Visual Dialog System - Interactive mode

Mohamed Adnen ABDESSAIED

Study program: Simulation Technology

Matriculation number: 3158373

supervised by:
Prof. Dr. Ngoc Thang VU
&
Dirk VÄTH

SimTech research project 2

University of Stuttgart
Institute for Natural Language Processing (IMS)
Winter Semester 2019-2020

Contents

I. Abstract

One of the ultimate goals of artificial intelligence in the field of NLP is to train agents capable of maintaining a conversation with a human for some period of time. Although this task has be solved by a couple of companies like Google and Amazon with their AI assistants; Google assistant and Alexa respectively, many attempts were made to push the boundaries of what such agents are capable of doing. Das et al. proposed a visual dialog task in [1] where the AI-agent is challenged to answer questions about a picture provided first with only its caption and then, as the conversation evolves, with the dialog history. The images used for this task are collected from the COCO dataset [2] and the conversations are generated using the Amazon Mechanical Turk (AMT).

In [1], a variety of encoder-decoder network architectures were suggested: 3 encoders - Late Fusion, Hierarchical Recurrent Encoder and Memory Network- and 2 decoders - discriminative and generative. All of the combinations of the aforementioned encoders and decoders outperformed many established and sophisticated baselines. However, until now the agents were trained offline on a huge dataset of 120K images and $1,2$M question-answer pairs.

We tried in this work to introduce an interactive training mode that allows us to integrate the human-in-the-loop concept into the visual dialog task. This interactive mode lets humans evaluate the agent and give it feedback that can be used for online training purposes. In addition, we investigate the possibility of improving the quality of learning using only limited amounts of data.

Further information about the dataset and the overall settings of the task can be found at https://visualdialog.org/.

Keywords: Visual dialog, encoder-decoder networks, interactive training, human-in-the-loop.

II. Introduction

In the last decade, machine learning and AI in general dominated many fields in science and engineering as they outperformed the most sophisticated non learning-based methods. Whether in computer vision (CV) [3], natural language processing (NLP) [4], classification or even high-level AI tasks like playing chess or GO [5], machine learning established itself as the number one option when trying to deal with and solve such tasks. In the last years, new problems started to emerge dealing with multi-modal data as a consequence of many fields becoming more and more intertwined with one another. In fact, some of the most abundant types of data are images and text which happen to be the main constituents of the task we will address in this work: Visual Dialog. It is one of the most daunting problems at the intersection between computer vision and natural language processing. The main objective here is to train intelligent agents capable of conversing with humans based on some image by accurately answering questions with respect to it. This has tremendous application potential and can be used to improve and facilitate people's lives. Some of the applications may include:

- Assisting people while driving their cars.
  Human: Are there any cars approaching our blind spots?
  Agent: Yes.
  Human: How many?
  Agent: only one.
  Human: How far is it?
  Agent: about 3 meters.
- Improving our experience with AI assistants.
  Human: OK Google! can you see the TV in the living-room?
  Agent: Yes.
  Huamn: Is it on?
  Agent: Yes.
  Human: Turn it off.

Visual dialog is not the only task in the intersection between computer vision and natural language processing. Image captioning [6] and visual question answering (VQA) [7] are classic examples that fall under the same category. However, visual dialog has an utterly different definition than these two.

Visual dialog VS image captioning: The main difference between image captioning and visual dialog is that the former does not portray any form of human conversation whatsoever. It simply provides a human with an over-all description of some image.

Visual dialog VS visual question answering: Despite the fact that visual question answering comes a little bit closer to achieving human-machine interaction than image captioning, it falls short at maintaining a somewhat satisfactory conversation as long as its duration/length is concerned. VQA is in fact a special case of visual dialog consisting of only one dialog round, i.e. one question-answer pair. Although the visual dialog data and the VQA data seem to appear similar, they have two fundamental differences:

- **Coreference in dialog:** As each dialog has 10 rounds, i.e. 10 question-answer pairs, the presence of many pronouns becomes inevitable. Thus, the agent has to overcome some ambiguities and figure out what/whom these pronouns refer to. Some statistics made in [1] show that the amount of such pronouns gradually increase as the conversation evolves which is in accordance with what actually happens in real world scenarios.
- **Temporal continuity in topics:** Das et al. showed that the conversations also have some continuity of the topics being discussed. In fact, they conducted a human study to annotate the questions of 40 random images, i.e. 400 questions in total. The annotation was based on predefined topics, e.g. *asking about an object, a scene, the weather or even the image itself.* Across the 10 rounds, approximately 4.55 topics were discussed suggesting that they are not independent of each other. This implies that the data has some sort of temporal topic continuity.

