Good, Better, Best: Multi-Choice VQA with Textual Distractors Generation via Policy Gradient

Jiaying Lu^{1, 3}, Xin Ye¹, Yi Ren², Yezhou Yang¹

¹School of Computing, Informatics, and Decision Systems Engineering, Arizona State University

²School for the Engineering of Matter, Transport and Energy, Arizona State University

³Department of Computer Science, Emory University

jiaying.lu@emory.edu, {xinye1, yiren, yz.yang}@asu.edu

Abstract

An increasing amount of studies have investigated the decision-making process of VQA models. Many of these studies focus on the reason behind the correct answer chosen by a model. Yet, the reason why the distracting answer chose by a model has rarely been studied. To this end, we introduce a novel task called textual Distractors Generation for VQA (DG-VQA) that explaining the decision boundaries of existing VQA models. The goal of DG-VQA is to generate the most confusing set of textual distractors in multi-choice VOA tasks which expose the vulnerability of existing models (i.e. to generate distractors that lure existing model to fail). We show that DG-VQA can be formulated as a Markov Decision Process, and present a reinforcement learning solution to come up with distractors in an unsupervised manner. The solution addresses the lack of large annotated corpus issue in previous distractor generation methods. Our proposed model receives reward signals from fully-trained multi-choice VQA models and updates its parameters via policy gradient. The empirical results show that the generated textual distractors can successfully attack several popular VQA models with an average 20% accuracy drop from 64%. Furthermore, we conduct adversarial training to improve the robustness of VQA models by incorporating the generated distractors. Empirical results validate the effectiveness of adversarial training by showing a performance improvement of 27% for the multichoice VQA task.

1 Introduction

With advancements in deep learning technologies, neural network based models are increasingly applied in different areas. Unfortunately, neural network models are often regarded as black-box model that are vulnerable to potential attacks (Goodfellow, Shlens, and Szegedy 2015; Yao et al. 2019; Gokhale et al. 2020) and lack of interpretability (Fang et al. 2019; Zhang, Niebles, and Soto 2019). Although thsee models have achieved remarkable performance in benchmark datasets, robustness and explainability are essential requirements for many real-life applications (Marino, Wickramasinghe, and Manic 2018; Gunning and Aha 2019).

In this paper, we are particularly interested in the problem of multiple-choice visual question answering (Zhu et al. 2016; Jabri, Joulin, and van der Maaten 2016), which is one type of Visual Question Answring (VQA) (Antol et al.



(a) Input Image and Question

A: More windows 🗸

A: The passing scenery

A: Snow falling

A: The backyard patio

(b) Original Answer Choices

A: More windows

A: A mirror
A: **Lights** X

A: A laptop

7..7.1aptop

(c) Generated Distractors

Figure 1: An example of DG-VQA task. The well-trained VQA model predicts the right answer choice for the input image and question (1a and 1b). However, it is easy to distinguish correct choice from distractor choices. The model will be fooled when encountering generated distractors (1c).

2015). The goal of multiple-choice VQA is to choose one choice given an image, a free-form textual question as context, and several (typically four) textual choices as answer candidates. Recently, the artificial intelligence community has achieved remarkable progress to bridge the gap between human performance and state-of-the-art neural network model on this task. However, it has been pointed out that the distracting choices are too simple or biased (Jabri, Joulin, and van der Maaten 2016) in the current benchmark datasets, which raises doubt about the trained models' true discriminative ability (see Figure 1).

