In: *Proceedings of the 31st International Conference on Artificial Neural Networks (ICANN 2022)*, LNCS, Vol 13530, pp. 123-135, Bristol, UK, Sep 6–9, 2022

Word-by-Word Generation of Visual Dialog using Reinforcement Learning*

Yuliia Lysa, Cornelius Weber, Dennis Becker, and Stefan Wermter

Knowledge Technology Research Group, University of Hamburg, Hamburg, Germany
yuliia.lysa@studium.uni-hamburg.de, {cornelius.weber, dennis.becker-1, stefan.wermter}@uni-hamburg.de
www.knowledge-technology.info

Abstract. The task of visual dialog generation requires an agent holding a conversation referencing question history, putting the current question into context, and processing visual content. While previous research focused on arranging questions to form dialog, we tackle the more challenging task of arranging questions from words, and dialog from questions. We develop our model in a simple "Guess which?" game scenario where the agent needs to predict an image region that has been selected by an oracle by asking questions to the oracle. As a result, the reinforcement learning agent arranges words to refer to the image features strategically to acquire the required information from the oracle, memorizing it and giving the correct prediction with an accuracy well above 80%. Imposing costs on the number of questions asked to the oracle leads to a strategy using few questions, while imposing costs on the number of words used leads to more but shorter questions. Our results are a step towards making goal-directed dialog fully generic by assembling it from words, elementary constituents of language.

Keywords: Visual Dialog Generation · Deep Reinforcement Learning · Compositionality.

1 Introduction

Deep neural networks have recently led to large progress in image- and natural language processing. Generating meaningful dialogs regarding visual content is required for various applications such as conversations with intelligent robot assistants [7]. One task allowing for research towards conversations is Visual Question Answering (VQA) [10], which consists of a single question-answer pair that is processed individually and, therefore, is limited in modeling complex communication. Visual dialog tasks [3] extend VQA and present sequences of questions, requiring a model to consider the question history and to recognize context. Visual dialogue generation tasks [7], in addition, require a goal-oriented agent to produce sequences of questions, focusing on efficient task completion.

* The authors acknowledge support from the German Research Foundation DFG under project Crossmodal Learning (TRR 169). Mengdi Li provided inspiration and feedback on the document.

Question Number	Agent's Actions	Oracle's Answer
1.	"green" + "?"	No
2.	"background" + "light" + "blue" + "?"	No
–	"stop"	–
Agent's Prediction	"2"	

Fig. 1. Left: Example image with the digit in position 2 selected by an oracle, as indicated by the red frame. This position needs to be predicted by the agent. Right: Task of the agent is to generate a suitable question sequence from individual actions, which express words and question marks, to receive answers by the oracle (whether the requested features are present in the selected sub-image; for example, question 1 asks whether it contains green) that allow it to correctly predict the selected position. The questions are created word by word, as the agent selects one new action each round. The agent ends its questions with the "stop" action, which precedes its prediction.

Examples of visual dialog generation tasks include the description of visual scenes based on question-answer pairs provided in a dataset [9] or the localization of objects in a visual context of the colored MNIST data set [1]. Our objective is to extend the Recurrent Attention Model [8], which has been used in various vision and robotics tasks, to enable the generation of questions using individual words. We will examine compositional questions, which have a stronger resemblance to natural speech patterns instead of a restricted set of questions. Furthermore, the development of an optimal questioning strategy and proper compositional question structure will be analyzed.

Our novel focus is on assembling dialog questions from individual words that refer to image features, thereby generating dialog in a more natural way. Challenges of this task include that the generated language needs to be goal-directed to ask relevant information, needs to be composed of tokens, and these tokens should refer to the visual features of the scene. In order to address those questions, we introduce a simple visually grounded language game (cf. Fig. 1), where the model predicts the location of a digit in the provided image based on a sequence of questions and corresponding answers about its visual attributes.

1.1 Task definition

The agent is trained on a variation of the MNIST-GuessNumber data set [1] that combines a set of grid images filled with digits and adds the corresponding descriptions for each specific image. The data set is modified to stimulate the generation of compositional questions in which individual tokens refer to distinct features of the images. To this end, the images can be referred to by color words, by attributes "light" and "dark", and by the specifiers "background" and "foreground", or a combination thereof.

