Enhanced Visual Dialog

Mohit Bajaj, Gursimran Singh, Siddhesh Khandelwal
University of British Columbia
Department of Computer Science
(mbajaj01, msimar, skhandel)@cs.ubc.ca

Abstract

We focus on the task of Visual Dialog, which requires an AI agent to hold a meaningful dialog with humans in natural, conversational language about visual content. Given an image, a dialog history, and a question about the image, the agent has to ground the question in an image, infer context from history, and answer the question accurately. Recently, Das et al. [4] came up with an interesting approach using deep reinforcement learning to tackle this problem. Their idea was to pose this task as a cooperative 'image guessing' game between two bots - Q-Bot and A-Bot. Here, A-Bot has access to the image and the task of Q-Bot is to identify the correct image by asking multiple natural language questions. In this work, we propose new architectures for the two bots, with the aim to improve performance on the task. Inspired by [13], we propose using a novel dynamic layer prediction mechanism that, given a question, generates a convolution filter to extract question-specific information from the image. To help reduce the redundancy in the generated questions and also improve the quality of the generated answers, we propose an attention memory to keep track of past dialog information. We also propose a new framework that enables end-to-end training of the two bots. Finally, we explore the use of Generative Adversarial Networks (GANs) [12] to make the dialogue more natural and human like.

1. Introduction

Recent advances in deep learning led to many breakthroughs in natural language [22], computer vision [6] and their intersection like image captioning [19]. Encouraged by these advances, we have seen an increasing interest in answering questions on images, which is setup as a visual Turing test. It has applications like image retrieval, aiding visual impaired people, and human-robot interaction. As a result, there has been significant work over the years in the field of Visual Question Answering (VQA) [1, 15, 11, 18, 21, 10]. While VQA takes a significant step towards human-machine interaction, it still represents only a single round of a dialog, one question followed by an answer. Motivated by the need to develop agents that possess the ability to hold a meaningful dialog with humans in natural language about visual content, [2] proposed a new task of Visual Dialog. Following this, [4, 5, 16] proposed models treating this task as a supervised learning problem. Arguing that the use of supervised methods alone doesn't capture interactive nature of the task, [4] treated this task as a cooperative 'image guessing' game between two agents - where one agent is oblivious to the true image and the other is not. The goal for that agent is to identify the true image by interacting with the other agent through natural language questions. They propose a deep reinforcement learning approach to simultaneously train the two agents. Their new method provides agents with the freedom to steer away from the training data, thus encouraging a more interactive dialog. An example of such an interaction is shown in Fig-

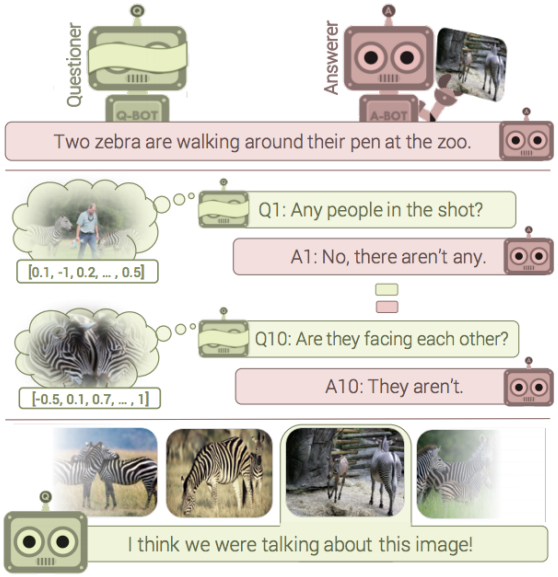

Figure 1. Visual dialogue using reinforcement learning. Figure taken from [4].

ure 1. As seen from the figure, the nature of the task encourages cooperation between the two agents.

However, we identify a few drawbacks of the method proposed by [4].

1. **Q-Bot:** As the Q-Bot is oblivious to the true image, its goal is to identify the correct image by communicating with the A-Bot through natural language questions. However, looking at the results shown in [4], it can be seen that some of the questions asked by the Q-Bot are repetitive in nature.

2. **A-Bot:** As the A-Bot has knowledge about the true image, its job is to correctly answer questions asked by the other agent. In the model proposed by [4], the image features are extracted using a pre-trained model like VGG-16 [17]. However, these features are extracted without being conditioned on the question.

3. **Training the bots:** As the bots are required to interact with each other, at each turn of the dialog a question and answer needs to be sampled from a categorical distribution of words. This sampling procedure introduces non-differentiability during training. [4] proposed a reinforcement learning based approach that uses the REINFORCE algorithm [20] to train both the agents. However, using such methods often lead to models that are harder to train. Also, the loss function used in [4] only influenced by whether the Q-Bot correctly identifies the true image. However, there is nothing enforcing the exchange between the two agents to be 'natural' and 'human like'. As shown in the synthetic experiments in [4], it is possible for the A-Bot to completely ignore the question and just encode the information of the true image in the answer.

