

Visual Dialogue State Tracking for Question Generation

Wei Pang, Xiaojie Wang*

Center for Intelligence Science and Technology, School of Computer Science,
Beijing University of Posts and Telecommunication
{pangweitf, xjwang}@bupt.edu.cn

Abstract

GuessWhat?! is a visual dialogue task between a guesser and an oracle. The guesser aims to locate an object supposed by the oracle oneself in an image by asking a sequence of Yes/No questions. Asking proper questions with the progress of dialogue is vital for achieving successful final guess. As a result, the progress of dialogue should be properly represented and tracked. Previous models for question generation pay less attention on the representation and tracking of dialogue states, and therefore are prone to asking low quality questions such as repeated questions. This paper proposes visual dialogue state tracking (VDST) based method for question generation. A visual dialogue state is defined as the distribution on objects in the image as well as representations of objects. Representations of objects are updated with the change of the distribution on objects. An object-difference based attention is used to decode new question. The distribution on objects is updated by comparing the question-answer pair and objects. Experimental results on GuessWhat?! dataset show that our model significantly outperforms existing methods and achieves new state-of-the-art performance. It is also noticeable that our model reduces the rate of repeated questions from more than 50% to 21.9% compared with previous state-of-the-art methods.

Introduction

Visual dialogue is an increasing interest of multi-modal task, which involves vision, language and reasoning between them in a continuous conversation. Recently, researchers have put forward many different tasks of visual dialogue, including VisDial (Das et al. 2017a), GuessWhat?! (de Vries et al. 2017), GuessWhich (Chattopadhyay et al. 2017), Image-Grounded Conversation (IGC) (Mostafazadeh et al. 2017), Multimodal dialogs (MMD) (Saha, Khapra, and Sankaranarayanan 2018) and so on. Among them, GuessWhat?! is an object-guessing game played by two players - a Questioner and an Oracle. Given an image including several objects, the goal of the Questioner is to locate a target object - supposed by the oracle oneself at the beginning of a game by asking a series of Yes/No questions. Specially,

*Corresponding author.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



Figure 1: The left part gives an example of GuessWhat?! game. The right part illustrates updates of visual dialogue states, including the distribution on objects (numbers on colorful strips) as well as representations of objects (colorful strips).

the Questioner has two sub-tasks, one is Question Generator (QGen) that generates visually grounded questions, the other is Guesser that is to identify the target object given the full dialog context. The Oracle answers questions with yes or no. The left part of Fig.1 shows a game played by the QGen, Oracle and Guesser. Among them, QGen plays the most important role. Much work has been done on QGen (de Vries et al. 2017; Strub et al. 2017; Zhao and Tresp 2018; Zhang et al. 2018; Lee, Heo, and Zhang 2018; Shekhar et al. 2018; Abbasnejad et al. 2018; Shukla et al. 2019; Shekhar et al. 2019; Abbasnejad et al. 2019), this paper also focuses on QGen.

De Vries et al. (de Vries et al. 2017) proposed the first QGen model. It is an encoder-decoder model, where images are encoded by VGG FC8 features, and questions are decoded by an LSTM (Long Short-Term Memory) (Hochreiter and Schmidhuber 1997). It is trained in a supervised learning (SL) way. Strub et al. (Strub et al. 2017) introduced reinforcement learning (RL) to train the QGen

model and gained significant improvements. Various of reward functions (Strub et al. 2017; Zhao and Tresp 2018; Zhang et al. 2018; Abbasnejad et al. 2018; 2019; Shukla et al. 2019) have been proposed for training QGen model towards producing goal-oriented questions.

However, most of previous work employed unchanged visual representations through a dialogue. For example, De Vries and Strub et al. (de Vries et al. 2017; Strub et al. 2017) encoded a raw image into a fixed-size vector using pre-trained VGG network and used the same vector as the image feature for question generation in each round of dialog. Zhao et al. (Zhao and Tresp 2018; Zhang et al. 2018; Shekhar et al. 2018; Abbasnejad et al. 2018; Shekhar et al. 2019; Shukla et al. 2019) employed similar methods on dealing with images. Visual information keeps statically unchanged through the dialogue.

Intuitively, with the progress of a GuessWhat?! dialog, a questioner will focus on different parts of an image. For example, initially, the questioner might pay one’s attention to all objects in the image equally. When he/she asks a question “Is it a person?” and receives a positive answer, his/her attention on the image will be changed, more attention might be paid on person objects.

More importantly, with the change of attention on different objects, the representation of the objects might be changed. For example, when a positive answer for the question “Is it a person?” is received, the questioner will pay attention on more detailed information of person objects, such as face, cloth. Therefore, the representation of person objects will be changed to reflect these details. While for non-person objects, their representation might become blurred for saving memory resources.