One of the digit positions in the grid is selected by an oracle as the target digit. At each step, the agent produces a question from individual tokens and a

final end-of-question mark and receives the corresponding question-answer pair as input. They should be based on the attributes of the digits displayed on the currently presented image. Based on the history of question-answer pairs and the learned strategy, the agent selects a new token each round and updates the current question-answer pair. After the agent decides not to ask any further questions by producing a "stop" action, or after a maximum number of questions have been asked, the agent will make the digit prediction. At the end of the task, the agent will receive reward feedback from the environment.

2 Related Work

Deep learning has enabled tasks involving images and text, like image captioning and visual question answering. Visual dialog tasks focus on answering a final question based on visual references and comprehending the relations between a set of given questions and answers. The goal of the task proposed by Hongsuck et al. [6] is to answer a final question about digits' features shown in an image based on a given question & answer history. This model uses supervised learning to investigate *retrieved* attention for visual reference resolution and its combination with *tentative* attention to predict the correct response. These tasks, however, do not require any generation of questions.

2.1 Visual dialog generation

The aim of visual dialog generation tasks is to create dialog sequences of question-answer pairs, relating to images. Zhao et al. [1] utilize a modified MNIST data set to generate a grid of differently colored numbers. The task is to identify and locate a specific number on the generated image. The model consists of three networks, "guesser", "answerer" and "questioner", that are pre-trained using supervised learning. Thereafter, only the guesser network is trained using reinforcement learning to learn the correct classification. The vocabulary for the question generation consists of a limited set of attributes that describe each digit, therefore deviating from the natural process of communication.

Vries et al. [9] introduce a GuessWhat game, where two bots hold a conversation about a visual environment, which is represented by an image. One of the bots asks questions and receives binary answers to determine which object was selected. The data set was collected from a series of manually typed questions from a previously conducted study. The questions were later assigned to the corresponding images, therefore providing a static set of questions that the bots choose from.

Das et al. [7] extended on a goal-driven approach for the training of dialog agents. The main objective of the research task is to identify an image by asking questions about its content. The "questioner" and "answerer" bots are trained with supervised learning to ensure that they utilize a common language for communication. Then reinforcement learning is used to improve the performance. This reinforcement learning agent perceives the generated language not

as a supervised learning task, therefore attempting to simulate natural human speech. Moreover, the results of this experiment show that the strategy of using reinforcement learning on visual dialog tasks provides better performance than the ones based only on supervised learning [7].

2.2 Compositionality in VQA

Compositional questions are not entirely studied in the context of visual dialog tasks, but there are some novel approaches in the Visual Question Answering field [5, 17, 14, 15]. The following research papers concentrate on multi-hop reasoning. The point of question answering with compositional reasoning is to divide the question into different components and attempt to analyze them separately by considering how the rest of the components will behave [4].

Koushik et al. [5] research compositional reasoning for VQA. They argue that the majority of solutions [10, 13, 16] exploit statistical properties of the feature distribution to produce a correct answer. Therefore, the models rely on educated guesses instead of reasoning. For their experiments, the CLEVR data set [11] is used to answer corresponding questions about the displayed objects. This is achieved using reinforcement learning methods and a deep LSTM, and additional attention mechanisms in subsequent experiments. While VQA with the attention module yielded satisfactory results, reinforcement learning failed to improve the performance. Nevertheless, it was suggested that the reinforcement learning approach can be further incorporated into tasks with compositional reasoning [5].

3 Methodology

3.1 Data pre-processing

We generate a collection of colored 28×28 pixel images of MNIST digits arranged on a 2×2 grid[1]. This results in four possible positions that can be selected by the oracle as the target, which the agent has to infer. The target number corresponds to the positions in the grid from left to right line by line. The generated images include two primary colors {green, blue}. These colors describe both background and foreground, which can either appear in light or dark. If the background is light, the foreground must be dark, and vice versa, so that the digits are well visible. A rudimentary image preprocessing step extracts the image information and represents the image RGB colors of the background and foreground as a 3×8-dimensional vector.