In this work, we propose a new architecture and training method that aims to address the issues mentioned above. In particular, we categorize our improvements into three broad categories.

1. **Improving Question Generation:** We propose a new architecture of the Q-Bot as shown in Fig 2. In the new architecture, we augment the exiting Q-Bot with an attention memory to keep track of previous dialog history. We show that after using the new Q-Bot architecture the results of the image retrieval tasks improve.

2. **Improving Question Answering:** Inspired by [13], we implement a novel dynamic layer prediction mechanism that, given a question, generates a convolution filter for better feature extraction. Similar to the reasoning in the previous point, as the A-Bot has to correctly answer the questions, we propose the use of an attention memory in the A-Bot as well.

3. **Improving Interactive Dialog:** We propose a completely differentiable framework that allows for end-to-end training of both the bots. We replace the non-differentiable sample from a categorical distribution with a differentiable sample from a Gumbel-Softmax distribution [7]. We also look at Generative Adversarial Networks (GANs) [12] to encourage more 'natural' conversations.

This paper is organized as follows: Section 2 mentions some of the related work in this area. Section 3 describes our approach in detail. Section 4 looks at the performance of our method on standard benchmark dataset.

2. Related work

2.1. Visual QA

Due to the challenging nature of the task, over the years, there has been significant work in the field of VQA. [1] provide a new dataset with open-ended questions, and also propose a model that combines VGG-16 [17] based image feature extraction with a LSTM based language encoder to predict the correct output. Arguing that VQA often requires multiple steps of reasoning, [21] propose a stacked attention mechanism that allows the model to look at the relevant regions of an image given a question. [10] assert that attention over the question is as important as visual attention. They propose a co-attention model for VQA that jointly reasons about image and question attention.

2.2. Visual Dialogue

[2] proposed a new task of Visual Dialog, which can be viewed as a generalization of VQA. Visual dialog is similar to VQA in the sense that the agent must understand the question and ground the information in the image. However, unlike visual question answering, here the agent is also supposed to keep a context of previous conversation to concisely answer the question. [5] propose a cooperative two-player game, called 'GuessWhat', where both players are given an image. One player is randomly assigned an object in the image. The goal is for the second player is to correctly identify through a series of questions. However, they limit the responses to these questions to 'Yes/No/NA', therefore discouraging natural conversation.

[16] propose a novel visual attention mechanism that employs an associative memory to help resolve ambiguous references in the question. This memory keeps track of all the previous image attention values. Given a question, the model retrieves the most relevant previous in order to resolve potentially ambiguous references. However, due to the supervised training, the machine is not able to steer the conversation since, after each round, the machine generated response is replaced with the ground truth response. Hence, the model is never trained to hold a natural dialogue.

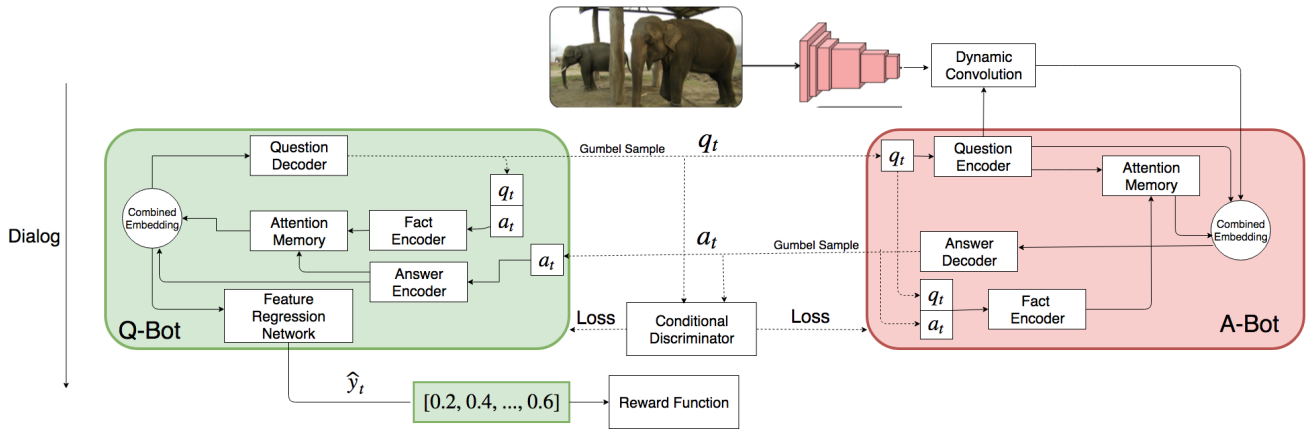

Figure 2. The proposed end-to-end differntiable framework with attention memory, dynamic convolution layer prediction and discriminator.