As you may have already guessed, acquiring data for the visual dialog task is a tedious and long process, especially if one needs huge amounts of data. This may put major limitations to the potential of this task as not many people have the appropriate hardware and the resources to collect such data. Another major issue of the current learning strategies is the dominance of offline training. The latter requires also special hardware such as powerful GPUs or the access to some cloud.

In this work we try to address these two issues at once by introducing an interactive mode that enables us to take action and actively contribute to the training of such agents whilst generating new data.

III. Related work

Many attempts were made to improve on the work of Das et al. [1]. In [8], Niu et al. introduced a novel recursive visual attention strategy to tackle the most daunting problem of visual dialg: co-reference between questions and dialog histories. [9] introduced a goal-driven training for visual question answering and dialog agents using deep reinforcement learning. Two agents Q-Bot and A-Bot communicate in natural language dialog so that Q-Bot can select an unseen image from a lineup of images. Although this improvement led to better performance at the downstream dialog-conditioned image-guessing task, this improvement saturates and starts degrading after a few rounds of interactions, and thus, cannot lead to a better visual dialog model. [10] tried to improve the latter and explained its failure by the fact that the repeated interactions between Q-Bot and A-Bot during self-talk are not informative with respect to the image. Their solution to this problem was devising a simple auxiliary objective that incentivizes Q-Bot to ask diverse questions. This reduced question repetitions and enabled A-Bot to explore a larger state space during reinforced learning.

IV. Approach

Our approach addresses one key improvement that was not considered by all of the aforementioned methods as the human interaction with the system was always left out of the equation. With other words, there was no human supervision during or after training. Our approach aims at solving this issue by introducing an interactive mode based on the human-in-the-loop concept and online training/fine tuning.

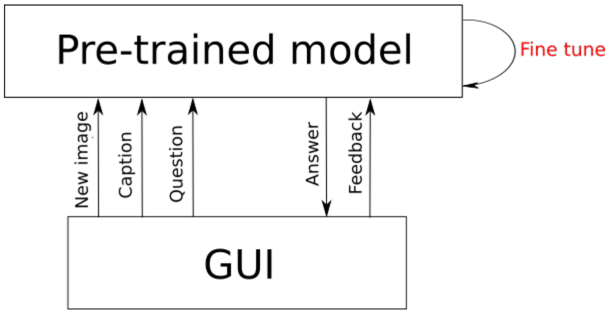

Figure 1: Interactive mode

Our approach is based on the network architectures introduced by Das et.al. in [1] and consists of 3 main steps as Fig. 1 shows: pre-training, interactive mode, fine tuning (FT).

A. Architectures

In [1], different network architectures were suggested to tackle the visual dialog task: 3 encoders (Late Fusion Encoder, Hierarchical Recurrent Encoder and Memory Network Encoder) and 2 decoders (discriminative and generative). In the following we adapt the same network naming convention as in [1], i.e. <encoder>-<input>-<decoder>. E.g. MN-QIH-G means that we use a Memory Network encoder with question-image-history input and generative decoder. Throughout this work, we only consider the generative decoders as they are easier to train on newly generated data, i.e. we do not need to provide a list of 100 candidate answers at each dialog round like in the discriminative setting. Furthermore, we think that generative decoders resemble more the way humans interact with each other: one person asks a question and the other one generates an answer rather than picking up the best response from a set of possible answers. Moreover, we only considered the Late Fusion and the Memory Network encoders. The reason for this choice is twofold. First, the Late fusion encoder is the simplest amongst the three and is the easiest to train. Second, the Memory Network encoder is the only one with an attention mechanism over the conversation history. Therefore, we expect it to perform somewhat better than the remaining two. Fig. 2 and 3 illustrate the architecture of these two encoders.

