TOHOKU UNIVERSITY
GRADUATE SCHOOL OF INFORMATION SCIENCES

Learning Discriminative and Generative tasks for Visual Dialog

A thesis submitted for the Master's degree (Information Science)
Department of System Information Sciences

by
Nguyen Van Quang

April 22, 2020

Learning Discriminative and Generative tasks for Visual Dialog

by Nguyen Van Quang

Abstract

The field of artificial intelligence has a long-standing ambition which is to build an intelligent agent that has the capacity of perceiving and understanding the visual content in order to communicate with humans in natural language. Deep learning has recently become a game changer obtaining significant success in many computer vision and natural language processing tasks. Visual recognition, for instance, has gained exceptional progress in which a lot of deep neural networks even outperform human in many tasks such as classifying objects. Towards the long-standing goal of artificial intelligence, many efforts from research communities have been made to connect two domains i.e. natural language and vision such as image captioning and visual question answering. So, what lies next for AI field? One believes that the further step of AI is to develop an intelligent agent which could engage in a natural dialog with human about the rich world of visual content.

In this thesis, we pay our attention to visual dialog problem to investigate some existing approaches and propose a simple yet effective method which allows the agent to improve its performance and to become more applicable in reality. Previous works followed different kinds of attention with complicated mechanisms based on different hypotheses. We believe that there are many reasoning scenarios grounded question to image and dialog history such as question $\rightarrow$ image $\rightarrow$ dialog history, or question $\rightarrow$ dialog history $\rightarrow$ image $\rightarrow$ dialog history and so on. The simple and effective way is to compute the attended features between each other simultaneously in one reasoning step. Therefore, we propose a cross-attention mechanism in our method to learn the attended features based on dense co-attention. This allows us to localize the contribution of salient features in both language and visual content to the final results at word-level and sentence-level. Secondly, we propose a multi-task framework which simultaneously learns two tasks i.e. a discriminative part to rank the answer options and a generative part to generate an answer. While

most of the previous methods focused on learning a separate task, we argue that the multi-task approach exploits the synergy among different tasks, which helps improve their individual performances. The discriminative task allows the model to enjoy high accuracy of performance while the generative task allows the model to be more human-like in response generation automatically without any requirement from human-written answer inputs. Hereby, the appearance of the generative part makes it more applicable in the real-world situation.

Several experiments on Visdial dataset were conducted to evaluate the effectiveness of the proposed method. The experimental results reported on the standard benchmark metrics show that our proposed method achieves competitive performance compared with other state-of-the-art methods. The results also suggest that our model architecture possesses the effectiveness regardless of its simplicity. Unlike the other state-of-the-art methods with complicated architectures, the simple and homogeneous design brings several benefits to parallelize the computations.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my current supervisor, Assoc. Prof. Jinhee Chun for a lot of her support and advice during my two years at Tohoku University. I received the best of her laboratory to conduct the research, her encouragement and guidance to the completion of my dissertation. I am also grateful to Prof. Takeshi Tokuyama, my previous supervisor, who offered me a precious chance to become a member of the laboratory and to pursue the years of research at Tohoku University.

I would like to express my appreciation to my dissertation committee: Prof. Ayumi Shinohara, Prof. Kentaro Inui and Prof. Takayuki Okatani for their valuable feedbacks and advice in my first pre-defense, which are the insights of experts guiding me to the better dissertation.

I greatly appreciate Prof. Takayuki Okatani who kindly provides me the access to the powerful server of his laboratory. This indeed contributes a lot to the completion of Chapter 5 in my research. Thanks to his valuable suggestion and discussion, I could have some ideas to perform more comprehensive experiments for my current and future research on this thesis.

I value the time at Tokuyama-Chun laboratory, now Chun laboratory, where I set my first foot on academic research at Tohoku University. I also have many times taken the invaluable advice and technical supports from several members of Okatani-Suganuma Lab.

Special thanks to Japanese Government and Tohoku University for selecting me as a MEXT recipient to earn a scholarship during my two years at Tohoku Univerrsity.

My deepest appreciation goes to my sweetheart, Dinh Thu Hien. She has all provided companionship and encouragement, making this long journey more enjoyable and meaningful. Lastly, I acknowledge a forever gratitude to my parents and my siblings for their unconditional love and support throughout the writing of this thesis and my life in general. They are inspiring motivation for every single step I have been walking through life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Remarkable progress. Artificial Intelligence has witnessed many unprecedented achievements AI tasks from Natural Language Processing (NLP) to Computer Vision (CV). In particular, many computer vision tasks have seen tremendous successes such as image recognition to recognize thousands of objects [49], [15], scene classification [52] and object detection [44], [27]. Similar breakthroughs are visible in NLP such as machine translation [51] and language representation [10], [42]. Deep learning community has also made considerable progress in high-level tasks such as reading and comprehending short stories [16], [53], and even surpassed humans in playing Atari video games [35], and playing Go [48].

The intersection of language and vision. One of the visionary goals in the AI field is to enable machines to hold a meaningful conversation with humans about the rich visual world. Inspired by the successes of single modality tasks, deep learning community has made more effort in multi-modal tasks to put steps closer to the long-standing goal of AI. At the intersection of linguistic and visual worlds, several tasks from image captioning [28] [22] to visual question answering [2], [32], visual grounding [41] were proposed as an attempt to move from simple detection tasks towards tasks that involve a detailed understanding of the visual content. These tasks require a machine the capability of understanding and reasoning about the complex information of images and text.

1.1 What lies next for AI?

After the introduction of the multi-modal tasks in image captioning, visual question answering or visual grounding, one may question *what is the next task in the AI field?* Some believe that the next generation of an intelligent system is to develop a machine to understand and communicate with humans in natural and conversational language. Therefore, the visual dialog task has been quite recently introduced in [7], [9]. Visdial dataset released by [7] consists of free-form natural language questions and answers while GuessWhat?! dataset introduced in [9] provides more goal-specific dialogs which aim at object discovery in an image via a sequence of yes/no questions between two dialog agents.

The visual dialog task requires the understanding of image content associated with a dialog. This task encompasses multiple sub-problems from computer vision such as object detection, image classification and disambiguation between multiple occurrences in addition to tasks from the natural language community like language generation, co-reference resolution and summarization.

1.2 The motivation of Visual Dialog

The motivation behind Visual Dialog lies on the encouraging goal of AI to make an intelligent system with the capability of perceiving and understanding the vision and natural language. In particular, the applications of Visual Dialog are so practical in the real world including but not limited to the following:

1. **Applications in aiding impaired people.** An agent helps impaired users in understanding their visual world surrounding them [3] or the visual content on social media easily. Imagine an AI agent that involves in the following conversation with an impaired user on social network. An example: AI: "Peter has posted a picture from on Facebook during his holiday", User: "Great, what is he doing on the pic?", AI: "He has a party with his friends".

2. **Application in decision making.** An agent supports decision making based on a large amount of surveillance visual data [50]. An example: Manager: "Did anyone enter this room last week?", AI: "Yes, 27 instances logged on camera", Manager: "Did you see any suspicious?".

3. **Applications in virtual assistants.** Understanding visual content in conversation is an important feature of virtual AI assistants [36]. An example: User: "Hey Google, can you see the baby in the baby monitor?", AI: "Yes, I can", User: "Is he sleeping or playing?".

4. **Applications in robotics**. It helps build an autonomous agent [33] such as a robot for missions of searching and rescuing people in distress where the operator is inaccessible and prefers operating the robots via commands and conversation. An example: Human: "Is there smoke in any room around you?", AI: "Yes, in one room", Human: "Go there and look for people".

1.3 Our Objective and Contributions

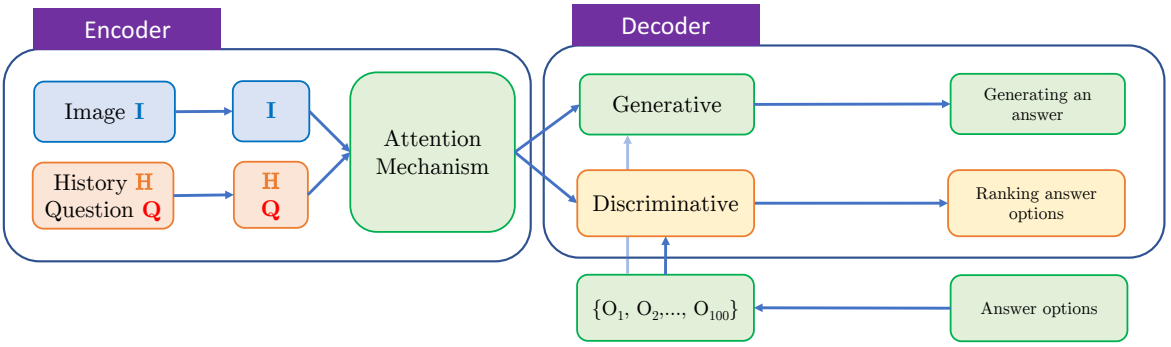
Most of the work in Visual Dialog fall into the category of data-driven techniques to solve specific tasks from a certain dataset. In this work, we principally tackle with a problem proposed in [7], Visdial challenge. Our aim is to develop a framework with a capability of ranking the answer options for the question in the dialog as well as generating an answer without specifying a set of answer options written by human.

The main objectives and contributions of this work are:

1. Investigate the Visdial challenge and several existing methods for this problem.

2. We propose a cross-attention mechanism in our method to learn the attended features based on dense co-attention. This allows us to localize the contribution of salient features in both language and visual content to the final results at word-level and sentence-level.

Figure 1-1: The overview of our proposed model



3. Besides recurrent neural networks such as LSTM, we introduce the adoption of transformer blocks to encode the text inputs rather than the adoption recurrent neural network in the existing methods.

4. Propose the end-to-end trainable framework learning two tasks simultaneously: (i) a discriminative task to rank the answer options and (ii) a generative task to generate an answer without the input list of human-written answers. We observe that training a multi-task framework allows both tasks to enjoy the synergy from each other to boost the overall performance. Moreover, our model possesses the capacity of generating an answer in addition to ranking the given answer options.

5. Perform several experiments to compare our models with several baselines and other state-of-the-art methods. The experimental results reported by standard evaluation protocol indicate that our model achieves performance on par with other state-of-the-art methods.

1.4 Thesis Outline

The rest of this work is presented as follows:

In **Chapter 2**, we provide relevant background about the Visdial task, the properties of Visdial problem as well as how to evaluate the performance based on several metrics. We also present how to obtain the representation of visual features from image source,

and linguistic features from history, caption, question, and answers. We describe the commonly standard feature representations provided by the strong baselines for image and text such as convolutional, recurrent neural networks as well as transformers.

In **Chapter 3**, we present some related works on the intersection of language and vision in general as well as the several previous works on Visdial task.

In **Chapter 4**, we propose a multi-task framework for Visdial that learns discriminative and generative tasks simultaneously.

In **Chapter 5**, we provide the details of experiments including dataset description, experiment settings, ablative studies, and experimental results.

In the **last Chapter**, we conclude our work, discuss some remaining challenges and provide some thoughts on the future work.

Chapter 2

Prelimilaries

In this chapter, the author will provide some background on the Visdial task specifically in the first section 2.1, the properties of Visdial problem in section 2.2.3, and several metrics for performance evaluation in section 2.3.

We also present some popular methods for processing image source in section 2.4, and text source from history, caption, question and answers in section 2.5. We describe the commonly standard feature representations provided by the strong baselines for image and text such as convolutional, recurrent neural networks as well as transformers.

2.1 Problem Formulation

The task in Visdial challenge [7] is formulated as follows: Given an image **I**, a history of a dialog **H** consisting of a sequence of question-answer pairs (For example: Q1: 'How many people are in wheelchairs?', A1: 'Two', Q2: 'What are their genders?', A2: 'One male and one female'), and a natural language follow-up question (Q3: 'Which one is holding a racket?'), the task for the machine is to find the most correct answer in (or to rank) the list of 100 answer options for this question (A3: 'The woman'). In Visdial task, the dialog history also consists of the caption of the image as default. Figure 2-1 demonstrates an example in Visdial dataset.