The generated images allow for eight possible actions that the network can produce form the token set: {"background", "foreground", "dark", "light", "blue", "green", "?", "stop"}. The digits were excluded from the label set due to the lack of compositional combinations with other features. A question includes

[1] The dataset and the code of the model implementation are available at: https://github.com/ylysa/Recurrent-Attention-Model

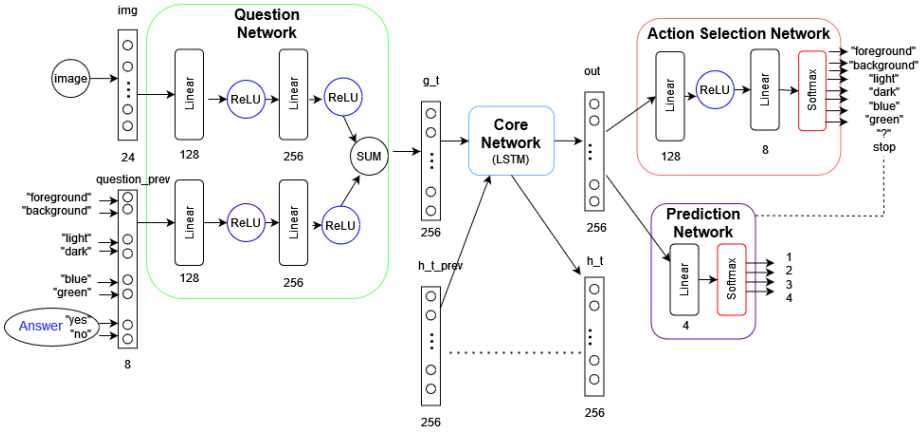


Fig. 2. The model architecture, including layer sizes and activation functions

between one and four actions, selecting one token at a time. For sequentially generating a sequence of tokens, the network receives the previously asked token as input until the "?" token is produced. The question mark concludes the question sequence generation and the model receives the corresponding answer from the oracle. The question sequence and the oracle's answer are combined into an 8-dimensional input vector consisting of two units for "background" or "foreground", two units for "light" or "dark", two units for each color, and two units for the binary answer (yes or no). Each unit is activated when the model has selected the corresponding action or the corresponding answer was received. During the sequence generation, the question token is fed back to the network. An answer to the question is only provided when the sequence is completed with the "?" token or when the limit of four actions has been reached. At the beginning of the next question, or if the generated question structure was invalid, the input vector is reset to a null vector. An answer is given only to valid questions.

For the question sequence, we define a simple grammar that requires a valid question structure that has to fulfill the following pattern: "background" or "foreground" feature followed by "light" or "dark" feature and the color. The order of the selected features must remain identical, but individual features can be omitted. The question must end with a "?" token to be considered valid. A question sequence that consists of only a question mark, without any other words, is invalid.

3.2 Architecture

The utilized model architecture is a modification of the Recurrent Attention Model proposed by Mnih et al. [8]. The agent is initially trained with supervised learning to predict the selected target digit's position and afterward with reinforcement learning to develop a questioning strategy. The agent consists of

four connected modules (cf. Fig. 2): *(i)* the Question Network processes visual information and the current question-answer vector; *(ii)* the Core Network is responsible for processing the concatenated input and integrating information over time, and produces a hidden state; *(iii)* the Action Selection Network outputs the actions to compose the current question; *(iv)* the Prediction Network outputs the final prediction about which position the oracle has selected.

The input consists of two vectors: the preprocessed image representation and the question-answer pair. These vectors are combined inside the Question Network, where each passes through a layer with 128 units and then they converge on a joint layer with 256 units. All layers use ReLU as an activation function.

The Core Network consists of a long short-term memory (LSTM) with 256 hidden units. In our experiments, an LSTM performed better than the simple recurrent network used in the original architecture [8]. At each time step, the LSTM receives the output of the Question Network as input, as well as its previous hidden state.

The Action Selection module consists of two layers: one hidden layer with 128 hidden units that receives its input from the Core Network and one output layer with eight outputs corresponding to the eight possible actions. The actions are sampled from a categorical probability distribution created from the output layer's Softmax function. For evaluation, the action with the highest probability is selected.