Borrowing an item from text-based dialogue (Henderson, Thomson, and Young 2014; Wu et al. 2019), we call the 2-tuple \langle distribution on objects, representations of objects \rangle visual dialogue state (visual state or dialogue state in briefly). As the dialog progress, visual state will be changed, which prompts the questioner to ask new questions (as shown in the right part of Fig.1).

This paper proposes a visual dialogue state tracking (VDST) based QGen model to implement above ideas. The model includes a visual-language-visual multi-step reasoning cycle. Representations of objects are firstly updated by current distribution of image objects. A comparison between different objects is employed by a difference-operation-based attention which guided by objects themselves. Results of the attention is then used to decode new question. Finally, a cross-modal matching between question-answer pairs and object representations is designed for updating distribution of objects. Experimental results show our model achieves the new state-of-the-art performance on GuessWhat?! task. Especially, our model reduces the rate of repeated questions from more than 50% to 21.9% compared with previous state-of-the-art methods.

To summarize, our contributions are mainly three-fold:

- We propose a visual dialogue state tracking (VDST) based model for question generator in GuessWhat?!. The VDST, including not only distribution of objects but also representations of objects, is tracked and updated with the

progress of a dialogue.

- We achieve new state-of-the-art performances on QGen task of GuessWhat?! on four different training methods including a separate and a joint Supervised Learning, a separate Reinforcement Learning and a Cooperative Learning.
- We reduce the rate of repeated questions in dialogues from more than 50% to 21.9% compared with previous state-of-the-art methods.

Related Work

GuessWhat?! Game is one of the visual dialogue tasks introduced in (de Vries et al. 2017), which consists of three sub-tasks: Oracle, QGen and Guesser. Most of work on GuessWhat?! focuses on QGen because it plays a key role in the game.

De Vries et al. (de Vries et al. 2017) proposed the first QGen model. The entire image is encoded as a fixed-size vector (e.g., 1000-dim) that is extracted from VGG-fc8 layer (Simonyan and Zisserman 2015). The previous dialogues, such as question-answer pairs, are encoded by a hierarchical recurrent encoder. Question generation is performed by an LSTM where input is a concatenation of above VGG feature and representation of previous dialogues. They cast QGen task as a supervised learning (SL) task, and the model is trained by minimizing cross-entropy loss between the predicted question and the ground-truth question.

There are mainly two lines of ongoing work for QGen task. One is on model learning. Strub et al. (Strub et al. 2017) first cast QGen as a reinforcement learning (RL) task, they use a zero-one reward to lead QGen to produce goal-oriented questions, and achieve a significant improvement over the supervised model (de Vries et al. 2017). Zhang et al. (Zhang et al. 2018) add two intermediate rewards. Recently, to better explore the space of actions (word selections), Abbasnejad et al. (Abbasnejad et al. 2019) design reward for each action based on a Bayesian model. Zhao et al. (Zhao and Tresp 2018) calculate a temperature for each action based on action frequencies. In addition, Zhao et al. first train QGen and Guesser by reinforcement learning from self-play. Shekhar et al. (Shekhar et al. 2019) cast QGen and Guesser as a multi-task problem, and train the two models by cooperative learning (CL) (Das et al. 2017b).

Lee et al. (Lee, Heo, and Zhang 2018) use information gain to sample questions on the GuessWhat?! dataset, rather than to generate new questiones. Shukla et al. (Shukla et al. 2019) use regularized information gain to reward their QGen model.

The other line of work for QGen task is on encoding and fusion of visual and linguistic information through a dialogue. Following (de Vries et al. 2017), the methods in (Strub et al. 2017; Lu et al. 2017; Zhao and Tresp 2018; Zhang et al. 2018; Lee, Heo, and Zhang 2018; Shekhar et al. 2018; Henderson, Thomson, and Young 2014; Shukla et al. 2019; Abbasnejad et al. 2019) all obtain the image features from VGG fc-8 layer. Shekhar et al. (Shekhar et al. 2019) develop a common grounded dialogue state encoder that fuse both image features and linguistic representations, where the

image features (e.g., 2048-dim) are extracted from the second to last layer of ResNet152 (He et al. 2016), the linguistic representation is obtained by an LSTM which takes input the previous dialogue history. The two types of information are concatenated and passed through a multi-layer perceptron (MLP) layer to get an embedding, which is called the dialogue state and given as input to both QGen and Guesser.

Whatever the encoder used in above models, for a given image, the encoded vector for the image keeps fixed through a dialogue. Different from previous work, the model proposed in the paper will have changed representation of objects in each round of dialog. We will show it will bring significant improvement to question generation.