On the other hand, we use the scheme presented in Fig. 4 to generate answers, i.e. the word at step i is the one that maximizes the probability distribution given all the previous words as well as the encoder output, i.e.

$$w_i = \underset{w \in V}{\arg\max}\, P(w|w_{i-1}, ..., w_1, out_{encoder}),$$

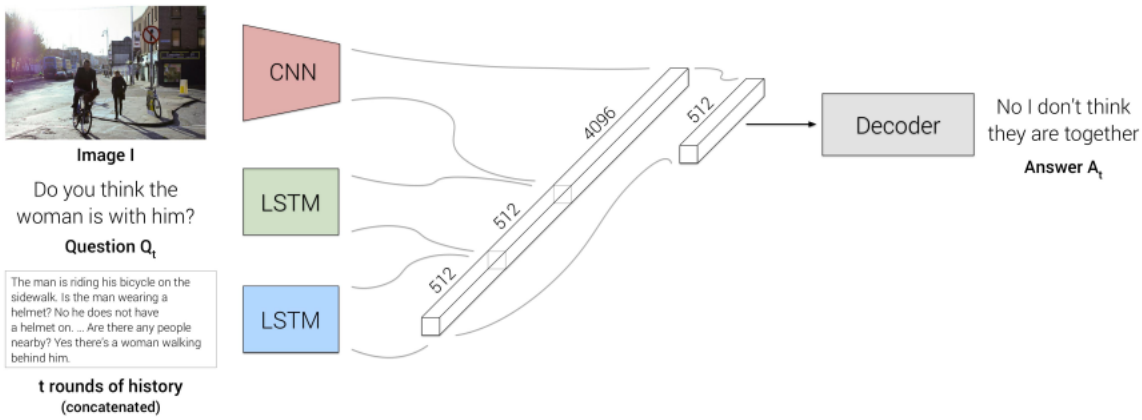

Figure 2: Late Fusion Encoder taken from [1]

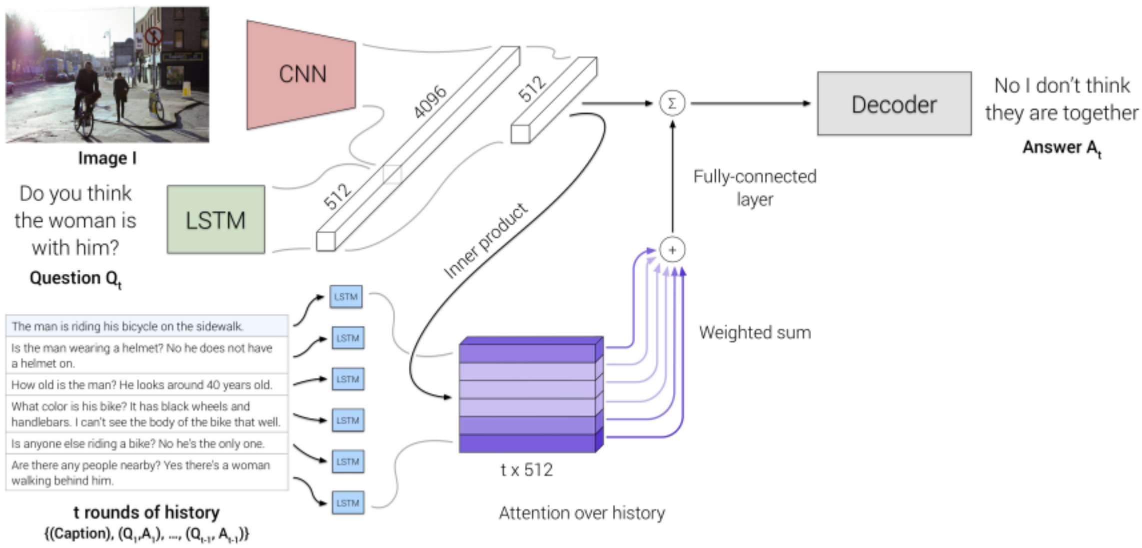

Figure 3: Memory Network Encoder taken from [1]

where w_i and $out_{encoder}$ denote the word generated at step i and the encoder output respectively and V is our vocabulary introduced in section IV-B1 .
In this work, we fix $w_1 =<$ SOS $>$, and generate words until $w_i =<$ EOS $>$ or $n = n_{max}$. We use a value of 10 for n_{max}