[4] fix this problem by posing the visual dialog task as a game play between two bots, viz. Questioner Bot (Q-bot) and Answerer Bot (A-bot). The game is setup such that only the A-bot has access to the image, while the Q-bot is tasked to guess the image by asking questions. The authors present it as a reinforcement-learning based problem where both the bots get a reward when the Q-bot predicts the correct image. Hence, in this cooperative game play, the two bots are encouraged to become better at having a more informative dialogue.

3. Method

In this section we define the model and all its elements. We first introduce the architecture of Q-Bot and A-Bot. Later on, we describe the structure of the discriminator and the training mechanism. The complete model is shown in Figure 2.

3.1. Problem Overview

The task is modelled as a cooperative 'image guessing' game between two bots - Q-Bot and A-Bot. The Q-Bot is only given the image caption c and its task is to identify the correct image by asking multiple natural language questions. The A-Bot has access to the image I and caption c, and it has to cooperate with the Q-Bot by correctly answering the questions asked.

3.2. Q-Bot Network

In this section we describe the various components of the Q-Bot.

Answer Encoder
Let a_{t-1} be a natural language answer given by the A-Bot in the previous round $t-1$. This answer is encoded by the Q-Bot using an LSTM. The final hidden state ($\mathbb{R}^5 12$) is used as the representation for this answer A_t^Q.

Fact Encoder
Let q_t be the question asked by the Q-Bot at time t and let a_t be the response it receives to that question. Q-Bot treats this (q_t, a_t) pair as a fact it knows about the image. This fact is encoded by passing it through an LSTM. The final hidden state ($\mathbb{R}^{512}$) is used as the representation for this fact F_t^Q. At timestep 0, the image caption c is treated as a fact.

Attention Memory
The architecture of Q-bot and A-bot mentioned in [4] consists of a state/ history encoder which summarizes the existing conversation. It is supposed to aid the bots to generate informative and meaningful questions/ answers. In particular, given a topic, it helps the bots to retrieve context about previous conversation, figure out things which has been already discussed and generate a response which adds to the previous dialogue. The current implementation achieves this by squashing the entire sequence of facts (q_i, a_i) into a state $\mathbb{R}^{512}$ vector through an LSTM encoder. After a couple of rounds the information of dialogues in the previous rounds become stale, which leads to reduced performance after the first few rounds.

We implement an attention memory module which, for a given answer, computes the attention over all the previous facts and agglomerates them as per the attention weights. Let F_i^Q be the embedding of a fact in Q-Bot at time step $i < t$. At time t, the A_t^Q be the embedding of the latest answer received by the Q-Bot. During each iteration, we generate a new representation E_t^Q by differently attending over the facts in the memory. In particular,

$$m_{i,t}^Q = W_c\,(W_a A_t^Q + W_f F_i^Q), \quad \forall i < t$$

$$r_{i,t}^Q = softmax(m_{i,t}^Q, \quad 1 \le i \le t-1)$$

$$E_t^Q = \sum_{i=1}^{t-1} r_{i,t}^Q F_i^Q$$

Here W_c, W_a and W_f are learned parameters.

Question Decoder
Let the context embedding C_t^Q be defined as the concatenation of attended history representation E_t^Q and the answer representation A_t^Q. The question decoder is a LSTM that takes in this context embedding C_t^Q as its initial hidden state, and generates a question q_t by sequentially sampling words. Instead of sampling from a non-differentiable categorical distribution, we instead sample from Gumbel-Softmax distribution [7]. This sampling is explained in Section 3.4.

Feature Regression Network
This is a fully connected layer that produces an image representation $\hat{y}_t$ from the current episode E_t^Q. This layer tries to combine everything that the Q-Bot knows about the image into a single representation.

3.3. A-Bot Network

In this section we describe the various components of the A-Bot.

Question Encoder
Let q_t be a natural language question asked by the Q-Bot. This question is encoded by the A-Bot using an LSTM. The final hidden state ($\mathbb{R}^5 12$) is used as the representation for this question Q_t^A.

Fact Encoder
This is similar to the fact encoder mentioned in Section 3.2. A-Bot treats a (q_t, a_t) pair as a fact. This fact is encoded by passing it through an LSTM. The final hidden state ($\mathbb{R}^{512}$) is used as the representation for this fact F_t^A. At timestep 0, the image caption c is treated as a fact.