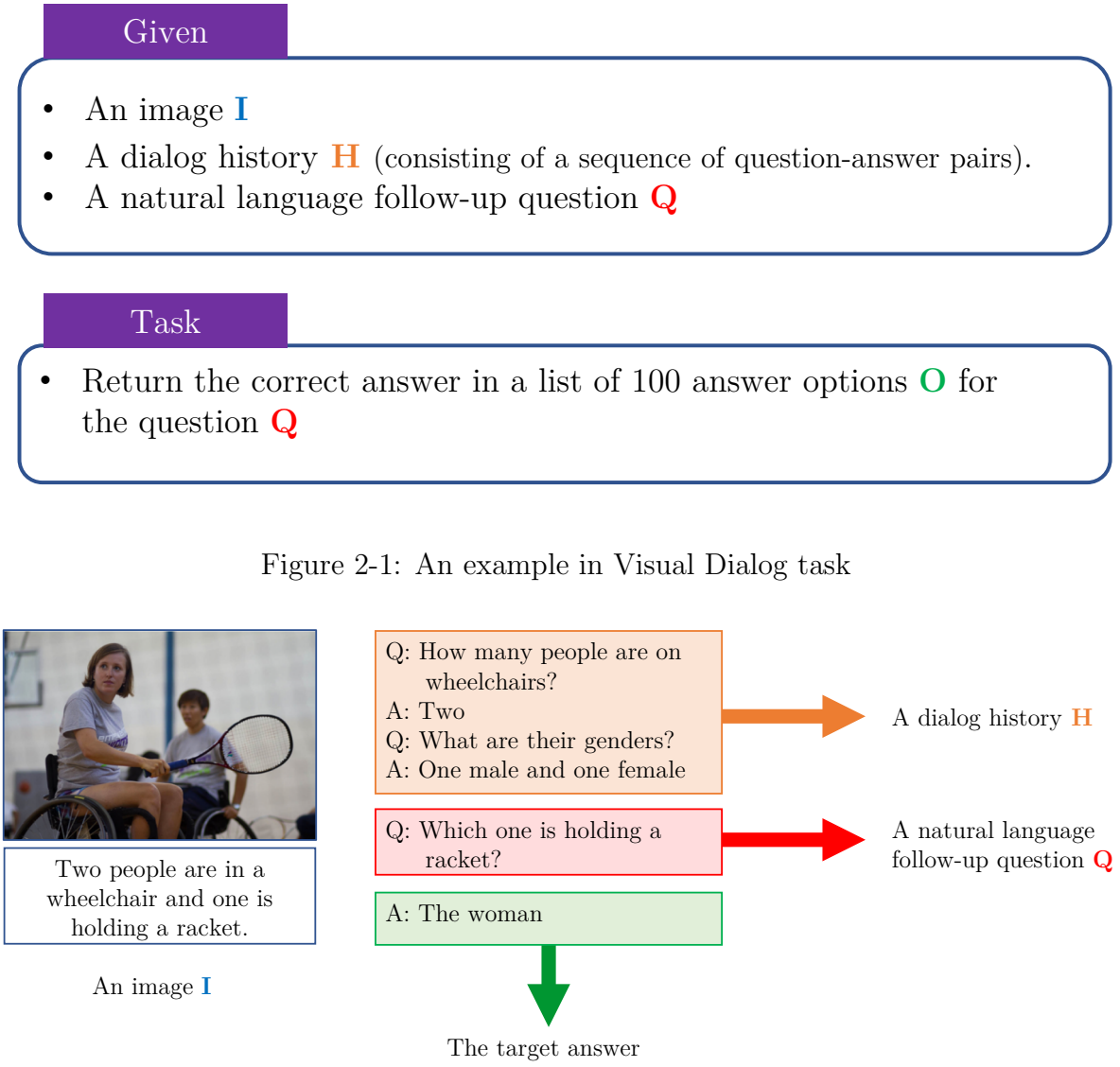


Figure 2-1: An example in Visual Dialog task

2.2 Visdial Dataset Analysis

Visdial dataset version 1.0 includes three splits:

1. **Training split.** There are total 123,287 images from COCO dataset. Each image has 10 rounds of question-answer. They all make up total of 1,232,870 pairs of question-answer.

2. **Validation split.** There are total 2,064 images from Flickr in which the author guarantee the distribution to be the same as COCO dataset. Each image has 10

rounds of question-answer. They all make up total of 20,640 pairs of question-answer.

3. **Test split.** There are total 8,000 images from Flickr dataset in which the authors guarantee the distribution to be the same as COCO dataset. Each image has N rounds of question-answer. The value N would be different.

Below we present the analysis of Visdial dataset provided by [7].

2.2.1 The properties of Visdial Questions

1. **Visual Priming Bias.** Visdial dataset was build to mitigate a 'visual priming bias' that existed in many previous works of image question-answering datasets such as VQA [2], Visual 7W [56]. In particular, the typical methodology to build the previous datasets is to allow both workers to see an image while asking questions about the image content. Several works showed that it leads to a visual priming bias in the questions, i.e. people have a tendency to only ask an obvious question that they can answer themselves. For example, one may ask 'Is there a clocktower in the picture?' on pictures actually containing clock towers. Language models without visual understanding might obtain remarkable performance on those VQA datasets and mislead the rapid progress inflatedly. It is worth mentioning that questions in VQA datasets usually start with 'Do you see a . . .'. Therefore, blindly answering 'yes' without understanding image content and the rest of the question could surprisingly result in the accuracy of 87% in those VQA datasets! Visdial dataset lowers the effect of visual priming bias by the simple constraint that questioners are not allowed to see the image.

2. **Distributions.** According to Figure 2-2, the most frequent lengths of questions in Visdial dataset range from 4 to 10, reaching a peak at 5. Figure 2-3 visualizes the distribution of question words in Visdial dataset versus VQA datasets based on the first four words. Besides some common properties shared by both Visdial and VQA datasets, there are several points to distinguish. Binary questions appear in Visdial

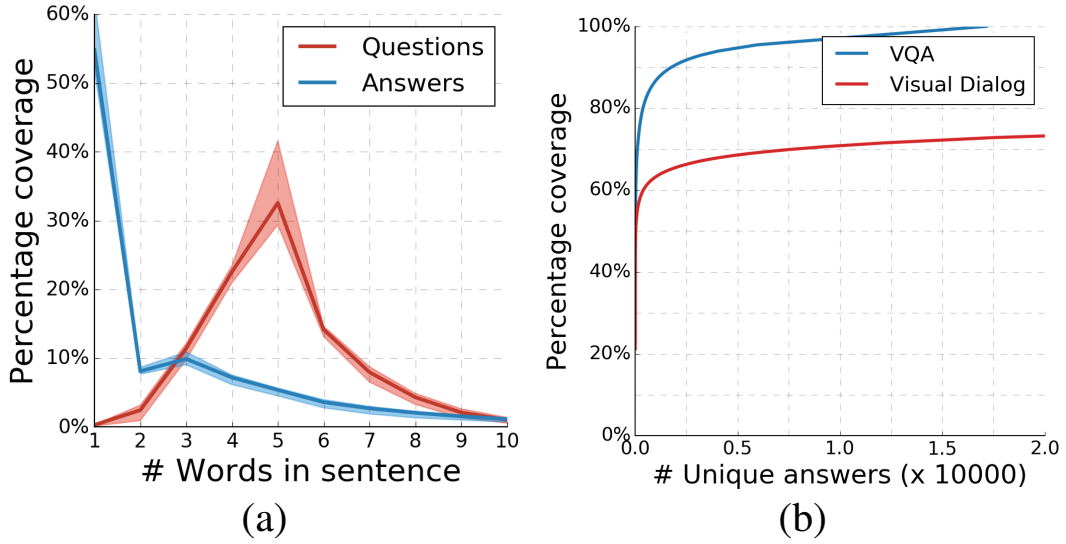


Figure 2-2: Distribution of lengths for questions and answers (left); and percent coverage of unique answers over all answers from the train dataset (right), compared to VQA. For a given coverage, Visdial has more unique answers indicating greater answer diversity. Source [7].

more frequently than VQA in which the most frequent first word of questions in Visdial dataset starts with 'is' versus 'what' in VQA.

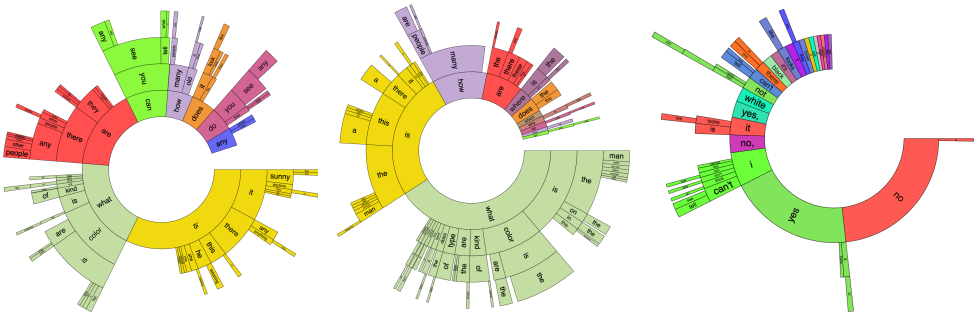
3. **Stylistic difference.** The questions in VQA datasets contain more specific details, often about the background such as 'What program is being utilized in the background on the computer?'. Meanwhile, in Visdial dataset questioners not allowed to view the image content tend to ask questions to obtain the mental sense about the image. Therefore, the questions in Visdial dataset tend to be open-ended, often being the following pattern:

   (a) Starting with the entities in the caption to ask the question: 'An elephant walking away from a pool in an exhibit', 'Is there only 1 elephant?',

   (b) Digging deeper into their parts or attributes: 'Is it full grown?', 'Is it facing the camera?',

   (c) The scene category or the picture setting: 'Is this indoors or outdoors?', 'Is this a zoo?',

   (d) The weather: 'Is it snowing?', 'Is it sunny?',



(a) VisDial Questions (b) VQA Questions (c) VisDial Answers

Figure 2-3: Distribution of first n-grams for (left to right) Visdial questions, VQA questions and Visdial answers. Word ordering starts towards the center and radiates outwards, and arc length is proportional to number of questions containing the word. Source [7].

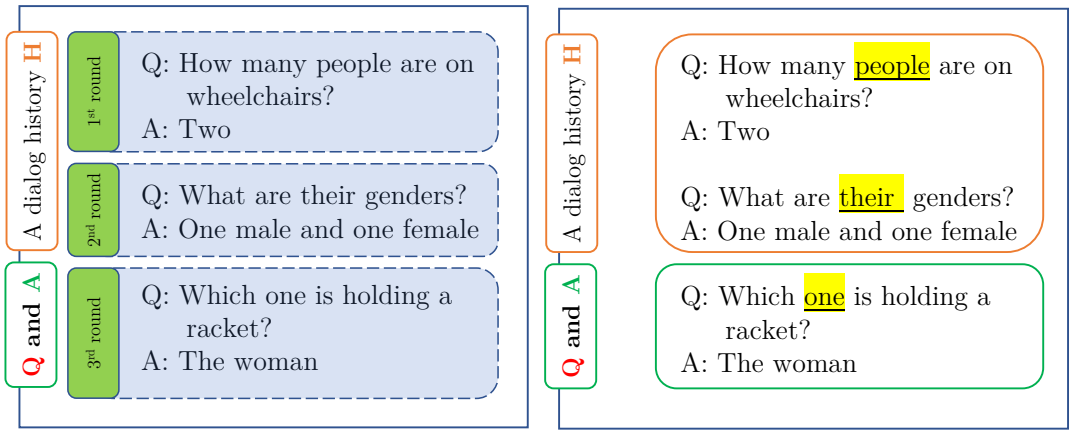
(e) Simply exploring the scene: 'Are there people?', 'Is there shelter for elephant?',

(f) And asking about the new visual entities discovered from these explorations: 'Thereś a blue fence in background, like an enclosure', 'Is the enclosure inside or outside?'.

2.2.2 The properties of Visdial Answers

1. **Answer Lengths.** Figure 2-2a shows the distribution of answer lengths which uncovers some differences between Visdial dataset and VQA datasets. First, answers in Visdial dataset have longer lengths and more descriptive than those in previous VQA datasets. Second, the coverage of all answers in Visdial dataset (about 63%) is lower than the top-1000 answers in VQA datasets (about 83%) of all answers. Most long answers in Visdial dataset are often unique.

2. **Answer Types.** Figure 2-3c visualizes the distribution of word frequence based on the first words. Interestingly, there are a lof answers in Visdial dataset which express doubt, uncertainty, or lack of information such as 'I think so', 'I can't tell', or 'I can't see'. This is mainly because that the questioners were not allowed to view the image - they tended to ask contextually relevant questions. Hereby, not all questions are answerable with certainty from the image content that was exposed



(a) A dialog consists of multiple rounds of question-answer pairs

(b) A dialog consists of the co-references among rounds

Figure 2-4: The properties of the visual dialog in Visdial task.

to answerers. The authors [7] argue that this is a rich dataset for building more human-like AI that refuses to answer questions it doesn't have enough information to answer.

3. **Binary Questions vs Binary Answers.** The difference between binary questions and binary answers in Visdial dataset and VQA datasets might be distinguished as follows. Binary questions are those starting in 'Do', 'Did', 'Have', 'Has', 'Is', 'Are', 'Was', 'Were', 'Can', 'Could'. Answers to such questions in Visdial dataset can (1) contain only 'yes' or 'no', (2) begin with 'yes', 'no', and contain some additional information or clarification, (3) involve ambiguity ('It's hard to see', 'Maybe'), or (4) answer the question without explicitly saying 'yes' or 'no' (Q: 'Is there any type of design or pattern on the cloth?', A: 'There are circles and lines on the cloth'). Binary answers in VQA datasets are biased towards 'yes' [6, 69] - 61.40% of yes/no answers are 'yes' while the distribution in Visdial dataset has about 46.96% 'yes'. It is mainly because that workers did not see the image, and were more prone to end up with negative responses.

2.2.3 The properties of Visdial Dialog

Consider an example in Visdial dataset shown in Fig. 2-1. The question 'What is the gender of the one in the white shirt?' requires the machine to selectively focus and direct attention to a relevant region. 'What is she doing?' requires co-reference (whom does the pronoun 'she' refer to?), 'Is that a man to her right?' further requires the machine to have a visual memory (which object in the image were we talking about?). Such machine also needs to be consistent with their outputs - Question: 'How many people are in wheelchairs?', Answer: 'Two', Question: 'What are their genders?', Answer: 'One male and one female' - note that the number of genders being specified should add up to two.

We present here the properties of visual dialog in Visdial task provided by [7] as follows:

1. **A dialog in Visdial dataset consists of multiple rounds**. All the images in the training set have the maximum of ten rounds. Each round is a pair of question and corresponding answer. The order of rounds in dialog should form a natural conversation with the consistency. Moreover, the follow-up question and its answer must be next to the last round in the dialog history. For example, the dialog in Fig. 2-4 (a) contains three rounds of question-answer. The follow-up question Q and answer A are in the round 3.