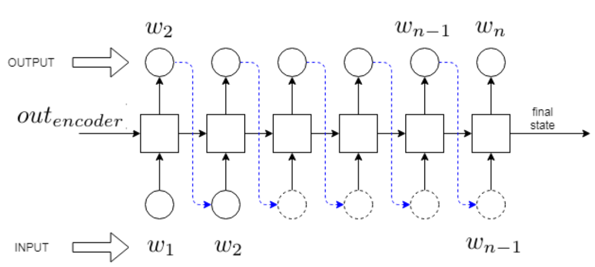

Figure 4: Answer generation diagram

B. Pre-training

Many efforts were made to collect valuable data for the Question-Answering task: Factoid Question-Answer corpus [11], the Q&A dataset of DeepMind [12], 100K SimpleQuestions dataset [13], the bAbI dataset [14], the SQuAD dataset for reading comprehension [15]. All of the above mentioned datasets do not take into consideration the visual part of our setting, i.e. they do not include neither images nor videos. Although the VQA dataset solves this problem, we use the VisDial dataset [1] - for those reasons discussed in II - to pre-train the baseline networks of Section IV-A . In this work, we opt for v-0.9 of the VisDial dataset instead of v-1.0 in order to be able to compare the performance of our pre-trained networks to those introduced in [1] as there is till now no official data about the performance of these networks on v-1.0.

1) Hyperparameters: All the networks have LSTMs with 2 layers and 512-dim hidden states. The word embedding is trained from scratch on the entire word corpus of the training data. The latter is shared across both the encoder and the decoder. We use the l2-normalized activations from the penultimate layer of VGG16 [16] to encode the image. As optimizer, we use Adam [17] and clamp the gradients to $[-5, 5]$ at each step to avoid gradient explosion. Furthermore, we set the dropout value to 0.2 and we train the network for 20 epochs with batch

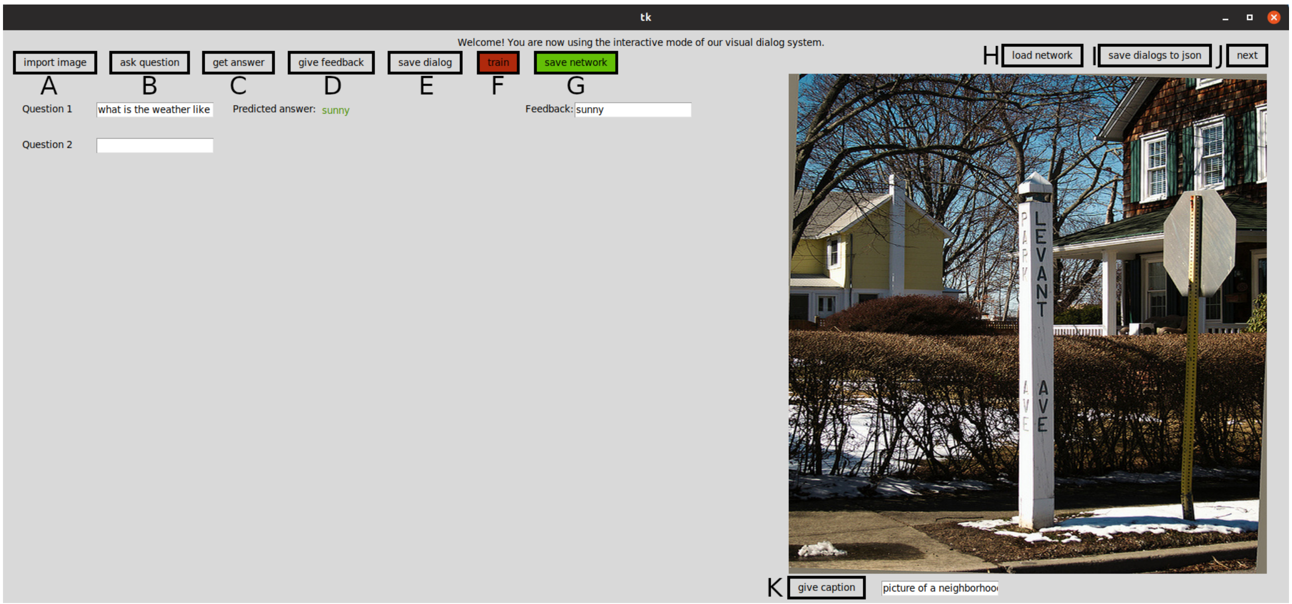

Figure 5: Annotated GUI of the interactive mode

size 8. Dropout was used in all LSTMs and after every fully connected layer.

2) Results: Table I shows the performance of the pre-trained LF-QIH-G and MN-QIH-G networks measured on the validation set of the VisDial v0.9 containing 40K samples.