The Prediction Network has a single output layer, receiving its input directly from the Core Network. The last layer of the Prediction Network uses the Softmax activation function, which predicts the probability of each target. The four outputs correspond to four possible target positions.

3.3 Loss function

The model parameters are updated using the policy gradient method REINFORCE [12] and portray a partially observable Markov decision process (POMDP) since the underlying states cannot be fully observed (the oracle's selection is unknown). The Question, Prediction, Core, and Baseline Network parameters are trained with supervised loss. The Action Selection module parameters are trained with reinforcement learning, while the weights of the previously trained modules are frozen.

The prediction loss is calculated with negative log-likelihood, where y stands for the ground-truth label from the data set and $log(\tilde{y})$ for the probability distribution over all possible digits generated by the model:

$$L_{pred} = -log(\tilde{y}) \cdot y. \tag{1}$$

A baseline is estimated to approximate the reward function and is used to stabilize the reinforcement learning. The Baseline Network (not shown in Fig. 2) has a single hidden layer with 256 units that maps the output of the Core Network into one output. R_t is the accumulative reward over the entire trajectory and b_t

is the output of the Baseline Network. The baseline loss uses the mean squared error:

$$L_b = \frac{1}{T}\sum_{t=0}^{T}(R_t - b_t)^2. \tag{2}$$

The total reward provided to the model after an episode is $R = r_p + r_l$. The first term is the prediction reward, where the model receives $r_p = 1$ if the prediction was correct and $r_p = -1$ otherwise. The second term is the latency reward [2], calculated as

$$r_l = \frac{1}{T+2}, \tag{3}$$

where T stands for either the episode length or the number of questions in that episode. We conducted experiments for both definitions of the latency reward function. The reward R_t is assigned to an entire trajectory, i.e. dialog, since not a separate question but rather the entire trajectory has to be evaluated.

Empirical sampling over the state-action space in REINFORCE is expressed as $log\pi(a_t \mid s_t)$, the probability of selecting action a_t in state s_t. R_t is the accumulative reward over the whole trajectory and b_t is the output of the Baseline Network. The action loss is defined as

$$L_{act} = \sum_{t=0}^{T} -log\pi(a_t \mid s_t)(R_t - b_t). \tag{4}$$

The baseline and action losses are backpropagated only to the Baseline module and Action Selection module, respectively, but not to the remaining modules. The total loss is the sum of all three components, with equal weights for each component

$$L = L_{pred} + L_b + L_{act}. \tag{5}$$

4 Experiments

The generated images include two colors: green, blue. The four sub-images in the grid are likely to contain identical colors and therefore require complex questions. Overall there exist 1680 unique images; in each image, we ensure that there are no two equal sub-images. This allows to uniquely identify the selected image via a suitable question strategy. Additionally, there are four times as many data samples, since each image consists of four different sub-images that can be selected by the oracle. Only a minority of 300 of the images are used for training, which poses challenges for generalization. The validation data set contains 1000 images, which do not overlap with the training set. The validation set is also used for testing because no hyperparameters are optimized using the validation set. To prevent the predictions from overfitting to the training images regardless of the question sequence, the target sub-image is always selected randomly for each epoch.

4.1 Pre-training

Learning to predict correctly requires question-answer histories that contain sufficient information, while learning the question strategies suffers from noisy rewards resulting from unreliable target predictions. This causes difficulties for the model when learning the predictions and question sequences simultaneously. Therefore, only the Prediction Network is pre-trained using an automatically generated question sequence for each image. During each epoch a different random question sequence is created, to present the model with a large variety of environment states. After the supervised pre-training for less than 20 epochs, the model reaches an accuracy above 90%. After pre-training of the Prediction Network, the Action Selection network is trained using reinforcement learning to learn a correct question generation starting with randomly initialized weights. The model is optimized with Adam Optimizer with a learning rate of 3e-4. Sequences are terminated by the stop action, or after a maximum of 13 time steps, including the initial step with a null vector input. The prediction is derived from the last available state.

4.2 Experiments with different time efficiency losses

For evaluation of the model, we test multiple implementations of time efficiency constraints, which contribute to the action loss function, and their effect on the resulting behavior.