Attention Memory
As the A-Bot is tasked with answering the questions as accurately as possible, it is important for it to remember its previous responses. Currently, in [4], all of that information is captured by a LSTM. Similar to the argument made in section 3.2, this doesn't help the A-Bot reason between multiple dialogues. We augment the A-Bot with the Attention Memory module to pay selective attention on different facts based on previous question and the image representation. Let F_i^A be the embedding of a fact in Q-Bot at time step $i < t$. At time t, let Q_t^A be the embedding of the latest question received by the A-Bot. During each iteration, we generate a new representation E_t^A by differently attending over the facts in the memory. In particular,

$$m_{i,t}^A = V_c\ (V_q Q_t^A + V_f F_i^A),\ \ \forall i < t$$
$$r_{i,t}^A = softmax(m_{i,t}^A,\ \ 1 \leq i \leq t-1)$$
$$E_t^A = \sum_{i=1}^{t-1} r_{i,t}^A F_i^A$$

Here V_c, V_q and V_f are learned parameters.

Dynamic Convolution Layer Prediction
Given an image I, the goal of this module is to represent I as a feature vector. Looking at the implementation in [4], it can be seen that they just use the base VGG-16 model to extract features. However, this implies that the image representation does not depend on the question asked. Inspired from [13], we propose a novel dynamic layer prediction mechanism that, given a question, generates a convolution filter for better feature extraction.

At time t, given the question q_t, we define a Parameter Prediction Network that takes as input the question encoding Q_t^A, the attended history vector E_t^A and generates a convolution filter c_t^A as follows,

$$c_t^A = P_c\ (P_q Q_t^A + P_f E_t^A)$$

Where P_c, P_q and P_f are learned parameters. This convolution filter c_t^A is inserted in the base VGG-16 model. The entire VGG-16 architecture can be divided into two parts: i) Convolutional Layers and ii) Fully Connected Layers. As the initial layers of the VGG network only focus on extracting general features for an image, for maximum impact we insert our dynamic convolution filter at the end of the last Convolution Layer in VGG-16 (after *conv5-3*). Let the image representation I_t^A be the output from the VGG-16 model appended with our dynamic convolution filter.

Answer Decoder
Let the context embedding C_t^A be defined as the concatenation of the attended history representation E_t^A, the current question representation Q_t^A and the image representation I_t^A. The answer decoder is a LSTM that takes in this context embedding C_t^A as its hidden state, and generates an answer a_t by sequentially sampling words. Instead of sampling from a non-differentiable categorical distribution, we instead sample from Gumbel-Softmax distribution [7]. This sampling is explained in Section 3.4.

3.4. Gumbel Sampling

Interaction between the bots requires the questions and the answers to be sampled. Sampling generally being a discrete step results in the model to lose end-to-end differentiability making the training difficult. [4] uses deep

reinforcement learning and treat the bots as policies. They used REINFORCE, a policy gradient method to update policy parameters. We aimed to make the training of this co-operative framework end-to-end differentiable. To do this we use Gumbel sampling to sample the questions and the answers from Qbot and Abot respectively. Gumbel sampling[7] keeps the sampling process differentiable by sampling from the Gumbel-Softmax distribution.

$$y_i = \frac{e^{(log\, p_i + g_i)/\tau}}{\sum_{j=1}^{K} e^{(log\, p_j + g_j)/\tau}}$$

Here $(p_1, ...p_K)$ are the parameters of categorical distribution and $(g_i)_1^K$ denote K IID samples drawn from the gumbel distribution, $g_i \sim F(g) = e^{-e^{-g}}$. τ denotes the temperature parameter which controls how closely drawn samples y approximate the one-hot encoding of the categorical representation. Using this technique, the complete model was trained end-to-end circumventing the need of policy gradient techniques.

3.5. Discriminator

We found when the bots were pre-trained in supervised fashion, the generated dialogs resembled closely to the human-human interaction indicating the effectiveness of pre-training. However, when both of the bots were set-up in interact mode, the conversation was found to deviate from natural language as observed by [4]. We tried to prevent this by using a discriminator that could guide the conversation and prevent the undesired deviation. In our framework, the Q-Bot A-Bot duo acted as a generator network of GAN and a separate network was trained as a discriminator. The network used two separate LSTM encoders to encode the sampled question and the answer. The encoded representation was concatenated with the image features and was fed to the input layer of three layered fully-connected neural network. The output y of neural network is a scalar that denotes the probability that the sampled question-answer pair resembles human dialog and is relevant to the conditioned image. The questions and answers were sampled using Gumbel-Sampling as described above.