To facilitate multiple-choice VQA better to serve as a robust"visual Turing test" (Turing 2009; Geman et al. 2015) and to explore factors which cause the failure of existing VQA models, we introduce a novel task as generating chal-

lenging distractors, dubbed as DG-VQA: textual Distractor Generation for VQA. The task definition is as follows: Given an image-question-answer triplet, generating distracting choices that lead neural models failing at choosing correct answer choice. Generating such distractors provides a tool for researchers to figure out whether well-trained VQA models are vulnerable to potential attacks and determine whether they are ready for real-world deployment. Moreover, multiple-choice question answering are widely used in the education area, and distractor generation is a crucial and time-consuming procedure. There are some previous works (Liang et al. 2018; Gao et al. 2018) focusing on automatic distractor generation (DG) to alleviate instructors' workload. But none of them involve in visual question domain considering there is a trend to apply multimodal materials on student learning. Last but not the least, DG-VQA provide explainability of the decision boundary of exisiting VQA models that is readable for humans and can further guide more robust models. For evaluation measurement of generated distractor, we put forward a novel metric by measuring the performance degradation of the VQA models.

The major technical challenge for DG-VQA is that there are few training data available. Manually craft such dataset is time-consuming and may contain biases and errors. Owing to the recent progress of deep learning, state-of-art VQA models' performance is close to humans. Therefore, we propose to utilize existing VQA models as discriminator to train an distractor generator, which is inspired by the GAN framework (Goodfellow et al. 2014). We propose an reinforcement learning (RL) setting (Rennie et al. 2017; Li and Ye 2018) to propagate the non-differentiable confidence score output by the discriminator. In this setting, the distractor generator is regarded as a policy agent which receiving rewards produced by well-trained VQA model of input image, question, answer choice triplets. The policy gradient algorithm is used to optimize the generator while the discriminator is fixed during the whole training phase.

The contribution of this paper is three-fold:

- We introduce a new DG-VQA task for vision and language understanding research, accompanied with a practical metric for evaluating the quality of the generated distractors. The DG-VQA task also provides a novel perspective for the interpretability of the exisiting neural models' decision boundary.
- 2. We propose a novel perspective to formulate DG-VQA task as a reinforcement learning task and optimize it with policy gradient. The proposed RL model addresses the lack of training data issue.
- 3. We present and show that by incorporating the challenging distractors one can train a more robust VQA model.

2 Related Work

Distractor Generation. Automatic distractors generation (DG) from text is explored in-depth in the Natural Language Processing domain. At the same time, there are only few studies in the visual questions domain. Most prior approaches to DG are based on unsupervised similarity measures. These include n-gram co-occurrence likelihood (Hill

and Simha 2016), word/sentence embedding-based semantic similarities (Kumar, Banchs, and D'Haro 2015), syntactic homogeneity (Chen, Liou, and Chang 2006) and ontology-based similarity (Stasaski and Hearst 2017). Besides, other works utilize supervised learning algorithms for DG. Sakaguchi et al. (2013) train a discriminative model to predict distractors, Liang et al. (2018) apply learning to rank algorithm, and Gao et al. (2018) use an end-to-end framework to produce distractors generatively. Although being successful, multimodality knowledge is still required to produce high-quality distractors.

Reinforcement Learning. Reinforcement learning (Sutton and Barto 2018) has been adopted in a variety of vision and language tasks, such as image captioning (Rennie et al. 2017), text to image synthesis (Reed et al. 2016), VQA (Liu et al. 2018; Fan et al. 2018) and visual dialogue (Zhang et al. 2017). Liu et al. (2018) propose a reinforcement learning based strategy to generate visual questions. Fan et al. (2018) enhance content and linguistic attributes of produced questions by introducing two discriminators in an RL framework.



Figure 2: The MLP Reinforce Framework

3 The Approach

Figure 1 displays an example of the DG-VQA task. Formally, given an image i, a natural language question q and four corresponding multi-choice answers as, which include one correct answer $a^{correct}$ and three wrong answers a^{wrong} s in original dataset, the task is to produce three plausible but incorrect distractors ds. A DG-VQA is learned so that its generated distractors maximize the expected accuracy drop on a given multi-choices VQA model, where accuracy is measured as the percentage of times the model picks up the correct choice.