1. The latency reward defined in Eq. 3 uses the number of selected tokens as the number of required time steps T. The reward function that depends on the number of tokens is referred to as R^{tok}.
2. Time steps T in Eq. 3 are defined as the number of questions posed to the oracle. The reward function that depends on the number of questions is referred to as R^{qu}.
3. The latency reward is applied only if the network predicts correctly, in which case $R = 1+r_l$, while for false predictions the total reward remains as $R = -1$ regardless of the number of time steps. Here, similar to the second definition of the reward function, T is the number of questions. The corresponding reward function is referred to as R^{qu+lat}.

5 Results

5.1 Accuracy

The model achieves an accuracy above 80% on the validation data (see Fig. 3), where the form of the time efficiency loss does not have a major impact. The model with the R^{qu+lat} loss had the highest accuracy by a small margin at the end. These results demonstrate the generalization capabilities given that only a minority of 300 images from 1680 possible images, were used for training, and utilizing unseen validation images.

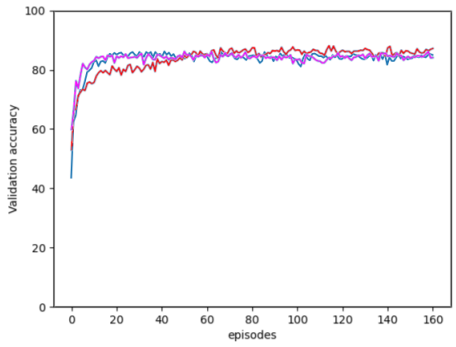


Fig. 3. Validation accuracy with different reward functions (blue: R^{tok}, purple: R^{qu}, red: R^{qu+lat})

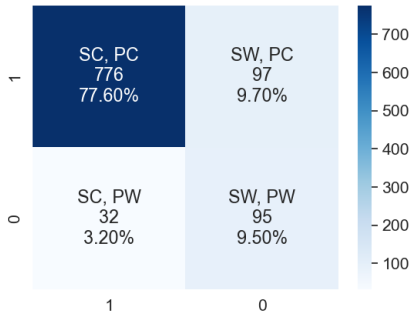


Fig. 4. Test accuracy of R^{q+lat} (SC: correct question sequence, SW: wrong question sequence, PC: correct prediction, PW: wrong prediction)

For analyzing the models' performance, the generated question sequence and resulting prediction have to be considered. Specifically, if the sequence of questions is considered correct (SC), meaning that it retrieves sufficient information from the oracle for a correct prediction (PC) of the chosen target sub-image. Similarly, the question history can be considered wrong (SW), not asking for sufficient information, possibly resulting in a wrong prediction (PW). The model with the R^{q+lat} reward function provides the best performance in these four criteria and is illustrated in Fig. 4. For most images, the model generates a valid sequence of questions (SC) and the correct prediction (PC). In 9.7% of cases (SW, PC), a correct prediction was partially guessed since an improper question sequence would not guarantee a correct prediction. In 9.5% of cases (SW, PW), the model predicts wrongly given an improper question sequence. In the remaining 3.2% of cases (SC, PW), the model should be capable of predicting the correct sub-image based on the question sequence, however, the Prediction Network predicts the wrong sub-image.

5.2 Question evaluation

As can be seen in Fig. 5, the average length of the questions for the model with R^{qu} and R^{qu+lat} reward functions is greater in comparison to the model with R^{tok}. The corresponding reward function stimulates longer compositional questions. The majority of questions vary between two and three tokens. For the R^{qu+lat} model, the distribution of question length leans toward three tokens and the number of four tokens questions is the highest among all experiments, which demonstrates the ability of this approach to generate compositional questions.