$$y = \sigma(h(\{f(Q), g(A), I\}))$$

Given a question Q and an answer A in response to Q, f(Q) and g(A) are the encoded representations from the outputs of LSTM encoders, I denotes image-features and h denotes fully connected neural network. For this framework, learning objective of the generator is to generate dialogs that are indistinguishable from human dialog and can fool the discriminator, while discriminator is trained to distinguish between these two categories. This can be viewed as min-max problem as follows:

$$\min_{\theta} \max_{\eta} \mathcal{L}(G_\theta, D_\eta)$$

Here G_θ and D_η are the parameters of the generator (Q-Bot and A-Bot) and the discriminator respectively. The objective function $\mathcal{L}$ would be:

$$\mathbb{E}_{S\sim P_D}[log\ r_\eta(Q, A)] + \mathbb{E}[log(1 - r_\eta(G_\theta(I)))]$$

Here, P_D denotes the question-answer pairs that are part of a human dialog from the VisDial dataset and $G(I)$ denotes the question-answer pair generated by the generator network.

3.6. Training

Similar to [4], to encourage the bots to have a meaningful conversation in English we use the following training strategy

- **Supervised Pre-training:** We first train both bots in a supervised fashion.

  1. Q-Bot is trained to generate the question in the training data, given the previous dialog history.
  2. A-Bot is trained to generate the answer in the training data, given the previous dialog history.
  3. The feature regression network is trained to generate the true output y.

  This pre-training ensures that the agents can generally recognize some objects/scenes and carry out their conversation in English.

- **Curriculum Learning:** After supervised pre-training, we allow the bots to converse with each other. To prevent the bots from diverging too far from the conversation, we continue supervised training for the first k rounds of dialog and allow the bots to communicate for the remaining $10 - k$ rounds. Unlike [4] that used policy-gradient updates for the two bots using the REINFORCE algorithm, we instead simply perform end-to-end training of both the bots owing to the differentiability of the Gumbel distribution.

3.7. Loss Function

This section describes the loss functions used during training.

MLE Objective: During supervised training, for each bot, this is calculated on each word output of the decoder to ensure that the generated sentence is close to the ground truth sentence. We try to minimize the cross entropy loss between the generated sequence and the true target sequence. Let p_i be the probability vector over the entire vocabulary at timestep i. Then, for a given target sequence

$w_1, w_2, \ldots, w_n$, the cross entropy is defined as

$$loss = -\sum_{i=1}^{n} \log p_i[y_i]$$

Where $p_i[y_i]$ refers to the y_i^{th} entry of the probability vector p_i.

Ranking Loss: During supervised training we use a ranking loss when training the A-Bot. The idea is to enforce the constraint that the ground truth answer $\tilde{a}$ should have a higher log-likelihood (rank) as compared to any incorrect answer. To achieve this we use the Hinge Embedding loss. Let $\tilde{a}_t$ be the ground truth answer at time t and let $\{o_{t,1}, o_{t,2}, \ldots, o_{t,k}\}$ be a set of k incorrect answers. The Hinge Embedding loss can be defined as,

$$loss = \sum_{i=1}^{k} \max\{0, \alpha - d(C_t^A, \tilde{a}_t) + d(C_t^A, o_{t,k})\}$$

Where C_t^A is the context embedding at time t. $d(\cdot)$ is the cosine similarity measure. The value of α is set to 0.2.

Image Guessing Loss: Let $\hat{y}_t$ be the image representation generated by the feature regression network of the Q-Bot (Section 3.2). Let y_t be the true representation of the image corresponding to the $fc7$ (penultimate fully-connected layer) output from VGG-16 [17]. The Image Guessing loss is defined as the L2-norm distance,

$$loss = ||y_t - \hat{y}_t||^2$$

This loss is used to train the feature regression network during supervised pre-training. Also, when the bots converse with each other during curriculum learning, this loss is backpropagated end-to-end through both the bots. More specifically, at a given time t, the Q-Bot generates a question q_t. Then the A-Bot generates an answer a_t corresponding to the question q_t. Finally, the feature regression network generates a new estimate image representation using the additional information (q_t, a_t). The loss is calculated over this newly generate representation $\hat{y}_t$ and the true representation y_t.

4. Experiments

This section mentions the details of the dataset used and experimental results.

4.1. Dataset

We conduct our experiments on the VisDial v0.5 [3] dataset. The details about the dataset size are mentioned in Table 1. For each image, the dataset contains a conversation of 10 turns between two humans. Each turn also contains a list of 100 candidate answers, including the ground truth answer. The images are taken from the MS-COCO dataset [9].