3.1 DG-VQA as an RL Problem

Inspired by recent advance in reinforcement learning and adversarial training (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Yao et al. 2019), RL methods are ideal approaches to address lacking large scale annotated training data and inconsistency between the training objective and test metrics. So we adopt a policy gradient framework to generate textual distractor (adversarial example) for visual multiple-choice questions. The framework has two major components: an environment model J_{ϕ} which is a well-trained VQA model, and an agent model G_{θ} learns to confuse J_{ϕ} by generating high quality distractors d_{θ} . Here we utilize a well-trained model as the discriminator rather than train a model from

scratch. The reason behind it is the concern of local convergence (Mescheder, Nowozin, and Geiger 2018). We put our approach under a semi white-box attack setting where G_{θ} can receive feedback signals regarding selected choices from J_{ϕ} , but can not access J_{ϕ} parameters or gradients. From the RL perspective, the well-trained VQA model J_{ϕ} serves as the environment, and the generative model G_{θ} is the policy agent.

We first denote distractors generation as a sequence generation process. The generative model G_{θ} is trained to produce a sequence $y_{1:T} = (y_1, y_2, ..., y_t, ..., y_T)$, where y_t is one word in the vocabulary of all candidate tokens. At each timestep t, G_{θ} is given an (image, question, answer sequence until last timestep) triple $(i, q, y_{1:t-1})$. Since G_{θ} outputs a probability distribution over each token in produced sequence, we can use decoding algorithms like greedy search or beam search to locate the top-3 distractors. Under the reinforcement learning setting, at timestep t the state s of the answer component is the current producedly tokens $y_{1:t-1}$ and the action a is the next token y_t to produce. So the state transition is deterministic once an action has been chosen. Following the notation in (Sutton and Barto 2018), the object of the G_{θ} is to produce a sequence to minimize its negative expected reward:

$$L(\theta) = -\mathbb{E}_{y_{1:T} \sim G_{\theta}}[R(y_{1:T})],\tag{1}$$

where $y_{1:T}$ is the distractor sampled from the model G_{θ} . Without the loss of generality, we adopt the REINFORCE algorithm (Williams 1992) to compute the policy gradient and take the predicted likelihood score of being true by the discriminator $J_{\phi}(i,q,y_{1:T})$ as the reward.

$$y_{1:T} = G_{\theta}(i, q; y_{1:T-1})$$

$$R(y_{1:T}) = J_{\phi}(i, q, y_{1:T})$$

$$\nabla_{\theta} L(\theta) = -\mathbb{E}_{y_{1:T} \sim G_{\theta}} [R(y_{1:T}) \nabla_{\theta} log G_{\theta}(y_{1:T})].$$
(2)

It is worth mentioning that the discriminator can only output a reward value from a completed sequence. However, in DG-VQA setting and under sequence generation scenario, the model should consider the long-term reward at every timestep. To tackle this challenge, researchers typically use Monte Carlo search to sample the unknown last T-t tokens. For simplicity, we generate the final distractor sequence d for one time step by selecting the output distractor over a distractors pool, and empirical results show that it is already effective. Thus, the distractor generator can be formulated as follows:

$$d = y_{1:T} = G_{\theta}(i, q). \tag{3}$$

The framework is dubbed as *MLP Reinforce* (MLPR) since we adopt a MLP architecture as the agent in a Reinforcement learning setting. Equation 4 defines the return of sampled distractor answer choices. In general, we take the output of the well-trained multi-choice VQA model as the reward. Furthermore, we punish the distractor d which is semantically equivalent to the correct answer $a^{correct}$ for the given context. The semantic similarity model is trained in the BERT (Devlin et al. 2018) architecture. We put details

about the similarity model in Section 4.2.

$$R(d) = \begin{cases} -1 & if \ IsSemEquiv(d, a^{correct}); \\ J_{\phi}(i, q, d) & otherwise. \end{cases}$$
(4)