2. **There exist the co-references among the rounds in the dialog**. The language in Visdial dataset is the result of a sequential conversation, it naturally contains pronouns - 'he', 'she', 'his', 'her', 'it', 'their', 'they', 'this', 'that', 'those', so on. In total, 38% of questions, 19% of answers, and nearly all (98%) dialogs contain at least one pronoun. This property of Visdial dataset requires a model to have capacity of handling the ambiguity of words by referring them to their corresponding words in previous rounds of the dialog. For example, in Fig. 2-4(b), the pronoun word 'their' in the second round should be referred to its appropriate word 'people' in the first round. Then, we can perform the reasoning grounded in the appropriate regions of the image consisting of people. Another co-reference example is the word 'one' in the third round. Without co-reference, it can be considered a noun of cardinal

number or a symbol, or probably a pronoun of a person or thing that appears before. To determine exactly the meaning of the word 'one', the machine needs to find the co-reference from previous rounds in the dialog. Obviously, it must be co-referred to the word 'their' and 'people' in the previous rounds, then grounding in the corresponding visual regions in the image.

3. **Temporal Continuity in Dialog Topics**. It is natural for conversational dialog data to have continuity in the 'topics' being discussed. Across 10 rounds, questions in Visdial dataset are not all independent questions. There is more continuity in Visdial dataset because questions do not change topics as often as VQA datasets.

4. **The reasoning is grounded from the question to image and history**. For example, a question of "How many people are there in the image?" should yield a short reasoning sequence like: *question*(the word 'people') $\rightarrow$ *image*(regions of people). However, a question of "Is there anything else on the table" should result in a long reasoning sequence including several steps such as *question* (the word 'table') $\rightarrow$ *image* (regions of table) $\rightarrow$ *question* (the word 'else') $\rightarrow$ history (context for 'else').

Such difficulties make Visdial challenge highly interesting and challenging.

2.3 Evaluation metrics

Evaluating the performance in dialog systems or captioning and machine translation still remains challenging due to the free-form nature of answers. Several proposed metrics such as BLEU, METEOR, ROUGE are claimed to correlate poorly with human judgment in evaluating the answers generated by dialog systems [29].

Handling this open problem, [7] proposed several deterministic metrics which evaluate individual answers at each round ($t = 1, 2, \ldots, 10$) with ranking multiple choices instead of evaluating on a downstream task [4] or holistically evaluating the entire conversation in the dialog.

Given an image **I**, the dialog history **H** (including the image caption C), $(Q_1, A_1), \ldots, (Q_{t-1}, A_{t-1})$, the question Q_t, and a list of $N = 100$ answer options, the system is asked to return a ranking of the answer options. Then it will be evaluated by several metrics used in Visdial challenge summarized in the below table in which ↑ means higher is better and ↓ means lower is better.

Table 2.1: The evaluation metrics in Visdial challenge.

Metric	**Description**	**Score**
Mean	The rank of human response	↓
Recall@k	The existence of the human response in the top **k**	↑
MRR	The mean reciprocal rank of the human response	↑
NDCG	The normalized discounted cumulative gain over the top **k**	↑

Since 2018, **NDCG** is the major metric to evaluate the performance of the system. For this computation, we consider the relevance of an answer to be the fraction of annotators that mark it as correct. **NDCG** is invariant to the order of options with identical relevance and to the order of options outside of the top **k**. For example, given five answer options with ranking (from high to low): ["yes", "yes it is", "probably", "two", "no"], their corresponding ground truth relevance scores: [1, 1, 0.5, 0, 0] such that the number of options with non-zero relevance scores is 3 and NDCG remains unchanged for two cases in the tables below:

1. Swapping options with same relevance.

2. Shuffling options after first**k** indices.

Table 2.2: Swapping options with same relevance.

Example	**Ranking**	**NDCG**
1	["yes", "yes it is", "two", "probably", "no"]	0.8670
2	["yes it is", "yes", "two", "probably", "no"]	0.8670

The evaluation protocol is compatible with both discriminative models (that simply score the input candidates, e.g. via a softmax over the options, and cannot generate new

Table 2.3: Shuffling options after first **K** indices.

Example	**Ranking**	**NDCG**
1	[“yes”, “two”, “yes it is”, “probably”, “no”]	0.7974
2	[“yes”, “two”, “yes it is”, “no”, “probably”]	0.7974

answers), and generative models (that generate an answer string, e.g. via Recurrent Neural Networks) by ranking the candidates by the model's log-likelihood scores.

Answer options. To generate 100 answer options, they first included the ground truth answer, together with some plausible choices. In addition, they also added 30 most popular answers to the list. The random answers were added up until the total answer options are 100. For types of answer options are described as below:

1. **Correct:** The ground-truth human-written answer to the question.
2. **Plausible:** Answers to 50 most similar questions. Similar questions are those that start with similar tri-grams and mention similar semantic concepts in the rest of the question. To capture this, all questions are embedded into a vector space by concatenating the GloVe embeddings of the first three words with the averaged GloVe embeddings of the remaining words in the questions. Euclidean distances are used to compute neighbors. Since these neighboring questions were asked on different images, their answers serve as ‘hard negatives’.
3. **Popular:** The 30 most popular answers from the dataset e.g. ‘yes’, ‘no’, ‘2’, ‘1’, ‘white’, ‘3’, ‘grey’, ‘gray’, ‘4’, ‘yes it is’. The inclusion of popular answers forces the machine to pick between likely a priori responses and plausible responses for the question, thus increasing the task difficulty.
4. **Random:** The remaining are answers to random questions in the dataset.

2.4 Image Representation

A Convolutional Neural Network (CNN, or ConvNet) are a special kind of multi-layer neural networks for handling data with some spatial topology (e.g. images, videos, sound spectrograms in speech processing etc). Many computer vision problems such as image recognition, object detection have adopted CNN models to obtained the state-of-the-art results. For the legacy use of CNN networks to extract the features from images in previous work in Visual Dialog, we describe how to extract image features from VGG16 [49], and the bottom-up features from Faster R-CNN [44]. In our proposed method, we use bottom-up features extracted from Faster R-CNN as feature representation for images while features extracted from VGG16 are used as the baseline for ablative studies.

2.4.1 Feature Extraction from VGG16

VGG is a convolutional neural network model for image recognition which was first introduced to in the competition ImageNet Challenge 2014, winning the first and the second places in the localisation and classification tracks respectively. Figure 2-5 illustrates the architecture of VGG16 with the input layer takes an image in the size of $(224 \times 224 \times 3)$, and the output layer is a softmax prediction on 1000 classes.

Some interesting points from VGG work can be summarized as below:

1. The network is considered as very deep at its time. Several VGG architectures were introduced such as VGG13, VGG16 and VGG19 with 13, 16, or 19 layers respectively.

2. The architecture is extremely simplified with only 3×3 convolutional layers and 2×2 max-pooling layers.

3. The stack of small filters with fewer number of parameters can replace a larger. For example, the filter of size 7×7 of 49 parameters will replace the stack of two 3×3 filters of 27 parameters. The number of parameters is reduced by 45%.

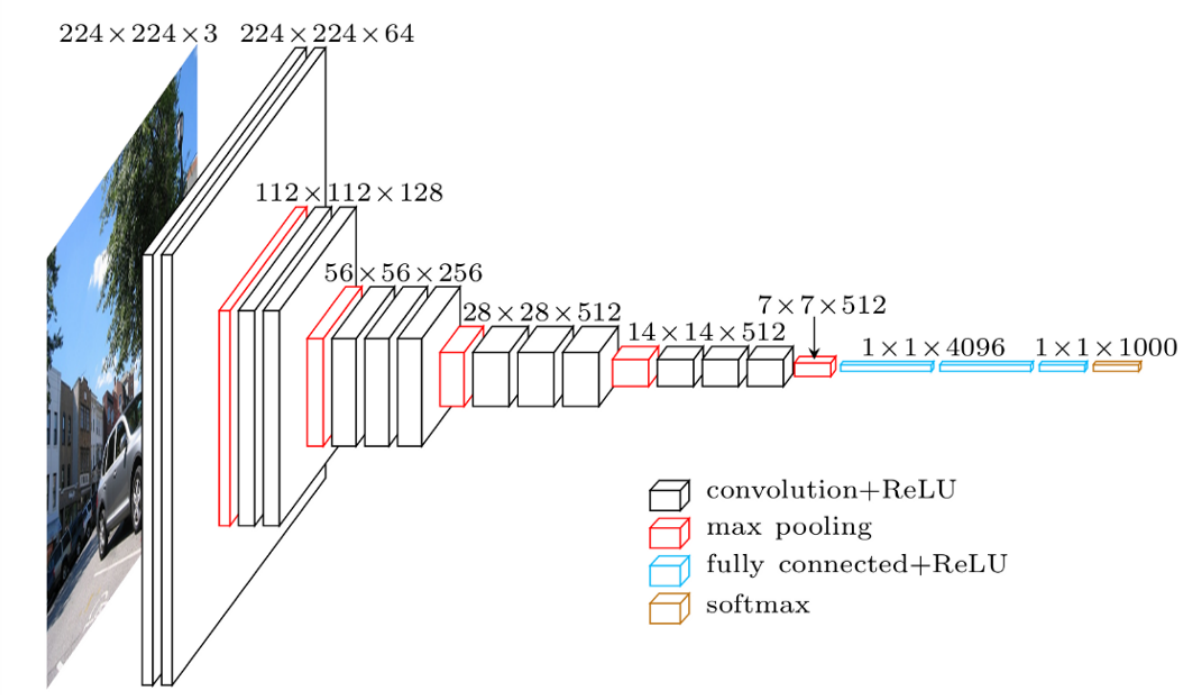


Figure 2-5: Illustration of the architecture of VGG16: the input layer takes an image in the size of $(224 \times 224 \times 3)$, and the output layer is a softmax prediction on 1000 classes. From the input layer to the last max pooling layer (labeled by $7 \times 7 \times 512$) is regarded as the feature extraction part of the model, while the rest of the network is regarded as the classification part of the model.

Feature extraction for Visual Dialog. For the legacy use of VGG for feature extraction in previous work in Visdial, we describe in short how to extract feature representation for images. As the input of VGG is a tensor of an image with the size of $224 \times 224 \times 3$, we forward it through the pretrained network VGG16 in ImageNet dataset. From the input layer to the last max pooling layer (labeled by $7 \times 7 \times 512$) is regarded as the feature extraction part of the model, while the rest of the network is regarded as the classification part of the model. Therefore, we extract the feature maps $7 \times 7 \times 512$ in the last max pooling layer. In other words, we will obtain a tensor of N$\times$512 as feature representation for the image where N$= 7 \times 7$.

2.4.2 Feature Extraction from Faster R-CNN

Faster R-CNN [44] is a convolutional neural network model for object detection which first introduced a Region Proposal Network (RPN) that shares full-image convolutional

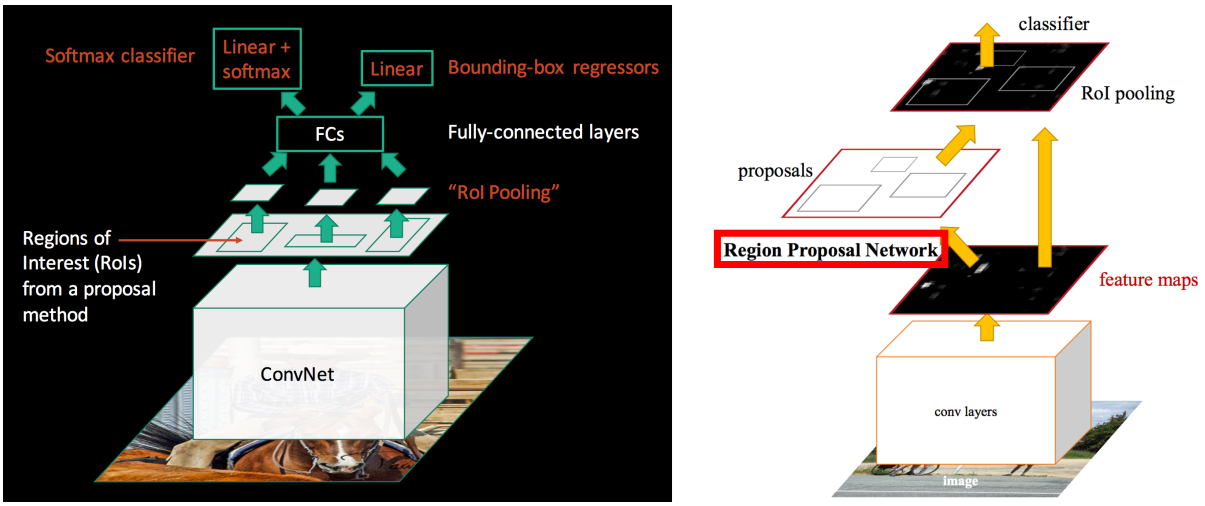


Figure 2-6: An illustration of Faster R-CNN model. Source: [44]

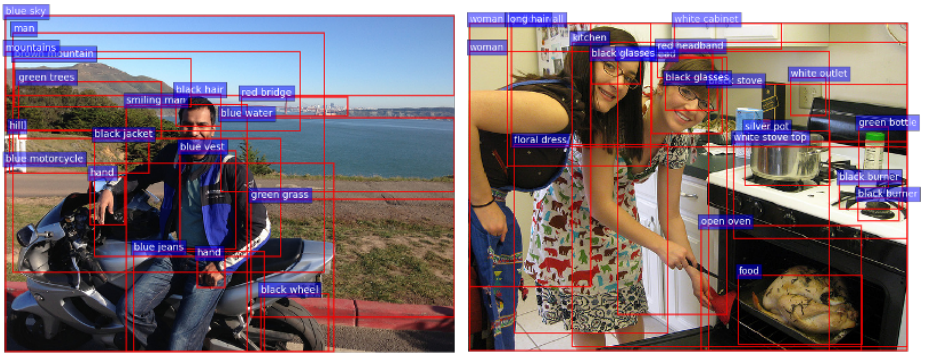
features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully-convolutional network that predicts simultaneously object bounds and objectness scores at each position. The regions with highest objectness scores will be extracted by RoIAlign and be forwarded to the classifier in the second stage. This classifier is responsible for generating the fine-grained class scores and bounding boxes.