Model: Visual Dialogue State Tracking

The framework of our visual dialogue state tracking model (for short, VDST) is illustrated in Fig.2. In each round of dialog, there are four modules for QGen: Update of Object Representation (UoOR), Object-self Difference Attention (OsDA), Question Decoder (QDer) and Cross-Modal Matching (CMM). Where, UoOR updates representations of objects according to the distribution of objects, new representations are used by OsDA to induce attention vectors which are then used to generate new question by QDer, finally, CMM is employed to update the distribution of objects. Details of each module are given respectively in remainders of this sections.

Update of Object Representation (UoOR)

Object representations for bounding boxes or image regions are extracted via Faster-RCNN (Ren et al. 2015) as in Eq.1.

$$O^f = \text{RCNN}(\text{image}), \quad (1)$$

where $O^f \in \mathbb{R}^{m \times (d_s+8)}$ includes m objects $\{o_k\}_{k=1}^m$. For each object o_k , its representation is a concatenation of d_s -dim static feature provided by bottom-up-attention (Anderson et al. 2018) and 8-dim vector of the spatial coordinates $[x_{min}, y_{min}, x_{max}, y_{max}, x_{center}, y_{center}, w_{box}, h_{box}]$, where x, y denote the coordinates and w_{box}, h_{box} denote the width and height of an object. Following (de Vries et al. 2017), the coordinates are normalized by the image width and height so that its values range from -1 to 1.

To map o_k and embedding of words to the same dimension d , an MLP layer is used as in Eq.2,

$$O^{(0)} = \text{MLP}(O^f), \quad (2)$$

where $O^{(0)} \in \mathbb{R}^{m \times d}$ is viewed as a set of initial object representations for objects.

Let $\pi^{(j)} \in \mathbb{R}^m$ be probability distribution (or attention distribution) on m objects at the j th dialogue round. It gives the accumulative attention paid on each object up to j th round dialogue. Please note that $\pi^{(j)}$ is the attention distribution instead of a distribution of each object as target object which will be calculated by the Guesser. Let the initial distribution $\pi^{(0)}$ be a uniform distribution, which means each object has been paid equal attention at the beginning. Update of Object Representation (UoOR) is defined as in Eq.3.

$$O^{(j)} = (\pi^{(j)})^T O^{(0)}, \quad (3)$$

where $O^{(j)} \in \mathbb{R}^{m \times d}$ denotes the representations of objects at the j th round.

From Eq.3, we can observe that representations of objects are updated by the distribution of objects at each dialog round. Because the distribution of objects is updated at each dialog round (will be specified in CMM module), the representation for a same object at different dialog round is therefore different. It models the intuition: with the change of attention on different objects, the representation of the objects might be changed.

Object-self Difference Attention (OsDA)

We define Object-self Difference Attention (OsDA) at j th round of dialogue as in Eq.4,

$$v^{(j)} = \text{softmax}([o_i^{(j)} \odot (o_i^{(j)} - o_k^{(j)})]W)^T O^{(j)}. \quad (4)$$

Where $o_i^{(j)} \in O^{(j)}$, $o_i^{(j)} \odot (o_i^{(j)} - o_k^{(j)}) \in \mathbb{R}^d$ is difference between i th object and k th object under the guidance of the i th object itself, $W \in \mathbb{R}^{md \times g}$ is a parameter matrix to be learned with g glimpses. The differences between the i th object and all other objects are concatenated into a md -dimensional vector. For m objects, operation $[o_i^{(j)} \odot (o_i^{(j)} - o_k^{(j)})] \in \mathbb{R}^{m \times md}$ constructs a $m \times md$ matrix. The softmax transforms the glimpse results of differences to attention weight, and then output the final attentive visual feature $v^{(j)}$. It will be used as visual context to decode a question for next round of dialog.

We have two notes on OsDA.

Firstly, visual comparison between objects could provide visually discriminative cues and help quickly distinguish multiple objects in an image. Motivated by that, for a specify object $o_i^{(j)}$, comparison to the k th object as the subtraction of $o_i^{(j)}$ and $o_k^{(j)}$ under guidance of itself, namely the difference operation $o_i^{(j)} \odot (o_i^{(j)} - o_k^{(j)})$, is designed.

Secondly, we aggregate their differences and map them with g glimpse to an attention distribution over objects, and then we get the final visual context $v^{(j)}$. Note that here $v^{(j)}$ is dynamically changed at different round with the UoOR.

Question Decoder (QDer)

A one-layer LSTM is employed as QDer for generating questions, as shown in Eq.5,

$$w_{i+1}^{(j)} = \text{LSTM}([v^{(j)}; w_i^{(j)}]), \quad (5)$$

where $w_i^{(j)}$ denotes the i th word of question $q^{(j)}$, $[\cdot; \cdot]$ refers to concatenation. Note that the last hidden state $h^{(j)}$ of the LSTM decoder is used as the representation of $q^{(j)}$ as shown in Fig.2.