Table 1. VisDial v0.5 dataset

	Number of Images
Train	50,729
Validation	7,663
Test	9,628

4.2. Implementation Details

All the models were implemented in PyTorch [14]. We used the Adam optimizer [8] to train our models. The learning rate was set to 0.001. All the LSTMs used in our models are two layered with a hidden state of 512. The training was done as mentioned in Section 3.6. The code is publicly available at
https://github.com/siddheshk/CS532L-Course-Project

4.3. Notation

We define a few natural ablations of the A-Bot and Q-Bot, which are compared in the sections below. Prefixes of the different models use the following naming terminology:
`<Bot Type>-<Training Method>-<Modules>`

- **Bot Type:** This can assume two values - `ABot` and `QBot`.
- **Training Method:** This can assume two types. `SL` assumes the bots are only trained using supervised pre-training. `Interact` assumes that the bots are trained using curriculum learning, that is the bots are trained by making them communicate with each other.

The different models are defined below, which are compared in the later sections.
For A-Bot,

- **`ABot-<Training Method>-LSTM`:** This is the baseline model proposed by [4]
- **`ABot-<Training Method>-AttMem`:** This is our A-Bot model using only the attention memory described in Section 3.3.
- **`ABot-<Training Method>-AttMem-RankLoss`:** This is our A-Bot model using the attention memory and Ranking Loss described in Section 3.7.
- **`ABot-<Training Method>-AttMem-DynCNN-RankLoss`:** This is our A-Bot model using the attention memory, Ranking Loss and the Dynamic Convolution Filter Prediction described in Section 3.3.

For Q-Bot,

- **`QBot-<Training Method>-LSTM`:** This is the baseline model proposed by [4]

Table 2. Ablation Results for A-Bot. MRR stands for mean reciprocal rank. Higher is better for MRR and recall@k, while lower is better for mean rank.

Model	MRR	Recall@5	Recall@10	Mean Rank
ABot-SL-LSTM	0.423	0.518	0.585	22.35
ABot-SL-AttMem	0.427	0.521	0.588	21.885
ABot-SL-AttMem-RankLoss	**0.430**	**0.527**	**0.595**	**21.782**
ABot-SL-AttMem-DynCNN-RankLoss	0.418	0.515	0.582	22.73
ABot-SL-AttMem-DynCNN-RankLoss (Masked Image Features)	0.416	0.514	0.580	22.821
ABot-Interact-AttMem-RankLoss	0.425	0.524	0.594	21.92

- **`QBot-<Training Method>-AttMem:`** This is our Q-Bot model using the attention memory described in Section 3.2.

4.4. Discriminator

We pre-trained the discriminator to distinguish between human like interaction from the unnatural conversation. We treated question-answer pairs drawn from VisDial dataset as real. To obtain fake samples we introduced some noise randomly in the question-answer pairs. We also shuffled some pairs to obtain more fake samples. Though the questions and answers from these pairs were semantically correct and had a sensible meaning but they were either not relevant to each other or to the conditioned image. The pre-trained discriminator had accuracy close to 79% for the classification. Then this discriminator was used in adversarial setting during interact set-up. We tried several training schedules that include updating only the generator keeping the discriminator fixed, updating them both alternatively and updating them at different rates. We found that the discriminator loss was not enough to prevent the conversation to deviate from natural language. This indicated the need of stronger regularization so we incorporated MLE loss with the image retrieval loss to preserve the naturalness of the conversation. We found that this solved the problem but the reason behind the failure of adversarial approach is not very clear. This analysis remains as one of the objectives for our future work. Due to the failure of the discriminator, we don't use the discriminator during the training process in the experiments mentioned later on.

4.5. Closeness to Human Dialog

To analyze the ability of our model to emulate human dialog, we look at the performance of the A-Bot on the test split of the VisDial v0.5 dataset. Similar to [4], we instead use a ranking based metric to quantify the performance of the A-Bot. At each turn, the 100 options associated with each question-answer pair are ranked by calculating the likelihood using the answer decoder. The relative rank of the ground truth answer in these 100 options is used to calculate the following metrics: mean reciprocal rank (MRR), recall@k for $k = 5, 10$ and mean rank.

We compare a few natural ablations of the A-Bot (Described in Section 4.3). The results are summarized in Table 2. We make the following observations,

- **Attention Memory improves performance.** It can be seen that `ABot-SL-AttMem` outperforms `ABot-SL-LSTM` across all metrics.
- **Ranking Loss improves performance.** It can be seen that incorporating the ranking loss further improves the performance of the A-Bot. `ABot-SL-AttMem-RankLoss` outperforms `ABot-SL-AttMem` across all metrics.
- **Dynamic Convolution Layer Prediction hurts performance.** As seen from the results, using a dynamic convolution filter prediction mechanism decreases the performance of the model significantly. To further analyze the reason behind this poor performance, we masked out (zero out) the image features from the trained `ABot-SL-AttMem-DynCNN-RankLoss` model before generating the answer. The results of this experiment is shown in Table 2. As observable, the performance of the model doesn't change a lot even when the image is zeroed out. This implies that our dynamic prediction layer parameters weren't learned well, which might be because the task itself doesn't rely on the image a lot. The performance of other variants of the A-Bot is slightly better because it uses the output from VGG-16, which was trained on a larger dataset for a task where the image is important.