The training procedure of Faster R-CNN for object detection is summarized as below:

1. We first pre-train a CNN network on object recognition with large dataset ImageNet.

2. In the first stage, we fine-tune the RPN (region proposal network) with the weights initialized by the pre-trained object classifier. The network slides a small $n \times n$ spatial window over the feature maps, and predict the objectiveness scores and the bounding boxes for multiple regions. Each position has a multiple anchors with different scales and ratios. For objectiveness class, the positive regions have Intersection over Union (IoU) $> \alpha_1$ (usually 0.7) while the negative regions have IoU $< \alpha_2$ (usually 0.3).

3. In the second stage, we train a Fast R-CNN object detection network (a classifier) using the proposal regions generated by the current RPN. Each region is extracted by RoIAlign and forwarded to the classifier to predict the fine-grained class scores as well as the respective bounding box.



(a) (b)

Figure 2-7: The 36 region proposals generated from Faster R-CNN with highest class-agnostic probability. Source [1].

Feature extraction for Visual Dialog. Since 2018, Anderson et al. [1] proposed using image regions, each with an associated feature vector, as the feature representation for images in Image captioning and VQA tasks. These so-called bottom-up features enable attention to be calculated at the level of objects and other salient image regions. Since 2018, Visdial challenge has started using this method to extract features for image representation. The feature vectors associated with the top 36 regions with the highest class-agnostic scores generated by RPN in the first stage of Faster R-CNN will be used as final features. In other words, we will obtain a tensor of 36×2048 as feature representation for the image.

2.5 Language Representation

In Visdial task, dialog history, questions and answers are a sequence of words $(\mathbf{w_1}, \ldots, \mathbf{w_T})$, in which each word is encoded as a one-hot vector that is essentially a sparse vector where only one entry has the value of 1. We would find another representation for words and sequences that can reveal some interesting relationships among each other. For example, **vector("cat") - vector("kitten")** is similar to **vector("dog") - vector("puppy")**. We will summarize some methods for learning the representation for words and sequences, specifically using Bi-LSTM in subsection 2.5.1, and using transformer encoder in subsection 2.5.2.

2.5.1 Learning representation with Recurrent Neural Networks

A recurrent neural network (RNN) [45] is a specific kind of neural network for processing a sequence of vectors $(\mathbf{x_1}, \ldots, \mathbf{x_T})$ using a recurrence formula of the form:

$$\mathbf{h_t} = f_\theta(\mathbf{h_{t-1}}, \mathbf{x_t}) \tag{2.1}$$

where f is a function that we describe in more details below and the same parameters θ are used at every time step, allowing us to process sequences with arbitrary lengths. The hidden vector $\mathbf{h_t}$ can be interpreted as a running summary of all vectors $\mathbf{x}$ until that timestep and the recurrence formula updates the summary based on the next vector. The hidden vector at timestep t in **Vanilla Recurrent Neural Network** can be computed as follows:

$$\mathbf{h_t} = \tanh(\mathbf{W_{xh}}\mathbf{x_t} + \mathbf{W_{hh}}\mathbf{h_{t-1}}) \tag{2.2}$$

Where $\mathbf{W_{xh}}$, $\mathbf{W_{hh}}$ are the parameters, and $tanh$ is a hyperbolic activation function. These equations 2.1, 2.2 omit the additional bias vector for brevity.

The undesirable nature of vanilla RNNs is that the gradients tend to either vanish or explode over long time periods. To handle the limitations of vanilla RNNs, Long Short-Term Memory (LSTM) was introduced in [17]. Its recurrence formula has a form that allows the inputs $\mathbf{x_t}$ and $\mathbf{h_{t-1}}$ interact in a more computationally complex manner that includes multiplicative interactions, and the LSTM recurrence uses additive interactions over time steps that more effectively propagate gradients backwards in time. In addition to a hidden state vector $\mathbf{h_t}$, LSTMs also maintain a memory vector $\mathbf{c_t}$. At each timestep, LSTM can choose to read from, write to, or reset the cell using explicit gating mechanisms as the illustration in Figure. 2-8.

Bidirectional Long Short Term Memory (Bi-LSTM) is a specific kind of LSTM which consists of one LSTM that processes the sequence from left to right and one that processes the sequence from right to left. In the forward direction one LSTM will process the sequence to return a sequence of hidden vectors $(\vec{\mathbf{h_1}}, \vec{\mathbf{h_2}}, \ldots, \vec{\mathbf{h_T}})$ with the input $(\mathbf{x_1}, \mathbf{x_2}, \ldots, x_T)$. Meanwhile in reverse direction another LSTM process the sequence $(\mathbf{x_T}, \mathbf{x_{T-1}}, \ldots, \mathbf{x_1})$ return a sequence of hidden vectors $(\overleftarrow{\mathbf{h_T}}, \overleftarrow{\mathbf{h_{T-1}}}, \ldots, \overleftarrow{\mathbf{h_1}})$. The concatenation of $(\vec{\mathbf{h_i}}, \overleftarrow{\mathbf{h_i}})$ as $\mathbf{h_i}$ will be the final hidden vector for the input vector $\mathbf{x_i}$.

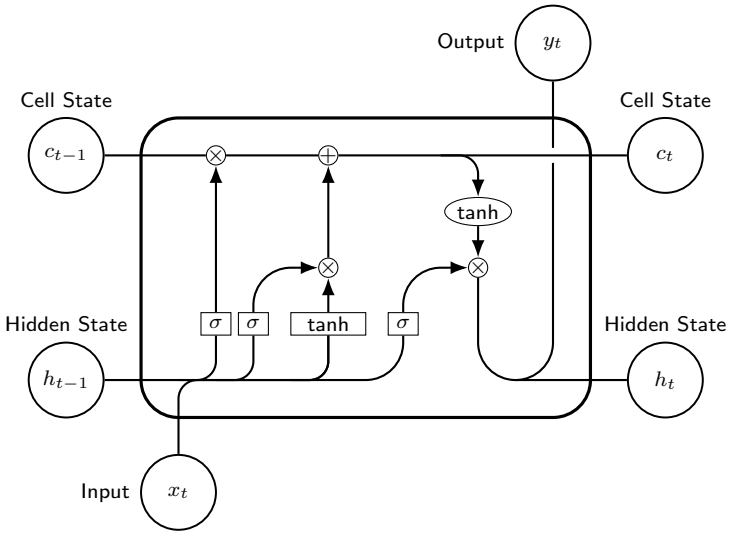


Figure 2-8: Illustration of Long short term memory cell.

Representation for words and sequences. Most of the works in Visdial using a stack of several LSTM or Bi-LSTM layers to learn the feature representation for text. The hidden vector $\mathbf{h_i}$ would be the feature representation for the one-hot vector $\mathbf{x_i}$ of the i^{th} word (or token) in the sequence $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T})$. Meanwhile, we can use the final hidden vector $\mathbf{h_T}$ in LSTM or the concatenation of $(\overrightarrow{\mathbf{h_T}}, \overleftarrow{\mathbf{h_1}})$ in Bi-LSTM as the feature representation for the whole sequence $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T})$.

2.5.2 Learning representation with Transformers

Transformer [51], a novel neural network architecture based on a self-attention mechanism was introduced first for the translation task in NLP. In several NLP tasks, transformers show superiority in performance while less inference time compared with recurrent neural networks.

Attention. An attention function requires a query of dimension d_k and a set of key-value pairs of dimension d_v. The function computes the output which is a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The function computes the dot products of the query with all keys, divide each by $\sqrt{d_k}$, then applies a softmax function to obtain the

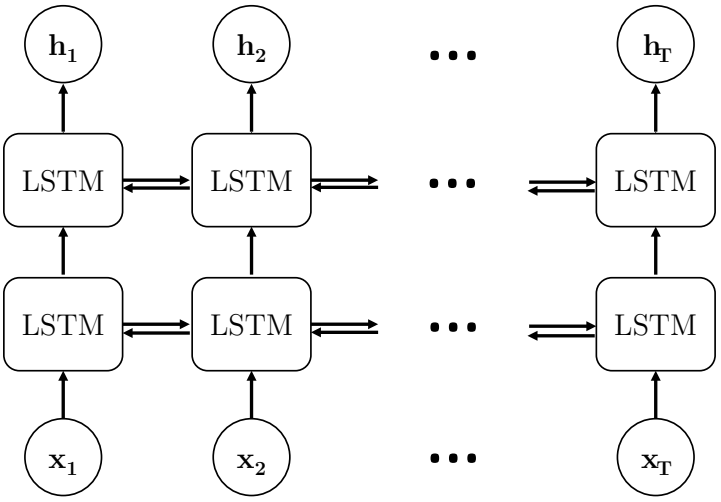


Figure 2-9: Illustration of processing a sequence with a stack of Bi-LSTM layers.

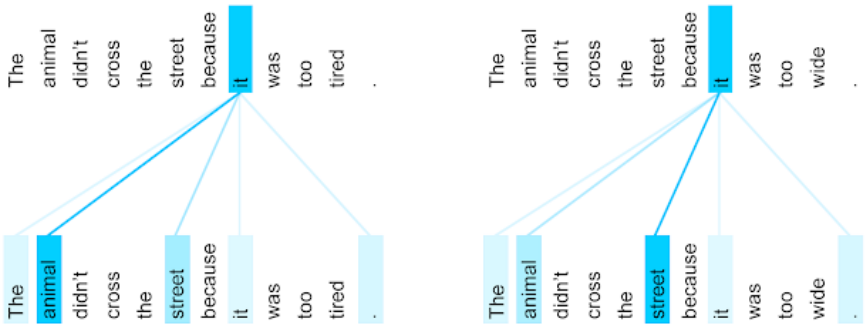


Figure 2-10: The encoder self-attention distribution for the word "it" from the 5^{th} to the 6^{th} layer of a Transformer trained on English to French translation (one of eight attention heads). Source: [51].

weights on the values. The function is presented as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{2.3}$$

To learn multiple attended features from different spaces at different position, transformers utilize multi-head attention mechanism as described in Figure 2-11.

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(head_1, \ldots, head_h)\mathbf{W^O} \tag{2.4}$$

where $head_i = \text{Attention}(\mathbf{QW_i^Q}, \mathbf{KW_i^K}, \mathbf{VW_i^V})$, and the projections are parameter matrices $\mathbf{W_i^Q} \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W_i^K} \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W_i^V} \in \mathbb{R}^{d_m \times d_v}$, and $\mathbf{W_i^O} \in \mathbb{R}^{d_m \times d_v}$.

Position-wise Feed-Forward Networks. Besides the self-attention module, the transformer layer also consists a fully connected feed-forward network, applied to each position

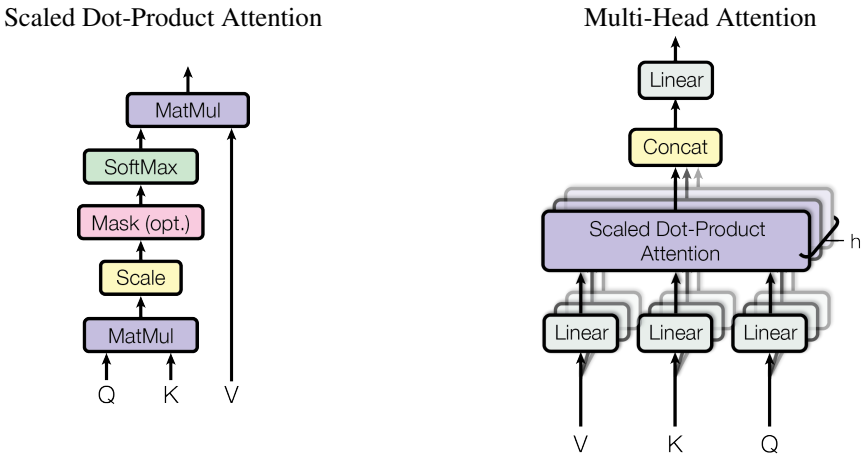


Figure 2-11: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. Source: [51].

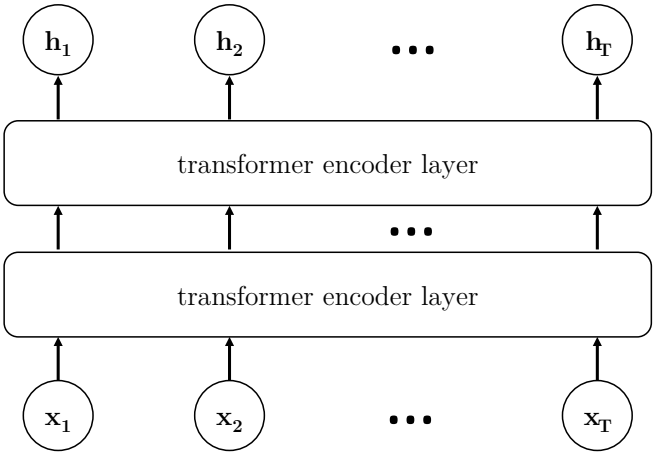


Figure 2-12: Illustration of using transformer encoder of two transformer encoder layers to process the sequence and obtain its feature representation.

separately and identically. The network includes two linear transformations with a ReLU activation in between. In the paper, the dimension of input and output is $d_m = 512$, and the inner-layer has dimension $d_{ff} = 2048$.

$$\text{FFN}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W_1} + b_1)\mathbf{W_2} + b_2 \tag{2.5}$$

Positional embeddings. The embeddings of tokens also include the "positional embeddings" to the information about the relative or absolute position of the tokens in the sequence. The positional embeddings adopt sine and cosine functions of different

frequencies as follows:

$$P_E(pos, 2i) = sin(pos/10000^{2i/d_m}) \tag{2.6}$$

$$P_E(pos, 2i+1) = cos(pos/10000^{2i/d_m}) \tag{2.7}$$

The transformer encoder. The encoder consists of a stack of transformer layers each of which includes self-attention layer and feed-forward layer. The input of a transformer layer is the output of previous transformer layers.