Cross-Modal Matching (CMM)

When answer $a^{(j)}$ for question $q^{(j)}$ is received from the Oracle, Questioner needs to renew his/her judgement on which

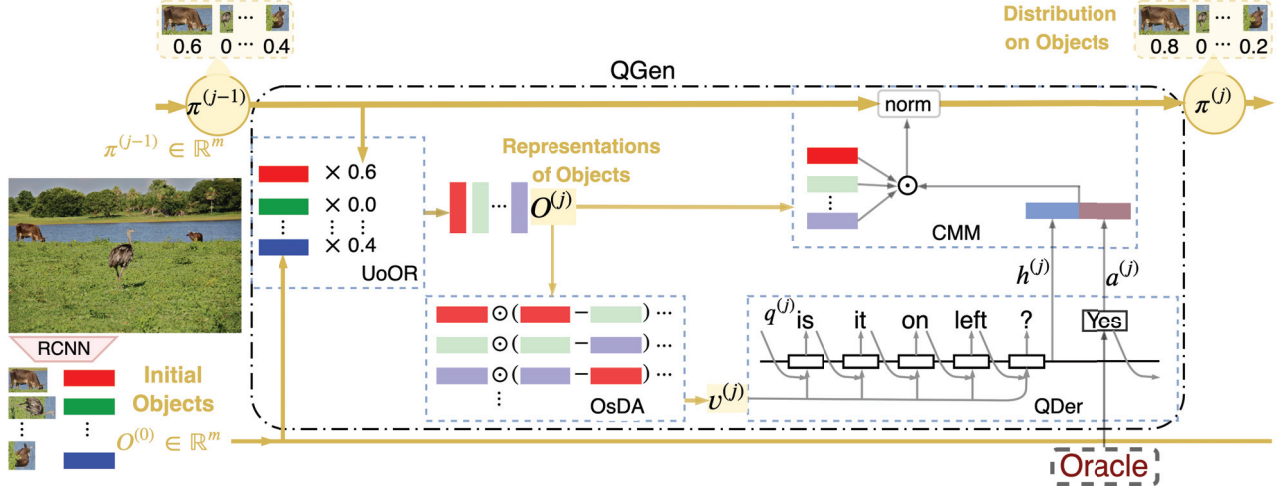


Figure 2: Overall structure of VDST model.

object should be paid more attention, i.e. the probability distribution on m objects, according to new information including in the question-answer pair. Three steps are designed for the update.

Firstly, the embedding of answer $a^{(j)}$ is concatenated to the representation of question, $h^{(j)}$, to form a representation of question-answer pair: $h_a^{(j)} = [h^{(j)}; a^{(j)}]$ as shown in Fig.2.

Secondly, a Cross-Modal Matching (CMM) network is defined as in Eq.6 to measure how many changes the new question-answer pair $\langle q^{(j)}, a^{(j)} \rangle$ bring to the distribution on objects.

$$\hat{\pi}^{(j)} = \text{softmax}\left(\frac{\tanh(O^{(j)}U^T \odot V^T h_a^{(j)})}{\sqrt{d}}\right) \quad (6)$$

Where $U \in \mathbb{R}^{d \times d}$, $V \in \mathbb{R}^{2d \times d}$ are learnable parameters, \sqrt{d} in denominator is used to control variance. \odot denotes Hadamard product used for fusion of multimodal information from visual object representations and textual question-answer pair. A softmax function is used to produce a probability distribution $\hat{\pi}^{(j)}$ over m objects, which is the change that $\langle q^{(j)}, a^{(j)} \rangle$ brings.

Thirdly, given the change $\hat{\pi}^{(j)}$ in this round and the accumulative probability distribution $\pi^{(j-1)}$ on objects at previous round. $\pi^{(j)}$, the accumulative probability distribution at j th round is defined as in Eq.7,

$$\pi^{(j)} = \text{norm}(\pi^{(j-1)} \cdot \hat{\pi}^{(j)}), \quad (7)$$

where norm is normalization operation to keep $\pi^{(j)}$ be a probability distribution.

Model Training

Because the paper focuses on visual state tracking, we following existing way for model training. As in (Strub et al.

2017), we first train QGen, Oracle and Guesser independently in supervised way with a cross-entropy loss, and then tune the QGen by RL with the oracle and guesser fixed.

The cross-entropy loss is defined as in Eq.8,

$$L_{SL}(\theta) = \sum_{t=1}^T -\log(p(w_t)), \quad (8)$$

where T is the length of a complete dialogue, $p(w_t)$ is the probability of the ground-truth word at step t in the dialogue, θ denotes the parameters.