Model	MRR	R@1	R@5	R@10
LF-QIH-G	0.4791	35.48	60.96	66.93
MN-QIH-G	0.4903	37.80	60.33	66.10

Table I: Performance on the VisDial v0.9 validation dataset measured by mean reciprocal rank (MRR) and recall @k.
LF-QIH-G: Network with late fusion encoder, question-image-history as input and generative decoder.
MN-QIH-G: Network with memory network encoder, question-image-history as input and generative decoder.

As we can see, our networks achieved comparable results to those of [1]. In addition, based on the R@1 metric, which happens to be the most significant one, MN-QIH-G outperformed LF-QIH-G as [1] suggested. R@k computes the percentage of the ground truth answers being in the top k predictions of the network. As a result, R@1 reflects how often the network generates the best answer possible, i.e. the ground truth answer for a given question. Although R@k for $k = 5, 10$ have to be high to ensure a good performance, they do not reflect the goodness of training as well as R@1 does. For example, the 9 best answers that the network generates to a particular question could be very bad and do not relate to the context of the conversation at all but the 10th best answer of the network happens to be the ground truth answer. However, these insights will not be reflected in the R@10 value which can result in erroneous and misleading assessment of the network. This problem can be solved by considering the R@1 values instead.
Based on the results of Table I , we can safely assume that our networks are ready to be deployed within our interactive mode. In fact, this will be solidified afterwards in section IV-C2 by examining their performance on our dataset. It is worth mentioning, that since MN-QIH-G performed better than LF-QIH-G based on R@1, it was used in the rest of this work for generating our dataset as well as for FT.

C. Interactive mode

1) Usage description: After pre-training, the network is used for inference on unseen data by our interactive mode. The latter takes the form of a graphical user interface (GUI) as illustrated in Fig. 5 . The interactive mode is similar to the process of collecting data described in [1]. The main difference consists in generating the data whilst assessing the network performance. Our goal here is to collect new data and use it to incrementally fine tune the last linear layer of the decoder net and therefore improve the overall performance of the system.

The user starts off by loading a pre-trained network and then importing an image using the button H and A of Fig. 5 respectively. If the image is successfully imported, it will be displayed on the right-hand-side of the GUI alongside the "give caption" button underneath it, i.e. button K of Fig. 5 . The user has to annotate the image with a suitable caption and then hit the "Enter" key to save it and be able to start asking questions. This can be done with the help of button B of the GUI. After the user has asked the question, they has to save it by hitting "Enter". At this stage, the user is able to get the generated response of the network using the button C of the interface. Since we want to compute some kind of accuracy afterwards, the user has to give a feedback in a text form to the network using the button D. After a feedback is given, it has to be saved by hitting the "Enter" key. This feedback will play the role of the ground truth of the corresponding question. The dialog can be saved using the button E and only after saving it can we train, i.e. fine tune the network (button F). This option makes online learning possible and allows us to asses the gradual improvements of our system as we continue interacting with it. The fine tuned version of the network can be saved at any time using the button G of our interface. After the user finishes asking questions about one particular images, they can proceed to the next one using the button J. It is worth mentioning that one can ask 10 questions at maximum on one particular picture since the network was trained with dialogs of only 10 rounds. At the end, the generated dialogs can be logged into a json file using the button I of the interface.

2) Our data: In this section, we will describe the data we collected using our interactive mode. We used the first 100 images from the visual dialog testing set that can be found at https://visualdialog.org/data. The dialogs generated have an average length of 7.08 rounds. The longest dialog consists of 9 rounds and the shortest of only 3. The following figures illustrate some examples.

Figure 6: Example 1

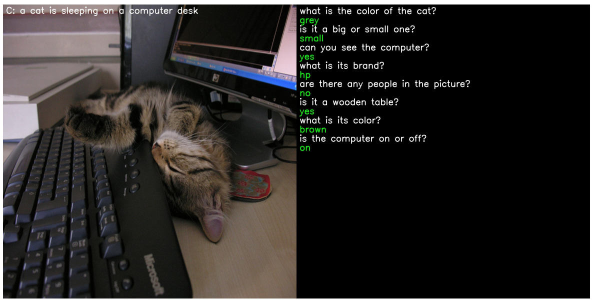

Figure 7: Example 2

D. Fine tuning

The fine tuning step consists of updating the weights of the last fully connected layer of the decoder network. This layer is responsible for generating words from the LSTM outputs. Thus, fine tuning the last layer will update the distribution of the words given all the previous ones as well as the encoder output , i.e.

$$P(w_i|w_{i-1}, ..., w_1, out_{encoder}).$$

The hyperparameters used for fine tuning are the same as described in IV-B1 . The only difference is that we train only for a total of 4 epochs with batch size 1. Furthermore, the network was fine tuned only once on our entire dataset and not gradually during data generation.