The average number of valid questions remains under four (see Fig. 6). In the majority of cases, where the sequence is longer than three questions, the

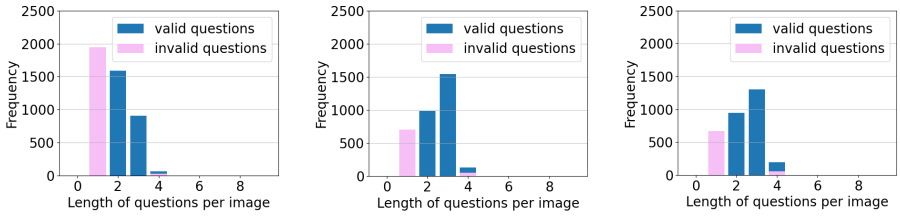


Fig. 5. Average *length* of questions (i.e. number of tokens per question) with different reward functions. Left: R^{tok}; Middle: R^{qu}; Right: R^{qu+lat}. Questions of length 1 contain only the "?" token and are invalid.

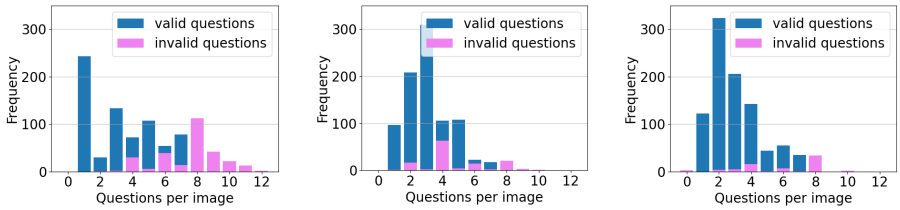


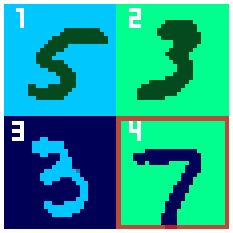
Fig. 6. Average *number* of questions per image with different reward functions. Left: R^{tok}; Middle: R^{qu}; Right: R^{qu+lat}.

remaining questions consist only of question marks. The invalid questions that include some tokens usually do not have the question mark accompanying them.

Upon inspection of the models' generated output, questions often contain a learned bias, such as preferring "dark" over "light", or "background" over "foreground". However, we did not notice any bias for color. These biases do not necessarily degrade performance, since these features are inherently symmetric in the data. It is equally efficient to inquire about a dark background or light background since the same number of digit sub-images is eliminated. The model often does not learn all possible question sequences. The sequences from the R^{qu} and R^{qu+lat} reward functions usually do not include any repetitions beyond the question mark, while we did observe models trained with R^{tok} to repeat questions. We also observed the model to be inefficient in applying the optimal question strategy, when it chose a color that is present only in a single sub-image in its first question, since such a question cannot rule out multiple sub-images.

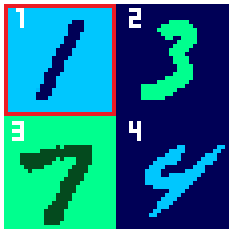
In typical dialog sequences (see Fig. 7), the questions appear sequentially and with correct structure, the repetitions are avoided and the stop action is selected accordingly. Moreover, the inquired features are all present in the image. The models avoid irrelevant questions and derive a correct prediction based on the question history and the input image. The R^{qu+lat} reward function provides the advantage of generating more complex compositional questions and the number of questions is shorter than those derived by the R^{tok} and R^{qu} reward functions.

a)



Question Number	Agent's Actions	Oracle's Answer
1.	"green" + "?"	Yes
2.	"light" + "blue" + "?"	No
3.	"blue" + "?"	Yes
–	"stop"	–
Agent's Prediction	"4"	

b)



Question Number	Agent's Actions	Oracle's Answer
1.	"light" + "blue" + "?"	Yes
2.	"background" + "light" + "blue" + "?"	Yes
–	"stop"	–
Agent's Prediction	"1"	

Fig. 7. Two example tasks being solved by the agent. The model in **a)** has been penalized via the number of tokens (R^{tok}), the model in **b)** via the number of questions (R^{qu+lat}) which leads to fewer but longer questions. The display is as in Fig. 1: left, the example image with the oracle's choice in red; right, the generated dialog.

