameter Setting

(b) Delta Hyperparameter Setting

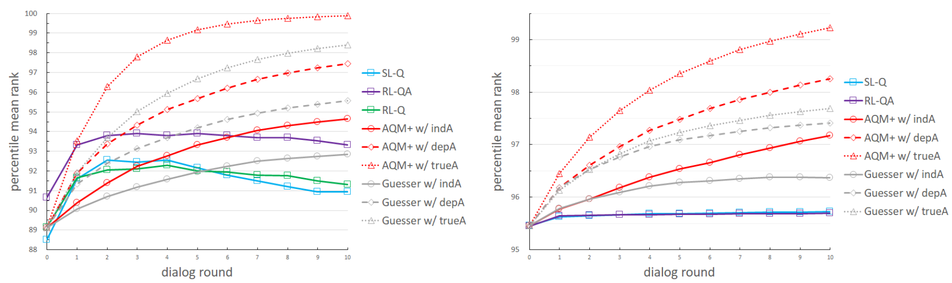

Figure 2: Test percentile mean ranks on GuessWhich experiments.

AQM+ Algorithm In this paper, we propose AQM+ algorithm, which uses sampling-based approximation, for tackling the large-scale task-oriented dialog problem. The core differences of AQM+ from the previous AQM are Infogain_topk explained as follows:

$$\tilde{I}_{topk}[C, A_t; q_t, h_{t-1}]$$
$$= \sum_{a_t \in \mathbf{A}_{t,topk}(q_t)} \sum_{c \in \mathbf{C}_{t,topk}} \hat{p}_{reg}(c|h_{t-1}) \tilde{p}_{reg}(a_t|c, q_t, h_{t-1}) \ln \frac{\tilde{p}_{reg}(a_t|c, q_t, h_{t-1})}{\tilde{p}'_{reg}(a_t|q_t, h_{t-1})}, \tag{3}$$

where $\hat{p}_{reg}$ and $\tilde{p}_{reg}$ are the normalized version of $\hat{p}$ over $\mathbf{C}_{t,topk}$ and $\tilde{p}$ over $\mathbf{A}_{t,topk}(q_t)$, respectively. Here, $\hat{p}_{reg}(c|h_{t-1}) = \hat{p}(c|h_{t-1}) / \sum_{c \in \mathbf{C}_{t,topk}} \hat{p}(c|h_{t-1})$, $\tilde{p}_{reg} = \tilde{p}(a_t|c, q_t, h_{t-1}) / \sum_{a_t \in \mathbf{A}_{t,topk}(q_t)} \tilde{p}(a_t|c, q_t, h_{t-1})$, and $\tilde{p}'_{reg}(a_t|q_t, h_{t-1}) = \sum_{c \in \mathbf{C}_{t,topk}} \hat{p}_{reg}(c|h_{t-1}) \cdot \tilde{p}_{reg}(a_t|c, q_t, h_{t-1})$.

Each set is constructed by the following procedures.

- $\mathbf{C}_{t,topk} \leftarrow$ top-K posterior test images (from Qpost $\hat{p}(c|h_{t-1})$)
- $\mathbf{Q}_{t,gen} \leftarrow$ top-K likelihood questions using the beam search (from Qgen $p^{\dagger}(q_t|h_{t-1})$)
- $\mathbf{A}_{t,topk}(q_t) \leftarrow$ top-1 generated answers from aprxAgen for each question q_t and each class in $\mathbf{C}_{t,topk}$ (from aprxAgen $\tilde{p}(a_t|c, q_t, h_{t-1})$)

Learning In all SL, RL, and AQM frameworks, Qbot needs to be trained to approximate the answer-generating probability of Abot. In AQM approach, aprxAgen does not share the parameters with Agen, and therefore also needs to be trained to approximate Agen. AQM can train aprxAgen by the learning strategy of the SL or RL approach. We explain two learning strategies of AQM framework below: indA and depA. In SL approach, Qgen and Qscore are trained from the training data, which have the same or similar distribution to that of the training data used in training Abot. Likewise, in indA setting of AQM approach, aprxAgen is trained from the training data. In RL approach, Qbot uses dialogs made by the conversation of Qbot and Abot and the result of the game as the objective function (i.e. reward). Likewise, in depA setting of AQM approach, aprxAgen is trained from the questions in the training data and following answers obtained in the conversation between Qbot and Abot. We also use the term trueA, referring to the setting where aprxAgen is the same as Agen, i.e. they share the same parameters. Both the previous AQM algorithm and the proposed AQM+ algorithm use these learning strategies.

Figure 3: Qualitative results on image retrieval of AQM+. Left column shows true images and their corresponding caption, and right column contains selected top-k images.

3 Experiments on GuessWhich

GuessWhich Task GuessWhich is a two player game played by Qbot and Abot. The goal of GuessWhich is to figure out a correct answer out of 9,628 test images by asking a sequence of questions. Abot can see the randomly assigned target image, which is unknown to Qbot. Qbot only observes a caption of the image generated from Neuraltalk2 [14]. To achieve the goal, Qbot asks a series of questions, to which Abot responds with a sentence.