In practice, the expected gradient can be approximated using several distractors d^s sampled from G_θ for each input image, question and correct answer triplet in a minibatch.

$$\nabla_{\theta} L(\theta) \approx \sum_{s} -R(d^{s}) \nabla_{\theta} log G_{\theta}(d^{s}).$$
 (5)

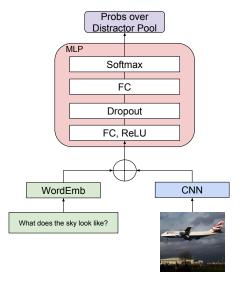


Figure 3: Multiple Layer Architecture for DG-VQA.

3.2 The Agent: Multiple-Layer Perceptron

The architecture of our generative model G_{θ} is a multiple-layer perceptron. It is widely used in many domain and has shown success in visual question answering by Jabri's work (Jabri, Joulin, and van der Maaten 2016). Figure 3 depicts the layers of the MLP model. Specifically, the questions are represented by averaging word2vec (Mikolov et al. 2013) embedding over all tokens. The images are represented using features computed by a pre-trained deep CNN image encoder. Unless otherwise stated, we use the penultimate layer of Resnet-101 (He et al. 2016) as the extracted features. Then, the multi-modality feature sets are concatenated and used to train a classification model that predicts the corresponding distractor label.

A multilayer perceptron (MLP) is adopted as the classification model trained on the concatenated features: 1) The word2vec embedding (300-dimensional), and 2) the image features (2048-dimensional). By default, the MLP has 4,096 hidden units unless otherwise specified. We denote the image and question features as x_i and x_q , respectively. By denoting the concatenation as $c=x_i\oplus x_q$, we formulate the models as follows:

$$z = (w_2 ReLu(w_1c + b_1) + b_2).$$
(6)

The MLP outputs a distribution over the distractor pool using the softmax function. Then, the system selects the distractors ds with the top-3 highest likelihood from $G_{\theta}(i, q)$.

$$P(d|i,q) = softmax(z). (7)$$

3.3 The Environment: VQA Models

Any multi-choice VQA model which produces a likelihood score of answers for given visual questions can serve as the environment in the MLPR framework. Here, we conduct attacks on three popular VQA models:

TellingVQA (2016) is a recurrent QA model with spatial attention. It first encodes the image through a pretrain VGG-16 model (Simonyan and Zisserman 2015). Then it uses a one-layer LSTM to read the image encoding and all the question tokens. It continue feed the answer choice tokens into LSTM, and would finally produce the likelihood score. **RevisitedVQA** (2016) proposes a quite simple architecture for VQA multiple choice task. RevisitedVQA receives an image-question-answer triplet, encodes it and utilizes a MLP to compute whether or not the triplet is correct.

MCB (2016) proposes a novel method called Multimodel Compact Biliniear pooling to efficiently and expressively combine language and vision features.

The MLPR can also support generating distractors over a bundle of well-trained models together as the environment by providing a mixed reward. It is worth noting that the proposed distractor generation method is not restricted to these models, but generally applicable to any VQA models which can produce confidence scores regarding the answers they choose.

4 Experimental Results

4.1 Datasets and Evaluation Metrics

We evaluate our model on **Visual7w** (Zhu et al. 2016), which is a public multiple-choice visual question answering dataset. **Visual7w** consists of 47,300 images from COCO and 327,939 multiple-choice QA pairs collected on Amazon Mechanical Turk.

As mentioned in Section 1, traditional metrics for DG such as reliability and validity highly rely on manual evaluations. Inspired by adversarial attack evaluation, we define the ability of generated distractors to fool well-trained VQA models as the metric, denoted as $\Delta Acc.\ \Delta Acc$ is the difference between VQA model's performance on the original distractors and on the generated distractors. The learning objective of proposed method is directly related to this metric and thus can eliminate the mismatch between training goal and test measurement. ΔAcc is an automatic metric. The higher ΔAcc is, the better generated distractors are.