Representation for words or tokens. As depicted in Figure 2-12 We take feature representations for words or tokens as the output sequence of vectors $(\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_T})$ in which output vector $\mathbf{h_i}$ would be the feature representation for the embedding vector $\mathbf{x_i}$ of the i^{th} word (or token) in the sequence $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T})$.

Chapter 3

Related Work

In this chapter, the thesis will first recap some works of two problems closely related to Visual Dialog at the intersection of vision and language understanding, namely, Image Captioning in subsection 3.1.1, and VQA in subsection 3.1.2. Later, we will provide some literature highlights of several outstanding works on Visdial challenge in section 3.2.

3.1 The Intersection of Vision and Language

The tasks involving visual and language understanding such as image captioning [6] and visual question answering (VQA) [12] have witnessed a rapid progress to inspire tremendous efforts at the boundary of computer vision and natural language processing.

3.1.1 Image Captioning

The concrete task of image captioning is the following: given the image **I**, the task is to generate the caption **C** that describes the content of the image **I**. The human-machine interaction only happens in one direction from machine to human. For example, given the image in Figure 3-1, the machine is expected to generate the caption "Two people are in a wheelchair and one is holding a racket" or others with similar meaning.

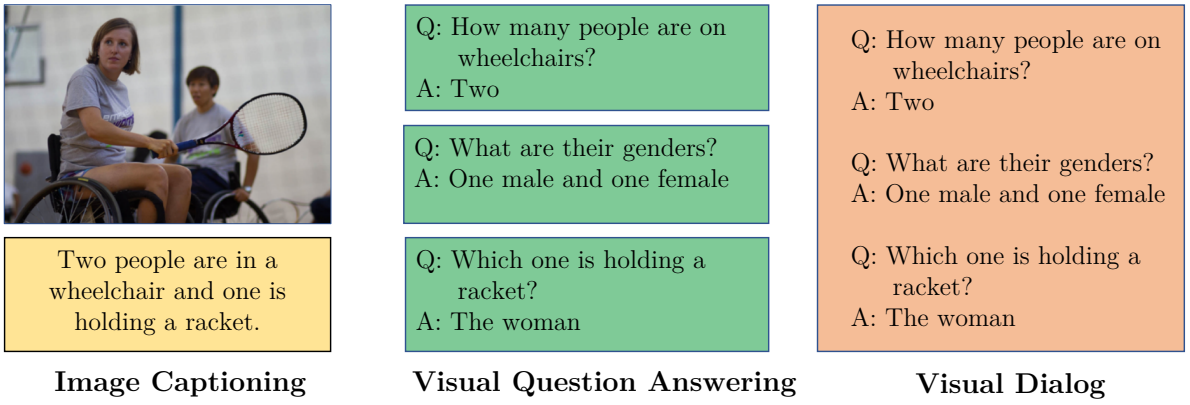


Figure 3-1: Illustration of differences among three tasks: (left) Image Captioning task, (middle) VQA task, (right) Visual Dialog task.

A lot of large-scale datasets have released for the few years such as Visual Genome [26], Flickr30k [41], MSCOCO [28]. Moreover, different kinds of metrics including BLEU, ROUGE, METEOR, CIDEr, SPICE were proposed to evaluate the performance of image captioning systems.

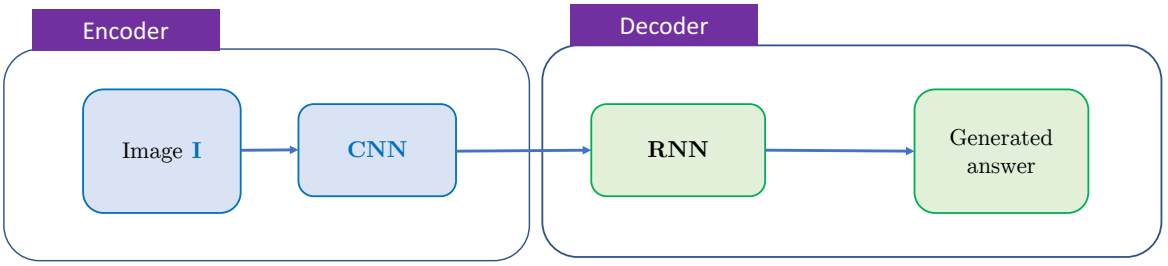


Figure 3-2: Illustration of a simple model for image captioning: the encoder using CNN to process an image into embedding vector, the decoder using a RNN to process the embedding vector to generate the caption for the image.

Unsurprisingly, most of state-of-the-art results have been obtained by deep learning based methods. Deep learning-based methods possess the capacity of handling the complexities and challenges of image captioning. Hossain et al. [18] released a well-written survey on Image Captioning as depicted in Figure 3-3.

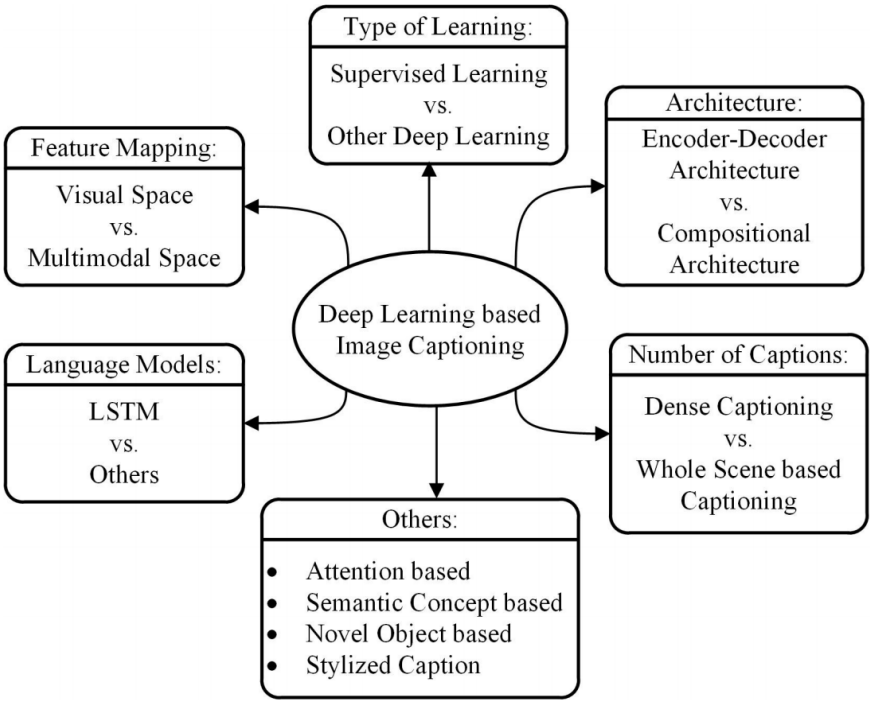


Figure 3-3: An overall taxonomy of deep learning-based image captioning. Source: [18]

3.1.2 Visual Question Answering

The concrete task of visual question answering is the following: given the image **I**, a question **Q** about that image the task is to answer the question correctly. The answer could be in any of the following forms: a word, a phrase, a yes/no answer, choosing out of several possible answers, or a fill in the blank answer. The human-machine interaction involves both human and machine. Human asks machine the question about the visual content while machine must return the answer to this question. It can be considered as a single round of a dialog which only consists of one pair question-answer. For example, in Figure 3-1 Human: 'How many people are on wheelchairs?', machine is expected to answer: 'Two'. The next question from human: 'What are their genders?', machine: 'One male and one female'. This pair of question and answer is independent of the previous round. It may be hard for the machine to decode the meaning of word 'their' in the question since there is no information about the first round of dialog to refer.

There have been several large datasets released for VQA task including the most popular: DAQUAR [32], Visual7W [56], COCO-QA [43], VQA (COCO) [2], etc. Following

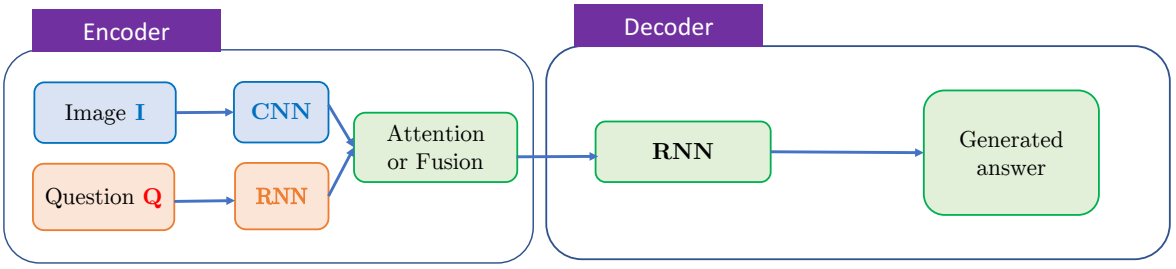


Figure 3-4: Illustration of a simple model for VQA task: the encoder using CNN to process an image into image features and using RNN to process the question into text features. The last module of encoder is responsible for fusing or using attention mechanism to aggregrate image and text features, the decoder using a RNN to process the output vector of the encoder to generate the answer for the question.

the survey [14], deep learning approaches to VQA tasks can be categorized into three main methods: non-attention approaches, modular networks, and attention-based models. Some typical works belonging to non-attention approaches include Norm I+ deeper LSTM [2], Learning by Asking Questions [34]. Modular networks use the modular design of the network to address VQA task such as Neural Module Networks [2], End-to-End Module Networks [19], ReasonNet [20]. In the third approach, there are a number of attention-based architectures in the literature as well. These methods are based on generating spatial maps to highlight image regions that are relevant to answering the question. Some typical works include Where to Look [47], Hierarchical Question-Image Co-Attention [31], or Dense Symmetric Co-Attention [37], etc.

3.2 Previous work on Visual Dialog

Visual Dialog task has been recently proposed by [7] and [9]. Specifically, [7] released the Visdial dataset, which contains free-form natural language questions and answers. And [9] introduced the GuessWhat?! dataset, where the dialogs provided are more goal-oriented and aimed at object discovery within an image, through a series of yes/no questions between two dialog agents. For the Visdial task, a typical visual dialog system follows the encoder-decoder framework proposed in [52]. Based on different aspects, the categories of

methods to Visdial task may be classified as below.

3.2.1 Encoders

The encoder of a model usually computes and processes the input triplets including image features, dialog history features and question features into a single or multiple context vectors. The output of the encoder will be then forwarded to the decoder of the model. There have been a wide variety of proposed encoder models including

1. **Late fusion** encoder [7] which uses a CNN network to extract an image feature, and recurrent networks to extract a single feature vector for history and question. Later, all three features are fused into a single vector and projected into a context vector of lower dimensions.

2. **Hierarchical recurrent network** encoder [7] which includes a dialog-level Recurrent Neural Network (RNN) for question-answer (QA) turn in the dialog history. In each QA-level recurrent block, the network uses attention-over-history mechanism to choose and attend to the round of the history relevant to the current question.

3. **Memory network** encoder [7] which considers each previous QA pair in dialog history turn as a fact in its memory. The network is trained to learn to pull the appropriate facts and the image for the current question to generate a context vector.

4. **Attention network** encoders which comprise most of the recently proposed methods such as sequential co-attention [54], history-conditional image attention [30], etc. In the encoder, there are different kinds of attention mechanism adopted to learn the attended features among each other.

3.2.2 Decoders

In the encoder-decoder architecture, the decoder usually receives the context vector from the encoder, as well as the input candidate answers (in discriminative cases). The output

of the model depends on the category of its decoder which may fall into several categories as below:

1. **Generative decoder** to generate the answer with a recurrent neural network. During the training, the decoder learns to maximize the log-likelihood of words in the ground truth answer for the question. In the inference, the answer can be generated by predicting the next word based on the previous word prediction, in which the first input word is usually the start-of-sentence (<SOS>) token. The inference procedure will end when meeting any of two criteria: (i) the end-of-sentence (<EOS>) appears, (ii) the maximum length reaches. This decoder is preferred in terms of its capacity of generating the answer for an question itself without any human-written input.

2. **Discriminative decoder** to rank the candidate answers for the current question. The discriminative decoder learns to minimize softmax-based cross-entropy loss [7] or a ranking-based multi-class N-pair loss [30]. The output of decoder is the probabilities which indicate the ranking of all candidate answers. Discriminative decoders usually lead to better scores **r@k** and **mean** than the generative decoders.

3. **Both generative and the discriminative decoder** in the model. While all of the proposed methods using either a discriminative decoder or generative decoder in the model, training a model for two tasks is a promising idea to exploit the synergy of two decoders. The benefit is not only from the synergy of two decoders but also the simplicity and effectiveness of a single encoder shared by two decoders. In fact, this thesis proposes a multi-task model for visual dialog system.