As `ABot-SL-AttMem-RankLoss` is the best performing model, we allow it to converse with Q-Bot and train it using the curriculum learning approach mentioned in Section 3.6. The Q-Bot used during this interaction is `QBot-SL-AttMem`. The performance of the resulting trained A-Bot, which we refer to as `ABot-Interact-AttMem-RankLoss`, is compared against its supervised counterparts. The result in shown

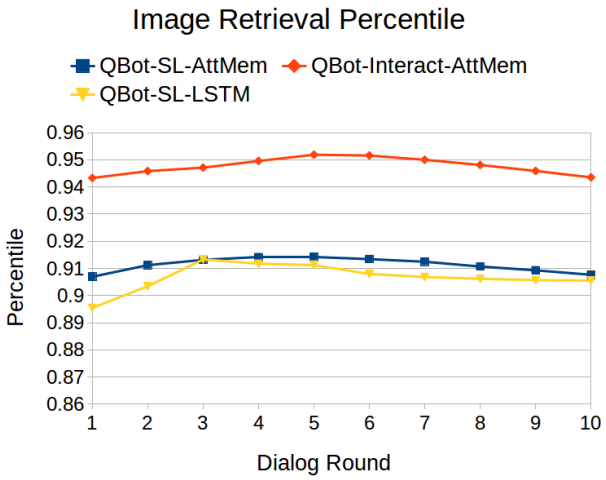

Figure 3. Image Guessing Game Evaluation

in Table 2. We observe that the performance of `ABot-Interact-AttMem-RankLoss` is lower than `ABot-SL-AttMem-RankLoss`. We think this might be because of the fact that during curriculum learning, the models are trained to improve their performance on trying to identify the correct image, instead of trying to emulate the ground truth answers. Therefore due to the degree of freedom that the bots have when the interact with each other, even though the A-Bot might generate correct answers, they may not necessarily mimic ground truth human responses.

4.6. Guessing Game Evaluation

To analyze how well the agents have learned to cooperate at the image guessing task, similar to [4], we instead set up an image retrieval experiment. We present an image and a caption to the agents, and allow them to communicate over 10 rounds of dialog. After each round, Q-Bot predicts a image feature representation $\hat{y}_t$. We sort the entire test set in ascending distance to this prediction and compute the rank of the source image.

We compare natural ablations of the Q-Bot (Described in Section 4.3). For `QBot-SL-LSTM` the corresponding A-Bot model used is `ABot-SL-LSTM`. For `QBot-SL-AttMem` the corresponding A-Bot model used is `ABot-SL-AttMem-RankLoss`. For `QBot-Interact-AttMem` the corresponding A-Bot model used is `ABot-Interact-AttMem-RankLoss`. We measure the mean percentile rank of the source image across rounds. The results are shown in Figure 3. We make the following observations,

- **Attention Memory in Q-Bot improves Image Guessing.** As seen from Figure 3, `QBot-SL-AttMem` outperforms `QBot-SL-LSTM`.
- **Attention Memory in Q-Bot makes the performance slightly more stable.** We observe a sharper rise and decrease in performance in the case of `QBot-SL-LSTM`. On the other hand, `QBot-SL-AttMem` seems more stable.
- **Allowing the bots to communicate improves image identification.** We observe that `QBot-Interact-AttMem` significantly outperforms its supervised counterpart. This indicates that our end-to-end differentiable framework is effective at training these agents for image guessing.

4.7. Qualitative Analysis

Figure 4 contains a few example of the dialogs sampled by making `QBot-Interact-AttMem` and `ABot-Interact-AttMem-RankLoss` converse with each other. As seen from the figure, the question and answers generated are sometimes very generic like "It's hard to say" and "I think so". We also see that some of the answers given are completely incorrect. This is further evidence to the observation that images are not that important in this task. However, we do observe a reduction in the number of repeated questions, which was a problem in [4].