4.2 Baselines

We implemente the following baselines:

Per Q-type prior: We select 3 most popular answers per question type as distractors.

Adversarial Matching: Following (Zellers et al. 2018), We choose the question and correct answer pair as one positive sample, and pick up the same specific question and some 5

answers from the whole answer pool as corresponding negative samples.

LSTM Q+I: We keep the fine-tuned VGG and two-layer LSTM in (Antol et al. 2015), while only change the training targets from correct choices to incorrect ones. These incorrect choices are failed predictions from well-trained VQA models

As we can see, Per Q-type prior is a heuristic method for distractor generation. Adversarial Matching leverages external knowledge to measure the two major characteristics of high quality distractors. LSTM Q+I further takes visual and textual clues. These three baselines tackle DG-VQA from different perspetives.

4.3 Experimental Settings

We adopt a two-channel vision and language neural network that outputs probabilities over K candidate distractors as the agent. We set the candidate distractor frequency threshold to 20, to filter the candidate pool size K to 1516, which covers 2% of all training and validation choices. The questions are represented by 300-dim averaged word embeddings from a pre-trained fastText (Bojanowski et al. 2017) model. We use all words in the training and validation dataset to train the embedding. In the experiment, we set dropout to 0.5 in each hidden layer with a ReLU activation. We train the MLP for 200 epochs, which is determined emprically.

For the environment, we adopt the best RevisitedVQA model in (Jabri, Joulin, and van der Maaten 2016). The well-trained RevisitedVQA model outperforms other state-of-the-art models which are mentioned in 3.3, and it achieves 65.8% accuracy on the Visual7W dataset. We evaluate the propose MLP Reinforce model with two ablated versions:

MLPR: Here, model parameters are updated only through policy gradient. We train the model with the rewards from the well-trained RevisitedVQA (2016) environments.

MLPR+ Pre-train: Reinforce algorithm is known to have large variance. Inspired by the concept of Imitation Learning (2016) and Teacher Forcing (2015), we first pre-train the MLP model with correct answer choice using cross-entropy loss for a small size (80) epochs. The pre-train process is to prevent generating unstable results. Then we trained the model as described before. An interpretation of adopting this practical training strategy is by an analogy with the undercover police: being integrated and then attack.

4.4 Experimental Results

Table 1 list the attacking results of the generated distractors for Visual7W Dataset. since the three defending models use different architectures, the distractor generation model requires high generalization capability to confuse all three of them. Baseline models yield poor overall ΔAcc . Per Qtype prior fails to fool any defenders. Adversarial matching and LSTM Q+I are lack of generalization capability, which are only able to make one defender's accuracy flipping. Our proposed MLPR methods yield statistically significant improvements on both defenders. Note that MLPR performs better on RevisitedVQA and MCB than MLPR+Pretrain, while worse on TellingVQA(ΔAcc -30.9%). It indicates

Model	TellingVQA (2016)		RevisitedVQA (2016)		MCB (2016)			
Wodel	Acc	ΔAcc	Acc	ΔAcc	Acc	ΔAcc		
Original	55.6%	-	64.8%	-	62.2%	-		
Baselines								
per Q-type prior	57.3%	-1.7%	68.7%	-3.9%	85.7%	-23.5%		
Adversarial Matching(2018)	54.7%	0.9%	71.7%	-6.9%	51.3%	10.9%		
LSTM Q+I(2015)	41.7%	13.9%	68.9%	-4.1%	85.7%	-23.5%		
Proposed Methods								
Reward by RevisitedVQA								
- MLPR	86.5%	-30.9%	0.01%	64.7%	26.5%	35.7%		
- MLPR + Pre-train	<u>33.7%</u>	21.9%	49.1%	15.8%	37.5%	24.7%		

Table 1: Attack Results for Visual7W Dataset. Human accuracy on the task is 96.0%. Higher ΔAcc values are better.

that without pre-train MLPR is vulnerable to overfitting, although it is impressive that MLPR successfully confuses RevisitedVQA in almost every question(Acc~0.01%)! Table 2 provides a case study (in Section 4.6) of the distractors with the given context extracted by our models.