V. Results

A. On our data

We first of all compare the performances of the network on the data we generated via the interactive mode, i.e. the data used for FT, before and after FT. This is a good way to see whether the network has learned something from the FT step or not. To do so, we investigated the generated answers in both cases and computed the corresponding network accuracies. This was done on a per dialog basis, i.e. a human assessed the goodness of the answers of the system and gave it 1 point if the answer was acceptable and 0 points otherwise. Note here that there is not a unique right/acceptable answer for each question. Thus, basing the accuracy solely on the ground truths, i.e. the feedback of the user as mentioned in section IV-C1 , does not make much sense. E.g. if the question and its ground truth were "*How big is the room?*" and "*It is big*" respectively than it would be unfair to reject an answer from the system like "*The room is big*" or just simply "*big*". Some datasets like VisDial [1] v.1.0 addressed this issue differently by introducing the so called "dense annotations". However, we opted for the simpler manual evaluation as our dataset only consists of 100 samples. The accuracies were then used to fit a PDF as illustrated in Fig. 8 . For testing we took a similar approach as fine tuning, i.e. we tested only once on our data after the network was fine tuned.

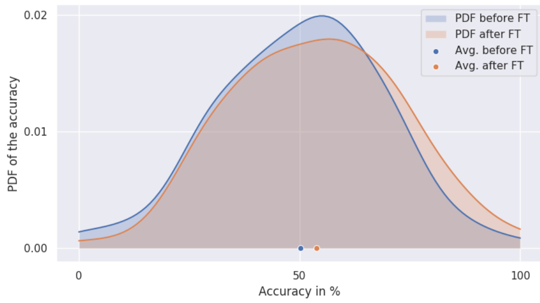

Figure 8: PDFs of the accuracy

As we can see, the PDF of the accuracy has shifted with a considerable amount towards 100% which indicates that the number of dialogs with high accuracy rates has increased after FT at the expense of those with low accuracy rates. This suggests that we can indeed improve the performance of the system by simply updating the last layer only. In fact, the network after FT scored an average accuracy of approx. 53.80%. That is a 3.55% increase in comparison with the 50.25% average accuracy before FT. This difference as small as it may appear is not to be underestimated especially due to the fact that the FT data is negligible in comparison with the original data used to pre-train the network.

We also want to emphasize on the fact that the fine tuned system did not simply reproduce the same answers given as ground truth by the user. In fact, it produced utterly different but acceptable answers in many cases. The following example illustrates this very clearly. We use Q_i, GT_i, A_i, A_i^{ft} to denote the question, the ground truth, the answer before FT and the answer after FT of dialog round i respectively.

Figure 9: A picture of a mother zebra and its son

Caption: A picture of a mother zebra and its son.
$\mathbf{Q_1}$: Where are the zebras?
$\mathbf{GT_1}$: **In the wild.**
$\mathbf{A_1}$: They are standing in the water.
$\mathbf{A_1^{ft}}$: **In a field.**
$\mathbf{Q_2}$: Do you see green plants?
$\mathbf{GT_2}$: Yes.
$\mathbf{A_2}$: no
$\mathbf{A_2^{ft}}$: No.
$\mathbf{Q_3}$: Are they running or standing?
$\mathbf{GT_3}$: Standing.
$\mathbf{A_3}$: Standing.
$\mathbf{A_3^{ft}}$:Standing.
$\mathbf{Q_4}$: Are they eating grass?
$\mathbf{GT_4}$: No.
$\mathbf{A_4}$: No.
$\mathbf{A_4^{ft}}$: Yes.
$\mathbf{Q_5}$: Is the sun shining?
$\mathbf{GT_5}$: Yes.
$\mathbf{A_5}$: Yes.
$\mathbf{A_5^{ft}}$: Yes.
$\mathbf{Q_6}$: Do they look friendly?
$\mathbf{GT_6}$:Yes.
$\mathbf{A_6}$: Yes.
$\mathbf{A_6^{ft}}$: Yes.