5. Conclusion and Future Work

To summarize, we improved the framework presented in [4] on three different aspects. Firstly, we improved the question generation by augmenting the Q-Bot with attention memory resulting in the improvement of image retrieval results. Secondly, we improved the answer generation capability of A-Bot by adding dynamic convolution filter prediction, attention memory and ranking loss. The attention memory helped in improving the performance of the A-Bot. However, we found that the dynamic convolution filter prediction did not help with the performance leading to the observation that images are not crucial for this task for the given dataset. Finally, we made the whole framework end-to-end differentiable using Gumbel-Softmax which makes the model easier to train. Further, we implemented a discriminator to prevent the messages exchanged between bots to deviate from human-interpretable meaningful conversation. However, we found that the discriminator is hard to train and didn't help much in this pursuit. It would be interesting to know if these observations are consistent to with other datasets. In the future we plan to verify our observations on other datasets like the VisDial v0.9 dataset [3] and the MNIST Dialog [16] dataset. Other interesting area of exploration that we wish to pursue is to test the effectiveness of our Dynamic Convolution Prediction layer on other tasks or datasets where images are important. We also plan to further analyze and look at techniques to improve our discriminator framework.

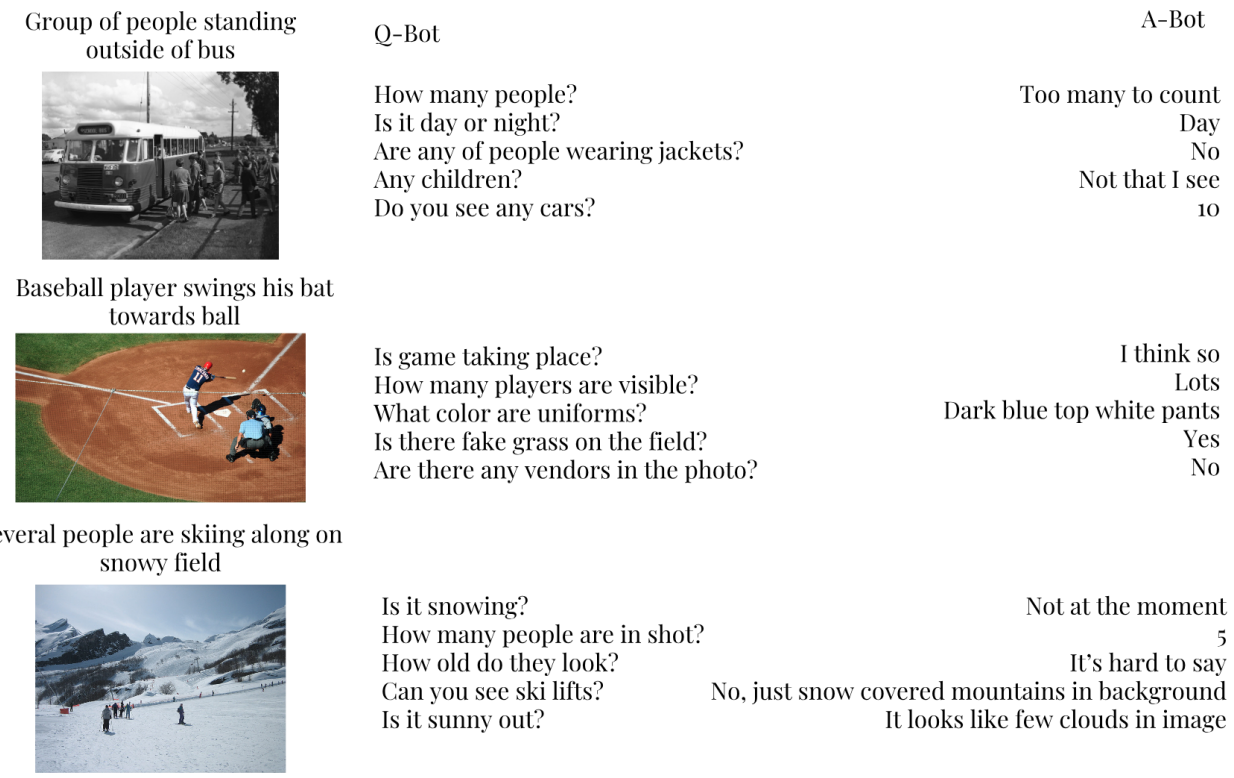

Figure 4. A few examples of dialogs sampled from our model. The image, caption and the outputs of the Q-Bot and A-Bot are shown.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[2] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[3] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585*, 2017.

[5] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proc. of CVPR*, 2017.

[6] S. Goyal and P. Benjamin. Object recognition using deep neural networks: A survey. *arXiv preprint arXiv:1412.3684*, 2014.

[7] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[11] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1–9. IEEE Computer Society, 2015.

[12] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[13] H. Noh, P. Hongsuck Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 30–38, 2016.

[14] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[15] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pages 2953–2961, 2015.

[16] P. H. Seo, A. Lehrmann, B. Han, and L. Sigal. Visual reference resolution using attention memory for visual dialog. In *Advances in neural information processing systems*, pages 3722–3732, 2017.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640, 2016.

[19] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2017.

[20] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[21] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.

[22] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *arXiv preprint arXiv:1708.02709*, 2017.