Furthermore, the attacking performance boost from baseline models to MLPR is considerable according to automatic metrics and case study. This shows that feedbacks from well-trained VQA models are beneficial to distractor generation. Only receiving rewards from one specific environment, MLPR + Pre-train method produces generalized distractors to fool all three defenders. We speculate that the pre-train process in this case provides a better beginning probability distribution over distractors. Therefore, it prevents the agent from falling into the local minima trap.

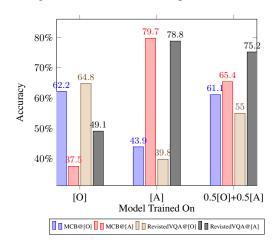


Figure 4: Data Augmentation Results

4.5 Augmenting VQA with DG-VQA

We adopt DG-VQA for training more robust VQA models as a data augmentation process. To validate it, an extra data augmentation experiment is conducted, where we keep the correct alternative of each question and swap the original incorrect choices to the generated distractors. MCB and RevisitedVQA are better suited for this setting since they take both correct and incorrect alternatives into consideration while training, while TellingVQA only takes the correct answer as input. We re-train the two VQA multiple-choice

models on two settings: 1) adversarial examples alone, 2) the union of these examples and the original training data. As a control group, we adopt the VQA models which are trained on the original data alone. Here, adversarial examples are produced by our MLPR + Pre-train method, which has been shown the most effective.

Figure 4 reports the results, where x-axis indicates on which dataset the models are trained. [O] and [A] refer to the original data and the augmented data respectively. And 0.5[O]+0.5[A] denotes 50% of all questions' incorrect alternatives are replaced by the generated distractors. Different bars indicate the VQA accuracy that models can achieve on a specific dataset. At first glance, we find that data augmentation training improve the models' performance on adversarial examples. However, it hurts the performance on the original test data. Models trained on the union of augmented and original data achieves the best performance with minimal Acc@[O] drop of 1.1% and highest Acc@[A] improvement by 27.9%. These results demonstrate the effectiveness of DG-VQA for training more robust VQA models.

4.6 Case Study

We collect sample adversarial distractor choices generated by baseline and the proposed methods. Table 2 showcases these samples and their corresponding images, questions and original choices. Oberseving the predictions made by the defender (RevisitedVQA model), our proposed methods, MLPR + Pretrain, exhibits an ability to learn certain interpretable and practical policies generating adversarial and distracting alternatives:

Semantic Similarity: It is not surprising the proposed distractors generator learned the strategy to take advantage of semantically similar tokens to the correct answer. Human beings follow the same strategy when they try to come up with distracting alternatives. As we can see, distractors generated by our method and the correct answer almost belong to the same general concept category. For example, "baseball", "soccer", "tennis" and "golf" are sport terms. And all distractors produced for the question "how many black cows are there" are all numbers, which belong to the same category of the correct answer: "3".

Context Matters: The other critical factor is the context. In the first column of Table 2, both distractors of the original dataset and of augmented ones are adjectives to describe the weather. However, "cloudy" is a better distractor to "stormy"



Q:What does the sky look like?



Q:How many black cows are there?



Q:What sport are they playing?



Q:Why is there a piece missing?