A zero-one reward is used for RL. i.e. $r_D = 1$ for a successful guessing, otherwise, $r_D = 0$. The RL loss is defined as the negative expected reward:

$$L_{RL}(\theta) = -\mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^T (r_A(a_t) - b) \log \pi_\theta(a_t) \right], \quad (9)$$

where b is the baseline function the same as (Strub et al. 2017), a_t denotes an action (i.e. word) choice at each time step t . In the case of reward $r_A(a_t)$ for each action, we spread the reward r_D uniformly over the sequence of actions. The model can be trained end-to-end by minimizing the loss in Eq.9.

Experiments and Analysis

Dataset and Evaluation We evaluate our model on the GuessWhat?! dataset introduced in (de Vries et al. 2017). This dataset has 155k dialogues on 66k images and consists of 822k question/answer pairs, the vocabulary contains nearly 5k words. We use the standard partition of the dataset to the training (70%), validation (15%) and test (15%) set as in (de Vries et al. 2017; Strub et al. 2017).

Following (Strub et al. 2017), we report the task success rate by three inference methods, sampling, greedy and beam search (beam size 20) on New Object and New Game respectively. New Object refers to the target object is new but the image has been seen in training, while New Game refers to the image and target are all previously unseen.

Implementation Details We limit the maximum number of questions to 5 (5Q) (de Vries et al. 2017) and 8 (8Q) (Strub et al. 2017; Zhao and Tresp 2018; Shekhar et al. 2019; Shukla et al. 2019) respectively for comparisons. We set the maximum number of words to 12 as (Strub et al. 2017). For each image, 36 bounding boxes (objects) are extracted from RCNN, image features are 36×2056 . The dimension of word embedding is 512. The image features are mapped to 36×512 with a fully-connection layer follows a swish activation (Ramachandran, Zoph, and Le 2018). The LSTM hidden unit number in QDer is 512. Attention glimpse is 2. Our code and other materials will be published in the near future.

Supervised Learning We train the Guesser and Oracle model for 30 epochs, and pre-train the QGen model for 50 epochs, using Adam optimizer (Kingma and Ba 2015) with a learning rate of $1e-4$ and a batch size of 64. Note that pre-training the QGen is important to quickly approximate a relatively good policy for RL fine-tuning.

Reinforcement Learning We follow the same RL agent environment as (Strub et al. 2017). After supervised training, we then initialize the QGen environment with the pre-trained parameters from the above supervised learning, and post-train the QGen model with REINFORCE (Williams 1992; Sutton et al. 2000) for 500 epochs, using stochastic gradient descent (SGD) with a learning rate of $1e-3$ and a batch size of 64.

Baseline and SOTA models Four SL based models are compared in experiments, there are the first SL model (de Vries et al. 2017), GDSE model (Shekhar et al. 2018; 2019), the first RL model (Strub et al. 2017) pre-trained in SL phase and TPG model (Zhao and Tresp 2018) pre-trained in SL phase. Six RL based models are compared, there are the first RL model, TPG model, VQG (Zhang et al. 2018), ISM (Abbasnejad et al. 2018), Bayesian model (Abbasnejad et al. 2019) and RIG model (Shukla et al. 2019). Where the RL, TPG and RIG model used a zero-one reward, which is as the same as that in our model. The VQG model used more considerate rewards which integrates three intermediate rewards for policy learning, including goal-achieved reward (i.e. zero-one reward), progressive reward and informativeness reward.

Comparisons on Success Rate of Guessing

We compare our model with the previous state-of-the-art (SOTA) methods on success rate of guessing in the subsection. Tab.1 reports the performance for all models.

The first part of Tab.1 shows results by SL. Our model (VDST) outperforms three previous models. The second part of Tab.1 shows results by RL. It can be seen that our model not only outperforms all the models that use zero-one reward but also outperforms the models with more considerate reward significantly. Our model achieves a new state-of-the-art results on both New Object and New Game. Specifically, compared with the VQG, Bayesian and RIG models, our model achieves nearly 7 points promotions on New objects on Greedy case, and achieves nearly 6 points promotions on New games on Greedy way. Compared with GDSE-CL, VDST achieves 10.66% and 9.33% higher accuracy for 5Q and 8Q. On Greedy case in New games, compared with the

RL baseline, VDST improves task success rate by 5.96% for 5Q and 11.43% for 8Q; compared with RIG models, our model improves by more than 6.55%.

Some models employed cooperation learning to co-train QGen and Guesser to promote their successful rate, and achieve even better performance (Zhao and Tresp 2018; Abbasnejad et al. 2019). Such as Abbasnejad et al. (Abbasnejad et al. 2019) post-trained QGen and Guesser with RL and improved the accuracy from 62.1% to 69.2% on New Object, which gives a 7.1% improvement. The paper focuses on QGen itself, and does not implement cooperation learning experiments due to time and space limitation.