3.2.3 Model Architectures

The most popular architecture for visual dialog problem is **codec**, i.e. encoder-decoder architecture. Besides several basic architectures based on the **codec** architecture, we would like to recap here some interesting architectures of state-of-the-art proposed methods as following:

1. **Graph Neural Network** (GNN) [55]. The authors consider relations in dialog as an unknown graph structures, a GNN is trained to learn the underlying semantic relationship between dialog entities including the question, the history dialog in each turn. The expectation maximization algorithm is designed to approximate the inference of edge weights and the values of unobserved nodes in the graph.

2. **Neural Module Networks** [24]. The authors introduced two neural modules namely Refer and Exclude that can perform the co-reference at a finer word level to the entities in the image and dialog history rather than the traditional sentence-level.

3. **Two-stage Synergy Network** [13]. Unlike most methods adopting a one-stage architecture, the authors proposed a two-stage architecture which in the first stage, learns to rank the candidate answers coarsely based on the relevance to the image and question pair. In the second stage, the top candidate answers with the highest probabilities from the first stage will be selected and re-ranked by synergizing with image and question features.

4. **Reinforcement Learning Architecture** [5, 8]. In two frameworks, the authors train the agents to play the guess game about the content of the image. The experiments show that two agents in [8] have ability to communicate with each other about certain attributes of the visual content such as the shape, color and styles.

5. **Reinforcement Learning and Generative Adversarial Learning architecture** [54]. The authors proposed a framework which requires both reinforcement learning and Generative Adversarial Learning to generate human-like responses to questions. The authors claimed that GAN architecture may better in the case of training data scarcity and less frequent words.

3.2.4 Attention Mechanisms

As classified above in terms of encoder category, the success of proposed models has been so far based on the design of attention mechanisms.

The simple attention mechanism works nicely in Visual Dialog is self-attention for each type of features. In general, let $\mathbf{X} \in \mathbb{R}^{d \times m}$, $\mathbf{X} = [\mathbf{x}_1, \dots \mathbf{x}_m]$ denote a set of m d-dimension feature vectors. We would like to find an aggregated vector representing for all vectors in $\mathbf{X}$. First, we find attention weights, $\boldsymbol{\alpha} \in \mathbb{R}^{1 \times m}$ for each vector in the set $\mathbf{X}$ using two learnable matrices, $\mathbf{W_1} \in \mathbb{R}^{d \times d}$ and $\mathbf{W_2} \in \mathbb{R}^{1 \times d}$. Then the aggregated vector can be obtained by multiplying the attention weights with $\mathbf{X}$:

$$\boldsymbol{\alpha} = \text{softmax}(\mathbf{W_2}\text{tanh}(\mathbf{W_1}\mathbf{X})) \tag{3.1}$$

$$\mathbf{x} = \mathbf{X}\boldsymbol{\alpha} \tag{3.2}$$

Therefore, we can find the aggregated vectors for dialog history turn, image regions, and question words respectively.

In terms of this category, we will discuss some attention mechanisms adopted in state-of-the-art methods as following:

1. **Recursive Visual Attention** mechanism [38]. The authors proposed three modules in its recursive attention algorithm including INFER module for determining whether the recursion is terminated and computing the weights of visual features, PAIR module for pairing the question with its relevant dialog history, and ATT module for outputting the visual attention guided by the question.

2. **Dual Attention** mechanism for visual reference resolution [21]. The authors proposed a dual attention mechanism based on the hypothesis that machine should solve the linguistic ambiguity from the dialog history first via REFER module, then find the visual references from the image via FIND module.

3. **Parallel Attention** for visual object discovery through dialogs and queries [57]. In the proposed method, the authors use two attention modules to compute the attended features from Question-Answer pair in dialog turn to the global visual content and to the object candidates in the image.

4. **Multi-step Reasoning via Recurrent Dual Attention** mechanism for visual dialog [11]. The authors argued that most of existing methods adopt a single-step

reasoning which leads to inaccurate answers. Therefore, the authors proposed multi-step reasoning for attention using RNN to update the representation of question several times. The semantic representation of the question is updated from both image and dialog history and refined recurrently in the next step of reasoning.

Chapter 4

Our proposed Method

In chapter 4, We will the motivations and the overview of our proposed method for Visdial challenge in section 4.1. Next we present the details of the architecture of encoder and decoder in our model in section 4.2 and 4.3 based on the background introduced in chapter 2.

4.1 The motivations and overview

The motivations. Our method is proposed based on several observations from previous work as follows:

1. **Simpler in architecture.** Following the embeddings of image and text including dialog history and question, all previous work try to incorporate the features together by simple concatenation or even using some complicated attention mechanisms. We believe that there are many reasoning scenarios grounded question to image and dialog history such as question $\rightarrow$ image $\rightarrow$ dialog history, or question $\rightarrow$ dialog history $\rightarrow$ image $\rightarrow$ dialog history and so on. The simple and effective way is to compute the attended features between each other simultaneously in one reasoning step. In our work, we compute the attended features with attention from question

to image and question to history by using a simple mechanism which we call cross-attention. Cross-attention mechanism is simply a combination of several dense co-attention layers proposed in [37]. It leads to the simplicity in architecture.

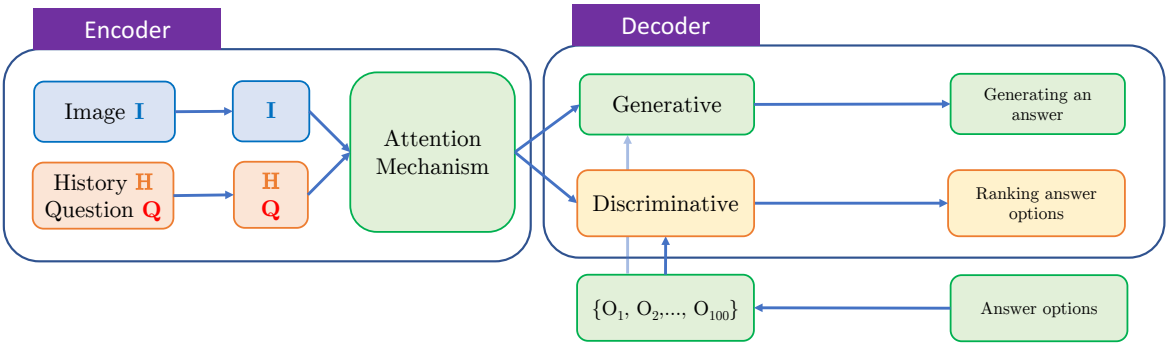
2. **Faster in runtime.** We observe that simple and parallel attention mechanism provides an efficient way for performing forward pass. Thanks to the cross-attention mechanism we can learn the attended features in parallel instead of learning and doing a forward pass in recurrent sequential like other attention mechanisms in the state-of-the-art methods.

3. **More accurate in performance.** In order to obtain state-of-the-art results on Visdial challenge, most works strive to learn discriminative task and generative tasks separately or in even diverse mechanisms such as GAN or two-stage mechanism. We, however, believe that the synergy of combining and learning two tasks help to take the advantages of each other to boost up the overall performance. Therefore, we introduce the multi-task model to learn two tasks simultaneously.

4. **More human-like in addressing tasks.** Discriminative models usually yield better performance in most metrics of Visdial challenge. However, those models only have capacity of predicting the ranking scores of the input answers given by human. This limitation makes it less applicable in reality because we need to develop a system that has the ability to generating an answer itself for the question from human about the visual content. Building such a system is one of our motivation to learn the generative task.

The overview. As presented above, those inspiring observations motivate us to propose a multi-task model as depicted in Figure 4-1. In encoder side, we examine two versions of language models including recurrent neural networks, and several of transformer encoder blocks to process dialog history and question into their embeddings while extracting features from images using the pretrained Faster R-CNN. In decoder side, the discriminative module has the responsibility to rank the answer options while the generative task is to generate the answer without the constraint of the input list of human-written answers. The details of two modules will be presented in the following sections.

Figure 4-1: The overview of our proposed model

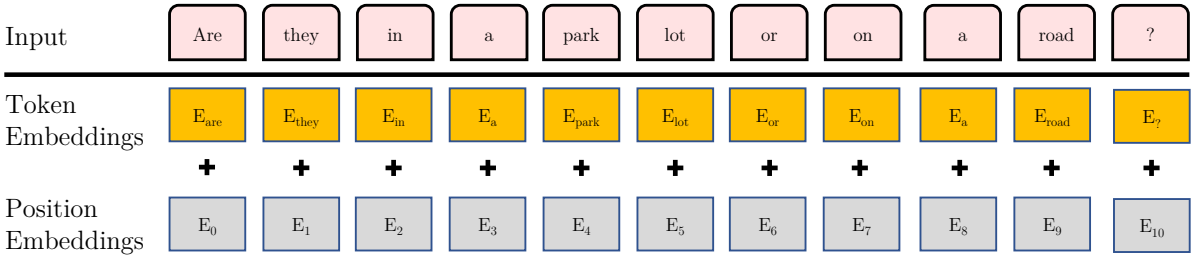


4.2 The encoder architecture

4.2.1 Language encoding

We use two versions of language modeling (**LM**) presented in section 2.5: (i) two Bi-LSTM layers, (ii) a stack of six transformer encoder layers (TEL) to process text inputs including dialog history **H**, question **Q**, and a list of candidate answers **A** for question **Q**.

Figure 4-2: The input embedding of text sequence which is the summation of token embeddings and position embeddings. The figure illustrates an example of computing the input embeddings for the sequence "Are they in a parking lot or on a road?".



Question encoding

Given a question **Q** of N tokens:

$$\mathbf{Q} = (\mathbf{w_1^Q}, \mathbf{w_2^Q}, \ldots, \mathbf{w_N^Q})$$

in which $\mathbf{w_i^Q}$ is a one-hot vector for the i^{th} token in the question. We convert the sequence $\mathbf{Q}$ into token embeddings $\mathbf{E^{QT}}$ using GloVe vectors [39]:

$$\mathbf{E^{QT}} = (\mathbf{e_1^{QT}}, \mathbf{e_2^{QT}}, \ldots, \mathbf{e_N^{QT}})$$

We also compute the position embeddings using the formula introduced in subsection 2.5.2 to obtain $\mathbf{E^{QP}}$:

$$\mathbf{E^{QP}} = (\mathbf{e_1^{QP}}, \mathbf{e_2^{QP}}, \ldots, \mathbf{e_N^{QP}})$$

The final input embedding for question $\mathbf{Q}$ is computed by the summation of token embeddings and position embeddings, $\mathbf{E^Q} = (\mathbf{e_1^Q}, \mathbf{e_2^Q}, \ldots, \mathbf{e_N^Q})$, as below:

$$\mathbf{e_i^Q} = \mathbf{e_i^{QT}} + \mathbf{e_i^{QP}} \text{ for } i = 1, ..., N \tag{4.1}$$

Next, we forward the input embeddings of question into the language modeling layers $\mathbf{LM}$ to learn the embedding vector for each token in the question with $\mathbf{q_i} \in \mathbb{R}^d$. See in subsection 2.5.1 and 2.5.2 for more details. Finally, we obtain the encoding features for question Q:

$$\begin{aligned}\mathbf{Q} &= [\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_N}] \\ &= \mathbf{LM}(\mathbf{E^Q})\end{aligned}$$

History encoding

Given a dialog history $\mathbf{H}$ of M rounds:

$$\mathbf{H} = (\mathbf{H^1}, \mathbf{H^2}, \ldots, \mathbf{H^M})$$

where $\mathbf{H^i}$ is the i^{th} round in the dialog history including a question $\mathbf{Q^i}$, and its ground truth answer, $\mathbf{O^{i(gt)}}$. We use only the truncated sequence of $\mathbf{H^i}$ including $2N$ tokens:

$$\begin{aligned}\mathbf{H^i} &= (\mathbf{w_1^{Q^i}}, \ldots, \mathbf{w_N^{Q^i}}, \mathbf{w_1^{O^{i(gt)}}}, \ldots, \mathbf{w_N^{O^{i(gt)}}}) \\ \mathbf{H^i} &= (\mathbf{w_1^{H^i}}, \mathbf{w_2^{H^i}}, \ldots, \mathbf{w_{2N}^{H^i}}) \text{ for } i = 1, ..., N\end{aligned}$$

For each round in the history $\mathbf{H}$, we follow the same procedure applied for question $\mathbf{Q}$ to get the output $\mathbf{F_{H^i}}$:

$$\mathbf{F_{H^i}} = (\mathbf{h_1^i}, \mathbf{h_2^i}, \ldots, \mathbf{h_{2N}^i}) \tag{4.2}$$

To obtain the aggregated feature for history round i, $\mathbf{H^i}$, we forward $\mathbf{h_j^i}$ for $j = 1, ..., 2N$ into an MLP of 2 linear layers with size (d, d), $(d, 1)$ respectively, using ReLU activation

in its hidden layer. Then we use softmax function to squash the logit outputs into range 0..1 for deriving the attention weights of all tokens, $(\alpha_1^i, \alpha_2^i, \ldots, \alpha_{2N}^i)$:

$$(\alpha_1^i, \alpha_2^i, \ldots, \alpha_{2N}^i) = \text{softmax}(\text{MLP}(F_{H^i})) \tag{4.3}$$

The we compute the aggregated feature representation $\mathbf{F_{H^i}}$for dialog history round i, $\mathbf{H^i}$:

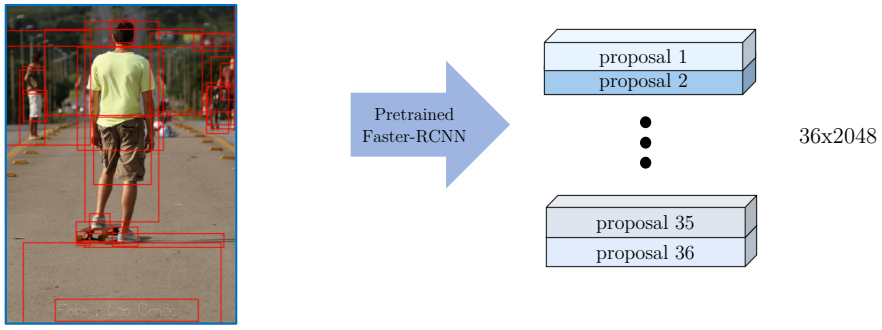
$$\mathbf{F_{H^i}} = \sum_{j=1}^{2N} \alpha_j^i \mathbf{h_j^i} \tag{4.4}$$

We repeat the above procedure for all rounds in the dialog history $\mathbf{H}$ to obtain the feature representation $\mathbf{H}$ as below:

$$\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_M}]$$

in which $\mathbf{h_i} = \mathbf{F_{H^i}}$ and $\mathbf{h_i} \in \mathbb{R}^d$.