Q:What two colors are in the flag directly above the cats head?

				the cats head?
Original Choices				
A: Stormy √	A: 3 ✓	A: Golf	A: Someone ate some	A: Green and yellow
A: Hazy	A: 9	A: Baseball 🗡	A: It was removed	A: Blue and white X
A: Windy	A: 8	A: Hockey	A: It was put somewhere else X	A: Black and red
A: Sunny	A: 7	A: Basketball	A: Someone took it	A: Green and black
Distractors by Adversa	rial Matching			
A: Stormy	A: 3	A: Golf ✓	A: Someone ate some	A: Green and yellow
A: Sky 🗡	A: Zero	A: Volleyball	A: Wood	A: Blue and black
A: Blue	A: 5	A: Playing soccer	A: Glass 🗡	A: Blue and red
A: Cloudy	A: 0 ×	A: Soccer	A: To rest	A: Blue and white X
Distractors by MLPR				
A: Stormy	A: 3	A: Golf	A: Someone ate some	A: Green and yellow
A: Shadows	A: Shadows 🗡	A: Shadows	A: Shadows	A: Shadows 🗡
A: Daylight	A: During daylight	A: During daylight 🗡	A: Daylight 🗡	A: Daylight
A: Shadow X	A: In the daytime	A: Daylight	A: During daylight	A: In the daytime
Distractors by MLPR +	+ Pretrain			
A: Stormy	A: 3	A: Golf	A: Someone ate some	A: Green and yellow ✓
A: Cloudy X	A: Two	A: Baseball 🗡	A: To eat 🗡	A: Blue
A: Blue	A: Four	A: Soccer	A: To cook	A: Legs
A: Clouds	A: One X	A: Tennis	A: For display	A: Orange

Table 2: Excerpts from sampled original and adversarial generated distractor choices. Green choices are correct answers. Bold texts indicate options chosen by the released RevisitedVQA model in Visual7W.

if we take a look at the image, compared to "hazy", "windy" and "sunny". Under the original choice setting, the defender is able to select the correct answer. But once encountering with the generated distractors, it is confused and misleadingly pick "cloudy" as the answer. Tackling vision and language tasks needs multimodal cognitive ability. In the DG-VQA task, a system should comprehensively utilize information from both the given questions and the images.

Attack the Weaknesses and Improve: Our architecture is able receive feedback from the defender. It is a common sense to exploit opponents' weaknesses to defeat them. By analyzing confidence scores of the alternatives, the distractor generator identifies the differences between the hard and the easy ones. Examples of this can be found in distractors generated by MLPR (See Table 2). It seems our system generates easy-to-human distractors like "shadows" or "daylight", the defender is observed to be defeated in fact. Intuitively speaking, the fatal drawback could be attributed to the trained model's overly biased objective function. Our method is able to identify and exploits them. Further, by considering these weakness for the next round of training, a model's robustness is improved. The above case studies support that our method in fact outputs more challenging distractors by considering all together the semantics of correct answer, the information of the context, and the feedback

from the trained discriminative model.

5 Conclusion and Future Work

We introduced the textual distractor generation task for visual question answering (DG-VQA). These "hard negative" distractors are significantly important when deep learning models have been applied in many real-life and safety-sensitive environments. To address the lacking human-labeled data issue, we developed a policy gradient based model dubbed as MLPR to utilize feedback from well-trained models for distractors generation. The generated distractors achieve high successful rates to make well-trained VQA models misclassify answer choices. Furthermore, the generated distractors also provide insights of factors causing VQA models vulnerable.

We hold the view that the DG-VQA task and the adversarial training towards distractor generation for visual questions pave a new pathway for further research in antiadversarial and explainable VQA. There are several caveats of our method that is worth mentioning. For instance, the alternatives generated are less diverse and the lingering concern of over-fitting by our proposed MPLR and MPLR+Pretrain methods. It sparks future directions such as distractor generation for VQA with explicit and explainable reasons.