In summary, our model consistently achieves the best performances on success rate of guessing on both New objects and New games compares with existing baseline and SOTA models.

Comparisons on Quality of the Dialogues

We then compare the quality of dialogues, especially the quality of questions generated by different models with the 8Q setting. We follow (Shekhar et al. 2019) in which two linguistic measurements: percentages of repeated questions in a dialogue and lexical diversity, are used. Where the former measured as the amount of games that contain at least one repeated question over all games, the latter measured as the percentage of unique words over all games.

Repeated questions Tab.2 shows that our model has much lower percentage of repetitive questions in a dialogue than all the other models. Only 21.9% of dialogues contain repetitive questions, compared to 52.19% in GDSE-CL and higher ones in other models. This indicate that tracking and update of visual dialogue state in a dialog could improve the quality of the dialog significantly. It enables the guesser shift its attentions to new objects or new features so that it tends to ask new questions.

Lexical diversity Our model also improves lexical diversity from 0.115 in GDSE-CL to 0.15. It doubles the score that in other baseline models. It is only natural that a model asking more new questions on new objects or new features will bring more lexical diversity.

Overall, our model generates significantly fewer repeated questions than all the compared methods, and is therefore with richer words than that in others.

Strategy for questions asking learnt by our model

We further investigate the policy of question asking in dialogues learned by our model. The class labels and a classifier in (Shekhar et al. 2019) are used. There are entity question (e.g., “is it a person?”) and attribute question (e.g., “is it a red?”). Entity question labels can be either an object category (e.g., car) or a super-category (e.g., vehicle). Attribute question has 6 fine-grained subclasses including color, location, size, shape, texture and action.

When questions in GuessWhat?! human dialogues are labeled with the class labels by the classifier, we find that 22.38% of sequences of human questions in a single dialogue implement an interesting question asking policy. It begins with an entity question. If an answer “No” for the question is received, it keeps asking an entity question on other

Table 1: Task success rate for each model.

New Object				
	Sampling	Greedy	BSearch	Max Q's
SL (de Vries et al. 2017)	41.6	43.5	47.1	5
(Strub et al. 2017) (pre-trained in SL)	-	44.6	-	8
(Zhao and Tresp 2018) (pre-trained in SL)	-	48.77	-	8
VDST(ours, pre-trained in SL)	45.02	49.49	-	5
VDST(ours, pre-trained in SL)	46.70	48.01	-	8
New Game				
SL (de Vries et al. 2017)	39.2	40.8	44.6	5
SL (de Vries et al. 2017)	-	40.7	-	8
VDST(ours, pre-trained in SL)	42.92	45.94	-	5
VDST(ours, pre-trained in SL)	44.24	45.03	-	8
RL (Strub et al. 2017)	56.5	58.4	58.4	5
RL (Strub et al. 2017)	60.8	56.3	52.0	8
VQG (Zhang et al. 2018)	59.8	60.7	60.8	5
ISM (Abbasnejad et al. 2018)	-	62.1	-	-
GDSE-SL (Shekhar et al. 2019)	-	47.8	-	5
GDSE-SL (Shekhar et al. 2019)	-	49.7	-	8
GDSE-CL (Shekhar et al. 2019)	-	53.7	-	5
GDSE-CL (Shekhar et al. 2019)	-	58.4	-	8
Bayesian (Abbasnejad et al. 2019)	59.0	59.8	60.6	5
RIG as rewards (Shukla et al. 2019)	64.06	59.0	60.21	8
RIG as a loss with 0-1 rewards (Shukla et al. 2019)	65.79	61.18	59.79	8
VDST(ours)	63.85	64.36	64.44	5
VDST(ours)	66.76	67.73	67.52	8

Table 2: Comparisons on quality of the dialogues.

	%Games with repeated Q's	Lexical diversity
SL (de Vries et al. 2017)	93.50	0.030
RL (Strub et al. 2017)	96.47	0.073
GDSE-SL (Shekhar et al. 2019)	55.80	0.101
GDSE-CL (Shekhar et al. 2019)	52.19	0.115
VDST(ours)	21.9	0.150
Human	-	0.731

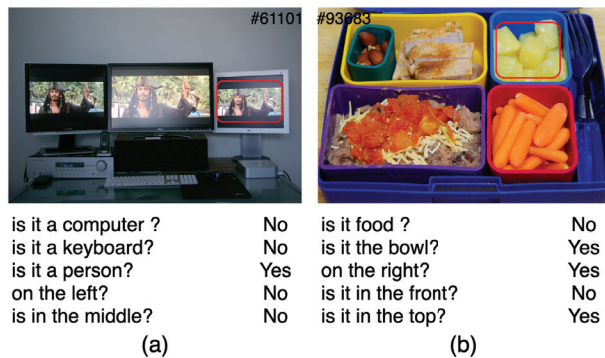


Figure 3: Questions generated by our VDST model. The target object is highlighted in red box.

categories until an answer “Yes” is received. It then shifts to ask an attribute question such as location questions, and keeps asking this kind of questions to the end of dialog.