Figure 4-3: The figure illustrates an example of etracting the feature representations for the image I



The image I

4.2.2 Image Encoding

As presented in 2, using pretrained Faster R-CNN we extract the features of 36 proposal regions with the maximum class-agnostic probability scores. The image representation of:

$$\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_{36}}]$$

in which $\mathbf{v_i}$ has the vector with dimension of 2048 extracted from *RoIAlign* pooling layers for the proposal region i using pretrained Faster R-CNN.

We project the $\mathbf{v_i}$ from 2048-dimensional space into d-dimensional space where $d = 512$

by using a learnable weight matrix of $(d \times 2048)$ and tanh activation as follows:

$$v_i = \tanh(\mathbf{W}\mathbf{v_i} + b) \tag{4.5}$$

4.2.3 Cross-attention Layer

Dense co-attention layer [37] was proposed to compute attended features $\hat{\mathbf{Q}}_\mathbf{l}$ and $\hat{\mathbf{V}}_\mathbf{l}$ at the l^{th} layer. In our work, we extend dense co-attention mechanism to compute $\hat{\mathbf{Q}}_\mathbf{l}$, $\hat{\mathbf{V}}_\mathbf{l}$ $and\hat{\mathbf{H}}_\mathbf{l}$ at the l^{th} layer as follows.

The $(l+1)^{th}$ cross-attention layer take the the input of $\mathbf{Q_l} = [\mathbf{q_{l,1}}, \mathbf{q_{l,2}}, ..., \mathbf{q_{l,N}}]$ of size $(d \times N)$, $\mathbf{H_l} = [\mathbf{h_{l,1}}, \mathbf{h_{l,2}}, ..., \mathbf{h_{l,M}}]$ of size $(d \times M)$, $\mathbf{V_l} = [\mathbf{v_{l,1}}, \mathbf{v_{l,2}}, ..., \mathbf{v_{l,T}}]$ of size $(d \times T)$ and return its updated versions.

Let $\mathbf{Q_0} = \mathbf{Q} = [\mathbf{q_1}, \mathbf{q_2}, ..., \mathbf{q_N}]$, $\mathbf{H_0} = \mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_M}]$ and $\mathbf{V_0} = \mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_T}]$.

The multi-head attentions can be computed in parallel to learn diverse attention maps by projecting features into lower dimensional spaces. Let h be the number of lower dimensional spaces and $d_h = d/h$ be their dimension. We denote the linear projections by the learnable weights $\mathbf{W^i_{V_l}} \in \mathbb{R}^{d_h \times d}$ and $\mathbf{W^i_{Q_l}} \in \mathbb{R}^{d_h \times d}$. Then we compute the affinity matrix between the projected features in the i-th space as:

$$\mathbf{A^i_l} = (\mathbf{W^i_{V_l}}\mathbf{V_l})^\intercal(\mathbf{W^i_{Q_l}}\mathbf{Q_l}) \tag{4.6}$$

We compute attention maps on question tokens guided by each image region in the i-th space as:

$$\mathbf{A^{(i)(V)}_{Q_l}} = \text{softmax}(\frac{\mathbf{A^{(i)}_l}}{\sqrt{d_h}}) \tag{4.7}$$

and also the attention maps on image regions guided by each question word in the i-th space as:

$$\mathbf{A^{(i)(Q)}_{V_l}} = \text{softmax}(\frac{\mathbf{A^{(i)\intercal}_l}}{\sqrt{d_h}}) \tag{4.8}$$

We compute the attention maps on d-dimensional space by averaging all corresponding attenion maps in lower dimensional spaces:

$$\mathbf{A^{(V)}_{Q_l}} = \frac{1}{h}\sum_{i=1}^{h}\mathbf{A^{(i)(V)}_{Q_l}} \tag{4.9}$$

$$\mathbf{A}_{\mathbf{V_l}}^{(\mathbf{Q})} = \frac{1}{h}\sum_{i=1}^{h}\mathbf{A}_{\mathbf{V_l}}^{(\mathbf{i})(\mathbf{Q})} \tag{4.10}$$

We derive the attended features $\hat{\mathbf{Q}}_{\mathbf{l}}^{(\mathbf{V})}$ of size (d) guided by image region and $\hat{\mathbf{V}}_{\mathbf{l}}^{(\mathbf{Q})}$ of size (d) guided by question word as:

$$\hat{\mathbf{Q}}_{\mathbf{l}}^{(\mathbf{V})} = \mathbf{Q_l}\mathbf{A}_{\mathbf{Q_l}}^{(\mathbf{V})\intercal} \tag{4.11}$$

$$\hat{\mathbf{V}}_{\mathbf{l}}^{(\mathbf{Q})} = \mathbf{V_l}\mathbf{A}_{\mathbf{V_l}}^{(\mathbf{Q})\intercal} \tag{4.12}$$

Similarly, we obtain attended features from history to question $\hat{\mathbf{Q}}_{\mathbf{l}}^{(\mathbf{H})}$, from question to history $\hat{\mathbf{H}}_{\mathbf{l}}^{(\mathbf{Q})}$, from history to image $\hat{\mathbf{V}}_{\mathbf{l}}^{(\mathbf{H})}$ and from image to history $\hat{\mathbf{H}}_{\mathbf{l}}^{(\mathbf{V})}$.

Fusing image, history and question representations. We update the represention $\mathbf{q_{l,n}}$ by concatenating with the attended features $\hat{\mathbf{v}}_{\mathbf{l,n}}$ and $\hat{\mathbf{h}}_{\mathbf{l,n}}$ to form a $3d$-dimensional vector $[\mathbf{q}_{\mathbf{l,n}}^{\intercal}, \hat{\mathbf{v}}_{\mathbf{l,n}}^{\intercal}, \hat{\mathbf{h}}_{\mathbf{l,n}}^{\intercal}]$. We project the concatenated vector into d-dimensional space and add it to $\mathbf{q_{l,n}}$ as follows:

$$\mathbf{q}_{(\mathbf{l+1}),\mathbf{n}} = \mathrm{ReLU}(\mathbf{W}_{\mathbf{Q_l}}\begin{bmatrix}\mathbf{q_{l,n}}\\ \hat{\mathbf{v}}_{\mathbf{l,n}}\\ \hat{\mathbf{h}}_{\mathbf{l,n}}\end{bmatrix} + b_{Q_l}) + \mathbf{q_{l,n}} \tag{4.13}$$

$$\mathbf{Q_{l+1}} = [\mathbf{q}_{(\mathbf{l+1}),\mathbf{1}}, ..., \mathbf{q}_{(\mathbf{l+1}),\mathbf{N}}] \tag{4.14}$$

The procedure is repeated for all the question words to obtain $\mathbf{Q_{l+1}}$. Similarly, we update the representations of image and history to obtain $\mathbf{V_{l+1}}$ and $\mathbf{H_{l+1}}$:

$$\mathbf{V_{l+1}} = [\mathbf{v}_{(\mathbf{l+1}),\mathbf{1}}, ..., \mathbf{v}_{(\mathbf{l+1}),\mathbf{T}}] \tag{4.15}$$

$$\mathbf{H_{l+1}} = [\mathbf{h}_{(\mathbf{l+1}),\mathbf{1}}, ..., \mathbf{h}_{(\mathbf{l+1}),\mathbf{M}}] \tag{4.16}$$

4.2.4 Self-attention

Given N d-dimensional vectors of $[\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}]$, we would like to compute the aggregated vector $\mathbf{s_x}$ of d-dimensional by using the self-attention mechanism as follows:

First, we forward $\mathbf{x_i}$ for $i = 1, ..., N$ into an MLP of 2 linear layers with size (d, d), $(d, 1)$ respectively, using ReLU activation in its hidden layer. Then we use softmax function to squash the logit outputs into range 0..1 for deriving the attention weights of all tokens,

$(\alpha_1, \alpha_2, \ldots, \alpha_N)$:

$$(\alpha_1, \alpha_2, \ldots, \alpha_N) = \text{softmax}(\text{MLP}(\mathbf{X})) \tag{4.17}$$

The we compute the aggregated feature representation as:

$$\mathbf{s_x} = \sum_{i=1}^{N} \alpha_i \mathbf{x_i} \tag{4.18}$$

4.2.5 Return the output vector of encoder

We use two cross-attention layers to perform two reasoning steps. After 2 reasoning steps, we obtain $\mathbf{V_2} = [\mathbf{v_{2,1}}, ..., \mathbf{v_{2,T}}]$ and $\mathbf{H_2} = [\mathbf{h_{2,1}}, ..., \mathbf{h_{2,M}}]$ and $\mathbf{Q_2} = [\mathbf{q_{2,1}}, ..., \mathbf{q_{2,N}}]$.

We use the self-attention mechanism above to obtain the aggregated features for image $\mathbf{s_v}$, history $\mathbf{s_h}$ and question $\mathbf{s_q}$. We concatenate all three features to obtain $[\mathbf{s_v^\intercal}, \mathbf{s_h^\intercal}, \mathbf{s_q^\intercal}]$ Next, we project the concatenated features of $3d$-dimensional back $\mathbf{s}$ d-dimensional with linear layers. Finally, the encoder yields the encoding vector $\mathbf{s}$ of d-dimensional.

4.3 The decoder architecture

4.3.1 Answer encoding

Given a list of 100 answer options $\mathbf{O}$ for question $\mathbf{Q}$:

$$\mathbf{O} = (\mathbf{O^1}, \mathbf{O^2}, \ldots, \mathbf{O^{100}})$$

where $\mathbf{O^i}$ is a sequence of N tokens:

$$\mathbf{O^i} = (\mathbf{w_1^{O^i}}, \ldots, \mathbf{w_N^{O^i}})$$

For each answer $\mathbf{O^i}$ in the selection list, we follow the same procedure applied for question Q in 4.2.1 to get the output $\mathbf{O^i}$:

$$\mathbf{O^i} = (\mathbf{o_1^i}, \mathbf{o_2^i}, \ldots, \mathbf{o_N^i})$$

To obtain the aggregated feature for answer $\mathbf{O^i}$, we forward $\mathbf{o_j^i}$ for $j = 1, ..., N$ into an MLP of 2 linear layers with size (d, d), $(d, 1)$ respectively, using ReLU activation in its hidden layer. The MLP module here has different learnable weights as one in 4.2.1. Next, we

use softmax function to squash the logit outputs into range 0..1 for deriving the attention weights of all tokens, $(\alpha_1^i, \alpha_2^i, \ldots, \alpha_N^i)$:

$$(\alpha_1^i, \alpha_2^i, \ldots, \alpha_N^i) = \text{softmax}(\text{MLP}(\mathbf{F_{O^i}})) \tag{4.19}$$

Then we compute the aggregated feature representation $\mathbf{F_{O^i}}$for answer $\mathbf{O^i}$:

$$\mathbf{F_{O^i}} = \sum_{j=1}^{N} \alpha_j^i \mathbf{o_j^i} \tag{4.20}$$

We repeat the above procedure for all 100 answer options in the list $\mathbf{O}$ to obtain the feature representation O as below:

$$\mathbf{O} = (\mathbf{o_1}, \mathbf{o_2}, \ldots, \mathbf{o_{100}})$$

in which $\mathbf{o_i} = \mathbf{F_{O^i}}$ and $\mathbf{o_i} \in \mathbb{R}^d$.

The ground truth answer. For the ground truth answer $\mathbf{O_{gt}}$ of question $\mathbf{Q}$, we keep the representation $S_{O^{gt}}$ for the generative task:

$$\mathbf{S_{O^{gt}}} = (\mathbf{o_1^{gt}}, \mathbf{o_2^{gt}}, \ldots, \mathbf{o_N^{gt}})$$

where gt is the index of the ground truth answer in the candidate list.

4.3.2 The discriminative module

We add the discriminative module into the decoder for learning to produce the distribution over the answer options $\mathbf{O}$ for question $\mathbf{Q}$. Given the encoding d-dimensional vector $\mathbf{s}$ from the encoder and feature representations for answer options $(\mathbf{o_1}, \mathbf{o_2}, \ldots, \mathbf{o_{100}})$, the discriminative module returns the ranking scores $(s_{o_1}, \ldots, s_{o_{100}})$ all answer options which learn to maximize the probability of the ground truth answer.