We are mostly interested in the first dialog round where the system initially failed to answer the question correctly. However, after FT it succeeded in generating a totally acceptable response which is different than the ground truth: "*In a field*" vs "*in the wild*".

B. On a subset of the VisDial v1.0 validation dataset

In the previous section, we showed that our FT method did indeed improve the performance of the system on our data, i.e. the data used for FT, and the network succeeded not to reproduce the same answers given as ground truths. However, we still wanted to see how the network performs on unseen data, i.e. data not seen during both pre-training and FT.
Although we do not expect the fined tuned network to outperform the pre-trained one since the amount of FT data is very limited and only the last layer was updated, we must not witness any huge drop in the generalizability of the system as this defies the whole purpose of our FT approach.
In order to assess the generalizability of our fine tuned network, we used a subset of the VisDial v1.0 validation dataset. This subset was randomly chosen and contains 100 samples. Moreover, it was not used for pre-training the network as we used the VisDial v0.9 training dataset for that purpose. Another reason that made us opt for this dataset is the fact that it comes with the dense annotations used to compute the normalized discounted cumulative gain (NDCG). It is a very fair metric to measure the performance of such AI agents since it alleviates the non-uniqueness problem of the answers addressed in section V-A . Detailed information about the NDCG metric can be found at https://visualdialog.org/challenge/2019#evaluation . In the following, we took the same approach to assess the networks as we did before, i.e. we investigate the PDF of each metric separately before and after FT.

Model	**NDCG** $\times$**100**	**MRR**	**R@1**	**R@5**	**R@10**
Before FT	57.89	46.01	34.90	56.30	62.60
After FT	57.67	45.56	34.10	56.10	62.40

Table II: Performance on a subset of VisDial v1.0 validation dataset measured by normalized discounted cumulative gain (NDCG), mean reciprocal rank (MRR), recall @k and mean rank.

Table II and Fig. 10 - 14 show the influence of FT when the network is tested on unseen data. Although the density functions of the different metrics are slightly modified after FT, their average values are almost the same as table II shows. This suggests that FT does not deteriorate the generalizability of the system. However, by using this interactive mode more intensively and thus generating richer data we could outperform the baseline system and achieve higher results after FT.

VI. Conclusion and Future Work

In this work, we have addressed the issues of efficiently acquiring valuable data for the visual dialog problem while incorporating human supervision. This was done by introducing an interactive mode that helps generate dialogs while interacting with a pre-trained system. Moreover, we have shown that incrementally fine tuning a pre-trained model can push its performance and helps it improve its affinity and creativity in answering questions as we have seen in the example of section IV-C2 . However, FT only the last layer of the decoder and with limited amounts of data seems not to have a huge impact when the network is tested on unseen data, i.e. data not used in both pre-training and FT. In fact, this offers a significant margin to improve our method in the future. Moreover, the choice of the optimal weights to fine tune remains a very daunting yet crucial task that we aim to solve in future endeavors in order to improve the generalizability of our agent. Our generated data using the interactive mode and our code can be found at https://github.com/adnenabdessaied/VisDiag .