We label the questions generated by our model in the same way as in above. We find our model learn the above policy well. In 67.88% of dialogues, the sequences of questions

Table 3: Experimental results of ablation studies. Row 2-3 shows the effect of each module by removing it from the full model.

		New Object	
#	Model	Sampling	Greedy
1	VSDT(full model)	66.22	67.07
2	-UoOR&CMM	62.73	64.30
3	-OsDA	64.06	65.21
		New Game	
1	VSDT(full model)	63.85	64.36
2	-UoOR&CMM	60.07	60.93
3	-OsDA	60.89	61.71

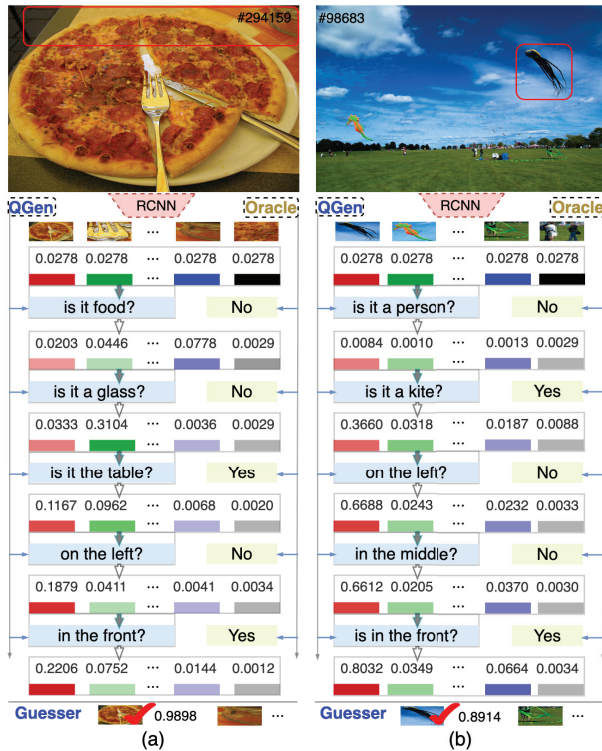


Figure 4: Two successful examples show the tracking procedure of VDST.

generated by our model implement this strategy.

Fig.3 gives two successful sequences of questions generated by our model in 5Q setting. In 3(a), the agent starts by an entity question “is it a computer?”, and keeps on asking entity question when a “yes” is received. The agent now identifies the category of target object, and then turns to ask attribute questions. Here the location question is chosen. It finally guesses the target object. In 3(b), one more success case is shown.

Ablation Studies on Major Modules

In this section, we conduct some ablation experiments to illustrate the effect of each major module in our model. Tab.3 shows the results for 5Q.

Firstly, we considerate the effect of update of the visual dialogue state which consist of the distribution of object and the representations of objects. The distribution of object is updated by CMM module and the representations of objects are updated by UoOR. As we can see, the performance of a model without UoOR and CMM drop nearly 3 points as shown in Tab.3. It shows visual dialogue state tracking plays a key role in the model. Please note that there is no meaning to employ UoOR without CMM or vice versa, because the visual state in our model is a 2-tuples.

Secondly, we considerate the effect of OsDA which is used to obtain attention for question generation. For removing OsDA, we replace the Eq.4 by an average representation of $O^{(j)}$. Experimental results show OsDA helps the model very much, especially in New Game case.

Qualitative Evaluation

In Fig.4, we give two examples to visualize the tracking procedure in our model;. We only show 4 objects and other objects are omitted for simplicity in each example. The colourful strips denote representations of objects, and the number values on strips denote their probabilities in the attention distribution on objects.

Taking Fig.4(a) as an example, we give a simple explanation. At the beginning of the game, representations of all objects are given by RCNN, all objects have equal probability (0.0278 as shown in the figure) as target. Basing on the initial information, the QGen first generates a question “is it food?”, and receives an answer “no” from the oracle. Then the attention probabilities of each object is changed, such as attention on the food object (pizza shown in the last column) drops to 0.0029. After round 3, attention on the target object (shown in the first column) increases to 0.1167. Meanwhile, with its increased attention, the representation of the target object is updated as shown in colour variations. In 4(b), at round 2, the target object has already obtained the maximum attention (36.6%) compared to other objects, begins at round 3, the agent focuses on asking location questions about the target object, and pays much less attention to others.