Comparative Models We compare AQM+ with three comparative models, SL-Q, RL-Q, and RL-QA [10]. In SL-Q, Qbot and Abot are trained separately from the training data. In RL-Q, Qbot is initialized by the Qbot trained by SL-Q and then is fine-tuned by RL. Abot is the same as the Abot trained by SL-Q, and is not fine-tuned further. In the original paper [10], it was referred to as Frozen-A. By the way, in an RL-QA setting, not only Qbot but also Abot is concurrently trained with Qbot. In the original paper, it was referred to as RL-full-QAf. We also compare our AQM+ with "Guesser" algorithm. Guesser asks a question generated from SL-Q algorithm and calculates posterior by Qpost of AQM+. We use percentile mean rank (PMR) as the performance measure. Here, 93.3% of PMR at the zeroth turn means that the model can predict the correct image to be more likely than the other 8,983 images out of 9,628 candidates after exploiting the caption information solely.

Non-delta vs. Delta Hyperparameter The important issue in our GuessWhich experiment is delta setting. The non-delta setting is the setting in the original paper, and the delta setting is the another hyperparameter setting, which is discovered in Github code[1] after the presentation of the original paper. We use both non-delta setting and delta setting to test the performance of AQM+.

Other Experimental Setting As shown in Figure 1, our model uses five modules, Qgen, Qscore, aprxAgen, Qinfo, and Qpost. We use the same Qgen and Qscore modules as the comparative SL-Q model. The prior function is obtained from $\hat{p}'(c|h_0) \propto exp(\lambda \cdot f^{\ddagger}(c|h_0))$ using Qscore, where λ is a balancing hyperparameter between prior and likelihood. We set the size of the sets $|\mathbf{C}_{t,topk}| = |\mathbf{Q}_{t,gen}| = |\mathbf{A}_{t,topk}(q_t)| = 20$.

Experimental Results Figure 2 shows the PMR of the target image for our AQM+ and comparative models across the rounds. Figure 2a corresponds to the non-delta setting in the original paper [10] and Figure 2b corresponds to the delta setting proposed in the Github code. Figure 3 shows the top-k images selected by AQM+'s posterior. Non-delta and indA setting is used. The figure shows that relevant images to the caption remained after few dialog turns. The bottom number in the image denotes posterior of the image AQM+ thinks of.

4 Conclusion

Asking appropriate questions in practical applications has recently been paid attention [15, 16]. We proposed AQM+ algorithm that is a large-scale extension of AQM framework. AQM+ can ask an appropriate question considering the context of the dialog, handle the responses in a sentence form, and efficiently estimate information gain of the target class with a given question. The performance of AQM+ can be boosted further by employing the models recently proposed in the visual dialog field such as other question generator models [11] and question answering models [17].

[1] https://github.com/batra-mlp-lab/visdial-rl

Acknowledgments

This work was supported by the Creative Industrial Technology Development Program (10053249) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

References

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

[2] Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. Query-reduction networks for question answering. *arXiv preprint arXiv:1606.04582*, 2016.

[3] Mihail Eric and Christopher D Manning. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *arXiv preprint arXiv:1701.04024*, 2017.

[4] Anonymous. Codraw: Collaborative drawing as a testbed for grounded goal-driven communication. In *Submitted to International Conference on Learning Representations*, 2019. under review.

[5] Anonymous. Graph convolutional network with sequential attention for goal-oriented dialogue systems. In *Submitted to International Conference on Learning Representations*, 2019. under review.

[6] Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew Stuttle. 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*, pages 119–122. Association for Computational Linguistics, 2006.

[7] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[8] Sang-Woo Lee, Yu-Jung Heo, and Byoung-Tak Zhang. Answerer in questioner's mind for goal-oriented visual dialogue. In *Advances in Neural Information Processing Systems*, 2018.

[9] 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.

[10] Abhishek Das, Satwik Kottur, Jose MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2970–2979. IEEE, 2017.

[11] Unnat Jain, Svetlana Lazebnik, and Alexander G Schwing. Two can play this game: Visual dialog with discriminative question generation and answering. In *Proc. CVPR*, volume 1, 2018.

[12] Jiaping Zhang, Tiancheng Zhao, and Zhou Yu. Multimodal hierarchical reinforcement learning policy for task-oriented visual dialog. *arXiv preprint arXiv:1805.03257*, 2018.

[13] 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.

[14] Oriol Vinyals and Quoc Le. A neural conversational model. In *ICML Deep Learning Workshop*, 2015.

[15] Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *arXiv preprint arXiv:1805.04655*, 2018.

[16] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. Ask the right questions: Active question reformulation with reinforcement learning. In *ICLR*, 2018.

[17] Satwik Kottur, Jose MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–169, 2018.