References

- Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.
- Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5: 135–146. ISSN 2307-387X.
- Chen, C.-Y.; Liou, H.-C.; and Chang, J. S. 2006. Fast: an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, 1–4. Association for Computational Linguistics.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Fan, Z.; Wei, Z.; Wang, S.; Liu, Y.; and Huang, X. 2018. A Reinforcement Learning Framework for Natural Question Generation using Bi-discriminators. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1763–1774.
- Fang, Z.; Kong, S.; Fowlkes, C.; and Yang, Y. 2019. Modularized Textual Grounding for Counterfactual Resilience. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 457–468.
- Gao, Y.; Bing, L.; Li, P.; King, I.; and Lyu, M. R. 2018. Generating Distractors for Reading Comprehension Questions from Real Examinations. *arXiv* preprint arXiv:1809.02768
- Geman, D.; Geman, S.; Hallonquist, N.; and Younes, L. 2015. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences* 201422953.
- Gokhale, T.; Banerjee, P.; Baral, C.; and Yang, Y. 2020. VQA-LOL: Visual Question Answering under the Lens of Logic. In *European conference on computer vision*. Springer.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.
- Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*. URL http://arxiv.org/abs/1412.6572.
- Gunning, D.; and Aha, D. W. 2019. DARPA's explainable artificial intelligence program. *AI Magazine* 40(2): 44–58.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

- Hill, J.; and Simha, R. 2016. Automatic Generation of Context-Based Fill-in-the-Blank Exercises Using Co-occurrence Likelihoods and Google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, 23–30.
- Ho, J.; and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 4565–4573.
- Jabri, A.; Joulin, A.; and van der Maaten, L. 2016. Revisiting visual question answering baselines. In *European conference on computer vision*, 727–739. Springer.
- Kumar, G.; Banchs, R.; and D'Haro, L. F. 2015. Revup: Automatic gap-fill question generation from educational texts. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, 154–161.
- LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553): 436.
- Li, Y.; and Ye, J. 2018. Learning adversarial networks for semi-supervised text classification via policy gradient. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1715–1723. ACM.
- Liang, C.; Yang, X.; Dave, N.; Wham, D.; Pursel, B.; and Giles, C. L. 2018. Distractor Generation for Multiple Choice Questions Using Learning to Rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 284–290.
- Liu, F.; Xiang, T.; Hospedales, T. M.; Yang, W.; and Sun, C. 2018. iVQA: Inverse visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8611–8619.
- Marino, D. L.; Wickramasinghe, C. S.; and Manic, M. 2018. An adversarial approach for explainable ai in intrusion detection systems. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, 3237–3243. IEEE.
- Mescheder, L.; Nowozin, S.; and Geiger, A. 2018. Which Training Methods for GANs do actually Converge? In *International Conference on Machine Learning (ICML)*.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.
- Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.
- Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, 1060–1069. JMLR. org.
- Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7008–7024.

- Sakaguchi, K.; Arase, Y.; and Komachi, M. 2013. Discriminative approach to fill-in-the-blank quiz generation for language learners. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, 238–242.
- Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.
- Stasaski, K.; and Hearst, M. A. 2017. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 303–312.
- Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.
- Turing, A. M. 2009. Computing machinery and intelligence. In *Parsing the Turing Test*, 23–65. Springer.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4): 229–256.
- Yao, H.; Wang, Z.; Nie, G.; Mazboudi, Y.; Yang, Y.; and Ren, Y. 2019. Improving Model Robustness with Transformation-Invariant Attacks. *arXiv preprint arXiv:1901.11188*.
- Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2018. From Recognition to Cognition: Visual Commonsense Reasoning. *arXiv preprint arXiv:1811.10830*.
- Zhang, J.; Wu, Q.; Shen, C.; Zhang, J.; Lu, J.; and Hengel, A. v. d. 2017. Asking the difficult questions: Goal-oriented visual question generation via intermediate rewards. *arXiv* preprint arXiv:1711.07614.
- Zhang, Y.; Niebles, J. C.; and Soto, A. 2019. Interpretable visual question answering by visual grounding from attention supervision mining. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 349–357. IEEE.
- Zhu, Y.; Groth, O.; Bernstein, M.; and Fei-Fei, L. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4995–5004.