Conclusion

This paper presents a visual dialogue state tracking (VDST) based QGen model. The visual dialogue state, include distribution on objects as well as representations of objects. They are tracked and updated with the progress of a dialogue. With the update of object distribution, the representations of objects are updated, and prompt the questioner to ask new questions. Experimental results on the GuessWhat?! dataset show the model achieves the new state-of-the-art in both settings of supervised learning and reinforcement learning. The model reduces the rate of repeated questions from more than 50% to 21.9% compared with previous state-of-the-art methods. There are lots of problems to be resolved in the future, such as how to learn more flexible policy of asking questions, when to stop the question asking and turn to guess.

Acknowledgements

We thank the reviewers for their comments and suggestions. This paper is supported by NSFC (No. 61906018), Huawei Noah's Ark Lab and MoE-CMCC "Artificial Intelligence" Project (No. MCM20190701).

References

- Abbasnejad, E.; Wu, Q.; Abbasnejad, I.; Shi, J.; and van den Hengel, A. 2018. An active information seeking model for goal-oriented vision-and-language tasks. *arXiv preprint arXiv:1812.06398*.
- Abbasnejad, E.; Wu, Q.; Shi, J.; and van den Hengel, A. 2019. What's to know? uncertainty as a guide to asking goal-oriented questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. of CVPR*.
- Chattopadhyay, P.; Yadav, D.; Prabhu, V.; Chandrasekaran, A.; Das, A.; Lee, S.; Batra, D.; and Parikh, D. 2017. Evaluating visual conversational agents via cooperative human-ai games. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*.
- Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D. 2017a. Visual dialog. In *Proc. of CVPR*.
- Das, A.; Kottur, S.; Moura, J. M.; Lee, S.; and Batra, D. 2017b. Learning cooperative visual dialog agents with deep reinforcement learning. In *International Conference on Computer Vision (ICCV)*.
- de Vries, H.; Strub, F.; Chandar, S.; Pietquin, O.; Larochelle, H.; and Courville, A. C. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proc. of CVPR*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. of CVPR*.
- Henderson, M.; Thomson, B.; and Young, S. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proc. of SIGDIAL*.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory.
- Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
- Lee, S.-W.; Heo, Y.-J.; and Zhang, B.-T. 2018. Answerer in questioner's mind: Information theoretic approach to goal-oriented visual dialog. In *The Thirty-second Annual Conference on Neural Information Processing Systems (NeurIPS)*.
- Lu, J.; Kannan, A.; Yang, J.; Parikh, D.; and Batra, D. 2017. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *Proc. of NeurIPS*.
- Mostafazadeh, N.; Brockett, C.; Dolan, B.; Galley, M.; Gao, J.; Spithourakis, G. P.; and Vanderwende, L. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proc. of CVPR*.
- Ramachandran, P.; Zoph, B.; and Le, Q. V. 2018. Searching for activation functions. In *Proc. of ICLR*.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. of NeurIPS*.
- Saha, A.; Khapra, M. M.; and Sankaranarayanan, K. 2018. Towards building large scale multimodal domain-aware conversation systems. In *Association for the Advancement of Artificial Intelligence (AAAI)*.
- Shekhar, R.; Baumgärtner, T.; Venkatesh, A.; Bruni, E.; Bernardi, R.; and Fernández, R. 2018. Ask no more: Deciding when to guess in referential visual dialogue. In *International Conference on Computational Linguistics (COLING)*.
- Shekhar, R.; Venkatesh, A.; Baumgärtner, T.; Bruni, E.; Plank, B.; Bernardi, R.; and Fernández, R. 2019. Beyond task success: A closer look at jointly learning to see, ask, and guesswhat. In *the 2019 Annual Conference of the North American Chapter of the Association (NAACL)*.
- Shukla, P.; Elmadjian, C.; Sharan, R.; Kulkarni, V.; Wang, W. Y.; and Turk, M. 2019. What should i ask? using conversationally informative rewards for goal-oriented visual dialogue. In *Association for Computational Linguistics (ACL)*.
- Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR*.
- Strub, F.; Vries, H. D.; Mary, J.; Piot, B.; Courville, A.; and Pietquin, O. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine learning*.
- Wu, C.-S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; ; and Fung, P. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proc. of ACL*.
- Zhang, J.; Wu, Q.; Shen, C.; Zhang, J.; Lu, J.; and van den Hengel, A. 2018. Asking the difficult questions: Goal-oriented visual question generation via intermediate rewards. In *European Conference on Computer Vision (ECCV)*.
- Zhao, R., and Tresp, V. 2018. Improving goal-oriented visual dialog agents via advanced recurrent nets with tempered policy gradient. In *Proc. of IJCAI*.