6 Conclusion

In summary, the Recurrent Attention Model is able to generate questions compositionally from tokens and arrange questions strategically to generate goal-directed visual dialog. The model produces a correct question structure and chooses question tokens sequentially while avoiding repetitions and reducing the total length of the sequence. Further, the model refers to the presented visual content by selecting relevant question sequences for novel image examples that were not shown during the training process. The results show that the learned strategy of the model heavily varies depending on the time efficiency losses used in the reward function. Penalizing the number of questions, instead of the number of words, results in a model that generates fewer questions, but also longer compositional questions. Hence, a suitable reward function may allow generating language of desirable characteristics.

Certain limitations are that the model does not necessarily yield optimal results, often producing correct but less efficient sequences. Future work may involve reinforcement learning algorithms that are more robust to noisy rewards, which promise also to overcome the need for supervised pretraining. Further work may address more complex scenarios and forming longer sentences with grammatical variations. Our model handles only minimal grammar with a small number of words, owed to the sole use of reinforcement learning for word generation. To exceed those limitations, an unsupervised language model [18] could be

used as an element of model-based reinforcement learning, to efficiently generate frequent word sequences for the generation of more complex sentences.

References

1. Zhao R., Tresp V.: Efficient Dialog Policy Learning via Positive Memory Retention. Spoken Language Technology Workshop (SLT). IEEE, 2018, pp. 868–875.
2. Li M., Kerzel M., Zeng Z., Wermter S., Weber C., Lee J. H., Liu Z.: Robotic Occlusion Reasoning for Efficient Object Existence Prediction. IROS, 2021.
3. Das A., Kottur S., Gupta K., Singh A., Yadav D., Moura J.M., Parikh D., Batra D.: Visual Dialog. IEEE Computer Vision and Pattern Recognition. CVPR, 2017.
4. Giannakopoulou D., Namjoshi K.S., Păsăreanu C.S.: Compositional Reasoning. Handbook of Model Checking, pp. 345–383, Springer International Pub. (2018).
5. Koushik J., Hayashi H., Sachan D. S.: Compositional Reasoning for Visual Question Answering. ICML, 2017.
6. Hongsuck S. P., Lehrmann A., Han B., Sigal L.: Visual Reference Resolution using Attention Memory for Visual Dialog. Advances in Neural Information Processing Systems. NIPS, 2017, pp. 3719–3729.
7. Das A., Kottur S., Gupta K., Singh A., Yadav D., Moura J.M., Parikh D., Batra D.: Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. IEEE International Conference on Computer Vision. ICCV, 2017, pp. 2970-2979.
8. Mnih V., Heess N., Graves A., Kavukcuoglu K.: Recurrent Models of Visual Attention. Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS). Volume 2, 2014, pp. 2204–2212.
9. de Vries H., Strub F., Chandar S., Pietquin O., Larochelle H., Courville A.: GuessWhat?! Visual object discovery through multi-modal dialogue. IEEE Conference on Computer Vision and Pattern Recognition. CVPR, 2017, pp. 4466-4475.
10. Agrawal A., Lu J., Antol S., Mitchell M., Zitnick C. L., Batra D., Parikh, D.: VQA: Visual Question Answering. International Journal of Computer Vision, Volume 123, 2017, pp. 4–31.
11. Johnson J., Hariharan B., van der Maaten L., Fei-Fei L., Zitnick C. L., Girshick R.: CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. CVPR, 2017, pp. 1988-1997.
12. Williams R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning 8, 1992, pp. 229–256.
13. Goyal Y., Khot T., Summers-Stay D., Batra D., Parikh D.: Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. IEEE Conference on Computer Vision and Pattern Recognition. CVPR, 2017.
14. Hudson D. A., Manning C. D.: GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, 2019, pp. 6693-6702.
15. Andreas J., Rohrbach M., Darrell T., Klein D.: Deep compositional question answering with neural module networks. IEEE Conference on Computer Vision and Pattern Recognition. CVPR, 2016.
16. Subramanian S., Singh S., Gardner M.: Analyzing Compositionality of Visual Question Answering. ViGIL@NeurIPS, 2019.
17. Agrawal A., Kembhavi A., Batra D., Parikh D.: C-VQA: A Compositional Split of the Visual Question Answering (VQA) v1.0 Dataset. CoRR, 2017.
18. Radford A., Jozefowicz R., Sutskever I. Learning to Generate Reviews and Discovering Sentiment. arXiv:1704.01444, 2017.