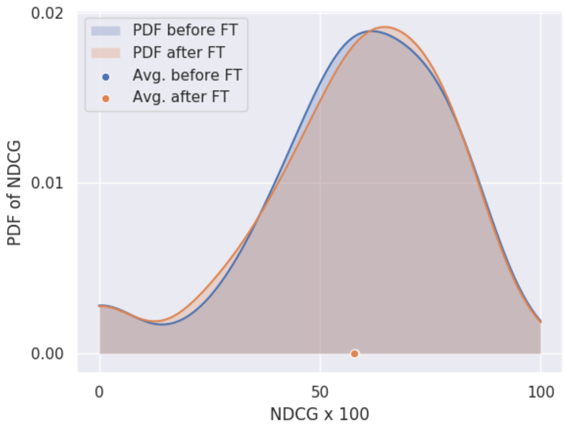

Figure 10: PDFs of the NDCG

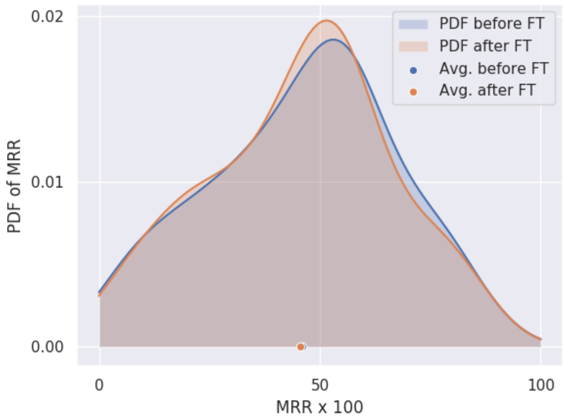

Figure 11: PDFs of the MRR

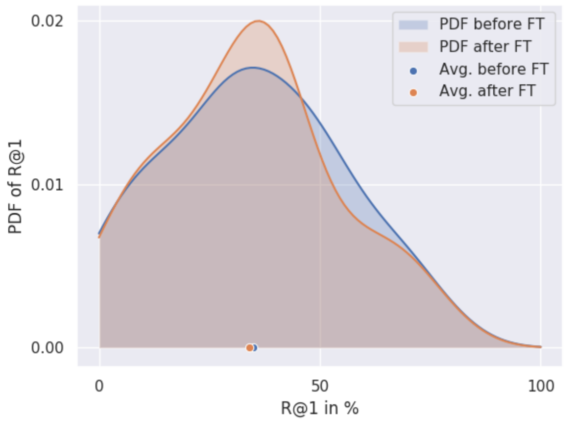

Figure 12: PDFs of the R@1

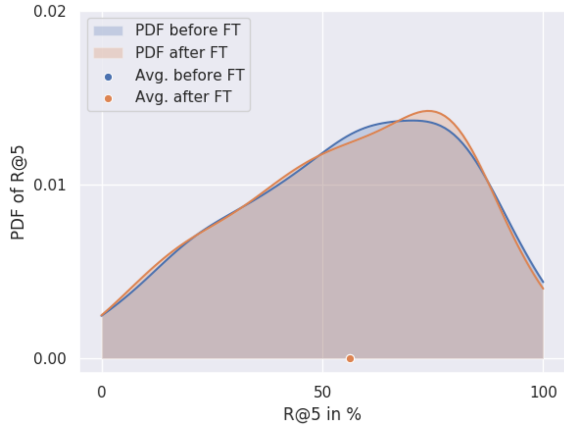

Figure 13: PDFs of the R@5

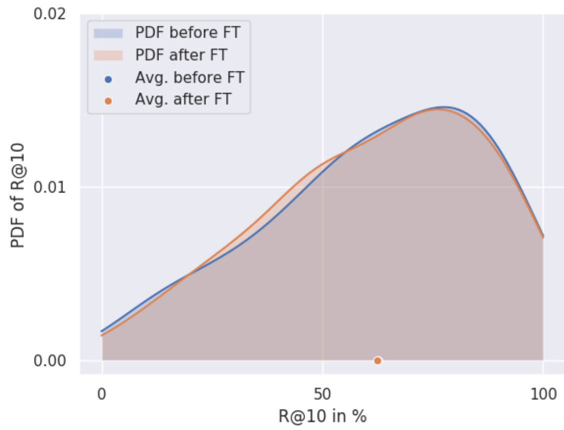

Figure 14: PDFs of the R@10

VII. Acknowledgment

At the end, we want to thank the Visual Dialog challenge creators who made huge efforts in collecting the data we used to pre-train our networks as well as providing good deep learning architectures capable of tackling such a challenging task [1].

References

[1] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[3] Joseph Walsh, Niall O' Mahony, Sean Campbell, Anderson Carvalho, Lenka Krpalkova, Gustavo Velasco-Hernandez, Suman Harapanahalli, and Daniel Riordan. Deep learning vs. traditional computer vision. 04 2019.

[4] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[5] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[6] R. Subash, R. Jebakumar, Yash Kamdar, and Nishit Bhatt. Automatic image captioning using convolution neural networks and LSTM. *Journal of Physics: Conference Series*, 1362:012096, nov 2019.

[7] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[8] Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. Recursive visual attention in visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6679–6688, 2019.

[9] Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] Dhruv Batra Devi Parikh Abhishek Das Vishvak Murahari, Prithvijit Chattopadhyay. Improving generative visual dialog by answering diverse questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[11] Kelvin Jiang, Dekun Wu, and Hui Jiang. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 318–323, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[12] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[13] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[14] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.