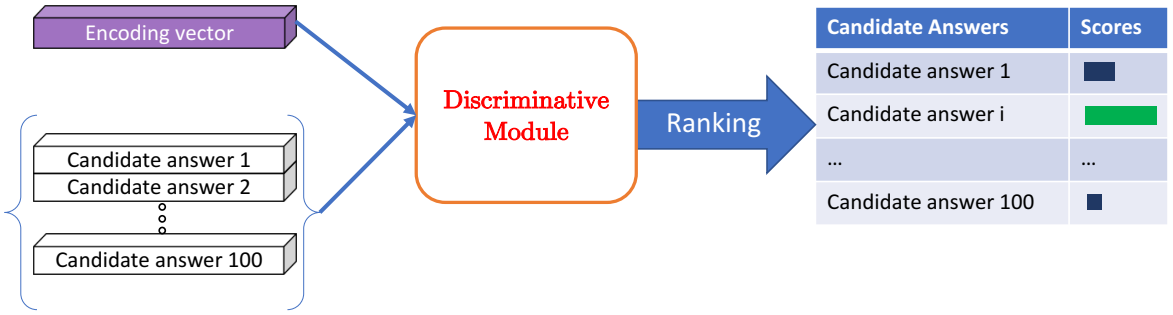
To compute the ranking scores, we first compute the logit score for each option by the dot-product of the encoding vector $\mathbf{s}$ and the option vector:

$$s_{o_i} = \mathbf{o_i^\intercal} \cdot \mathbf{s} \tag{4.21}$$

Then we use softmax activation to obtain the final ranking scores:

$$(s_{o_1}, \ldots, s_{o_{100}}) = \text{softmax}(s_{o_1}, \ldots, s_{o_{100}}) \tag{4.22}$$

Figure 4-4: Illustration of discriminative module to produce the ranking scores for answer options.



4.3.3 The generative module

We add the discriminative module into the decoder for learning to generating the answer for question **Q**. Given the input the encoding d-dimensional vector **s** from the encoder and the ground truth answer of N tokens $(\mathbf{w_1^{gt}}, \mathbf{w_2^{gt}}, \ldots, \mathbf{w_N^{gt}})$ with the embedding representations $(\mathbf{o_1^{gt}}, \mathbf{o_2^{gt}}, \ldots, \mathbf{o_N^{gt}})$. First, we transform the ground truth answer into the input sequence $\mathbf{S_{in}^{gt}} = (\mathbf{w_{<SOS>}}, \mathbf{w_1^{gt}}, \mathbf{w_2^{gt}}, \ldots, \mathbf{w_N^{gt}})$ by adding 'start-of-sentence' token (<SOS>) in the start of the sequence, and $\mathbf{S_{out}^{gt}} = (\mathbf{w_1^{gt}}, \mathbf{w_2^{gt}}, \ldots, \mathbf{w_N^{gt}}, \mathbf{w_{<EOS>}})$ by adding the 'end-of-sentence' token (<EOS>) at the end of the sequence. The generative task is to learn generating the next word prediction from $\mathbf{S_{in}^{gt}}$ to $\mathbf{S_{out}^{gt}}$ using a stack of two LSTM layers following by a linear layer with learnable weights of $d \times N_{vocab}$ in which N_{vocab} is the number of word in vocabulary:

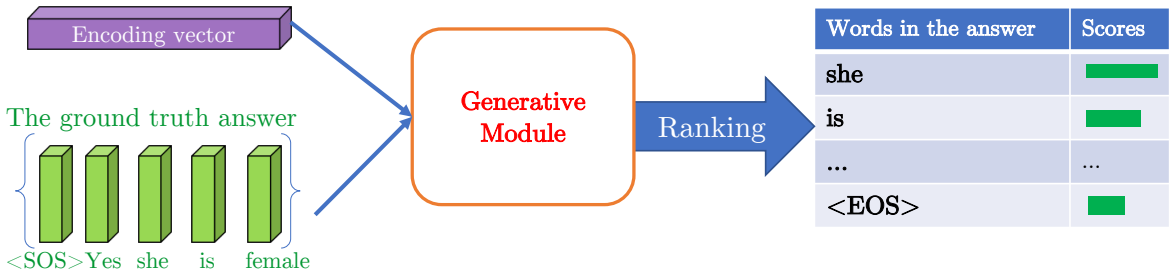
$$\mathbf{h_i} = \text{LSTM}(\mathbf{o_{i-1}}, \mathbf{h_{i-1}}) \tag{4.23}$$

$$\mathbf{S_{out,i}^{pred}} = \text{softmax}(\mathbf{W_{out}}\mathbf{h_i} + b_{out}) \tag{4.24}$$

where $\mathbf{o_i}$ is the embedding representation, and $\mathbf{S_{out,i}^{pred}}$ is the prediction for the token $\mathbf{S_{out,i}^{gt}}$ and $\mathbf{h_i}$ is the hidden values at the timestep i.

We initialize $\mathbf{h_0}$ by the encoding vector **s**.

Figure 4-5: Illustration of generative module to learn generate the answer by next word prediction from the ground truth answer during training.



4.4 The loss function

4.4.1 The discriminative loss

The discriminative module to maximize the log-likelihood of the ground truth answer $\mathbf{O^{gt}}$ in the candidate answer list $\mathbf{O}$.

$$\text{loss}_{disc} = -\frac{1}{N_Q}\sum \log(s_{o_{gt}}) \tag{4.25}$$

where N_Q is the number of the questions.

4.4.2 The generative loss

The generative module to maximize the log-likelihood of the prediction for next word prediction task for all the tokens in $\mathbf{O^{gt}}$. We use the cross entropy to compute the loss for each word prediction and averaging all predictions for every token i in $\mathbf{S_{out}}$:

$$\text{loss}_{gen} = -\frac{1}{N_Q}\frac{1}{N_T}\sum \log(\mathbf{S^{pred}_{out,i}}\mathbf{S^{gt}_{out,i}}) \tag{4.26}$$

where N_T is the number of maximum tokens (words) in the ground truth answer.

4.4.3 The model loss

We use $\gamma = 1$ to learn two tasks simultaneously:

$$\text{loss} = \gamma\text{loss}_{disc} + \text{loss}_{gen} \tag{4.27}$$

Chapter 5

Experiments and Results

5.1 Visdial Dataset

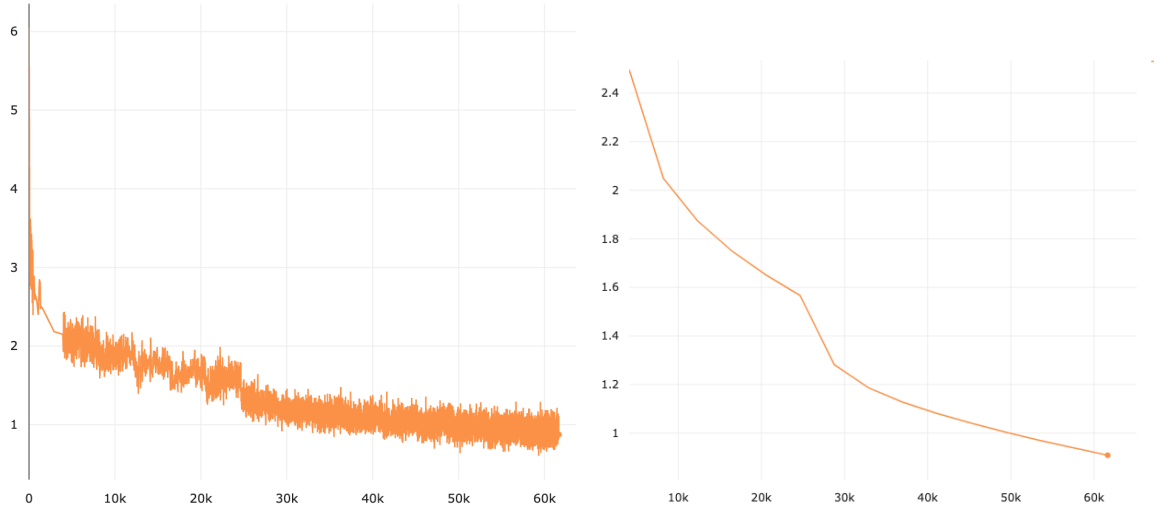
We use the version 1.0 of Visdial in our experiments. More detailed analysis of Visdial dataset was presented in chapter 2. Below is the excerpt from chapter 2 about three splits of VisDial dataset version 1.0:

1. **Training split.** There are total 123,287 images from COCO dataset. Each image has 10 rounds of question-answer. They all make up total of 1,232,870 pairs of question-answer pairs.

2. **Validation split.** There are total 2,064 images from Flickr in which the author gaurantee the distribution to be the same as COCO dataset. Each image has 10 rounds of question-answer. They all make up total of 20,640 pairs of question-answer pairs.

3. **Test split.** There are total 8,000 images from Flickr dataset in which the author gaurantee the distribution to be the same as COCO dataset. Each image has **N** rounds of question-answer. The value **N** would be different according to its image.

We use the training split for training the models, validation split for validating and comparing models on ablation study, and test split for obtaining the final scores reported by



(a) The batch loss during the training procedure (b) The epoch loss during the training procedure. rounds

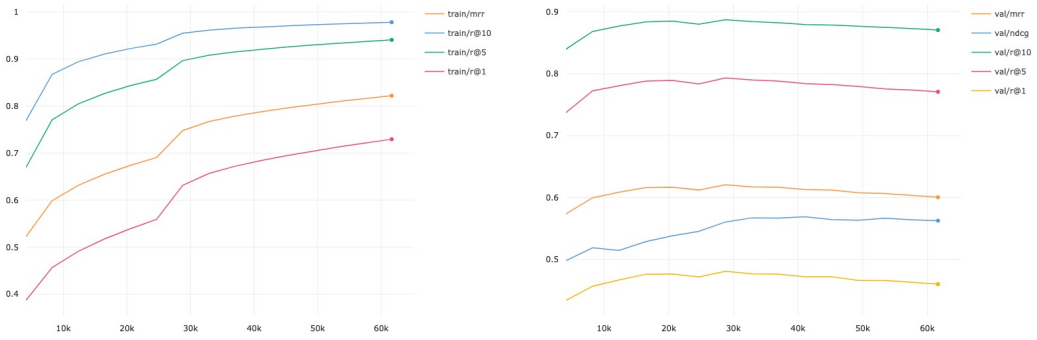
Figure 5-1: The visualization of batch loss and epoch loss during the training procedure of Bi-LSTM version on Visidal dataset.

the server.

Preprocessing. The caption is considered to be the first round in the dialog history as the same in previous works. We truncate captions/questions/answers that are longer than 40/20/20 words, respectively. Later, we use a dictionary with a vocabulary of 10154 unique words that appear at least 5 times in train split. Also, all the text intputs are embedded to a 300-dimensional vector initialized by GloVe embeddings [39]. The word embedding vector is fine-tuned during training.

5.2 Experimental Settings

In our experiments, the encoder layers for processing dialog history, question and answer options have two versions: (i) Bi-LSTM version - a stack of 2 bidirectional LSTM layers with hidden size 512, and (ii) Transformer version - a stack of 6 transformer layers with the hidden size of 512 and the number of heads in attention of 8 while the feed foward network layer of transformer has the hidden size d_{ff} of 2048. For cross-attention mechanism, the hidden size is fixed at 512 with the number of multiple heads in attention is 4. The number of multiple reasoning for cross-attention mechanism is set at 2.



(a) The MRR, R@1, R@5, R@10 scores of training set during training procedure

(b) The MRR, R@1, R@5, R@10, NDCG scores of validation set during training procedure

Figure 5-2: The visualization of metric scores of training and validation sets during the training procedure of Bi-LSTM version on Visidal dataset.

We use two cross-attention layers to do reasoning twice. Inside each cross-attention layer, each dense co-attention [37] has $K = 3$ "nowhere to attend" memory which the authors claimed it yields the best performance in VQA tasks. The number of parallel attention maps is also set at $h = 4$.

Due to the limitation of computing resources, we do not perform any hyperparameter tuning and regularization methods rather than applying dropout and early stopping on the validation set. The dropout ratio is set to 0.2.

We use Adam algorithm [23] for optimization with the initial learning rate of $1e^{-3}$, and training in 25 epochs with a learning rate scheduler which halves the learning rate after 10 and 15 epochs respectively. Figure 5-1 (a) visualizes the batch loss during the training procedure while Figure 5-1 (a) visualizes the epoch loss on average during the training procedure. Moreover, Figure 5-2 show us the metric scores of our model on training and validation set during training procedure.

5.3 Experimental Results

Baselines. We compare our proposed method (Bi-LSTM version) with state-of-the-art models on Visdial test set v1.0, including Memory Network (MN) [7], Late Fusion

Table 5.1: Comparison of our proposed model to state-of-the-art methods on VisDial v1.0 validation set. Higher is better for NDCG, MRR and Recall@k while lower is better for mean rank.

Model	**Publication**	**NDCG**	**MRR**	**R@1**	**R@5**	**R@10**	**Mean**
Our model	-	57.40	61.24	48.05	77.20	86.53	5.16
Published							
ReDAN [11]	ACL2019	57.63	64.75	51.10	81.73	90.90	3.89
DAN [21]	CVPR2019	57.59	63.20	49.63	79.75	89.35	4.30
Sync [13]	CVPR2019	57.32	62.20	47.90	80.43	89.95	4.17
RvA [38]	CVPR2019	55.59	63.03	49.03	80.40	89.83	4.18
FGA [46]	CVPR2019	54.46	67.25	53.40	85.28	92.70	3.54
GNN [55]	CVPR2019	52.82	61.37	47.33	77.98	87.83	4.57
CorefNMN [25]	ECCV2018	54.70	61.50	47.55	78.10	88.80	4.40
LF-Att [7]	CVPR2017	49.76	57.07	42.08	74.83	85.05	5.41
MN-Att [7]	CVPR2017	49.58	56.90	42.43	74.00	84.35	5.59
MN [7]	CVPR2017	47.50	55.49	40.98	72.30	83.30	5.92
HRE [7]	CVPR2017	45.46	54.16	39.93	70.45	81.50	6.41
LF [7]	CVPR2017	45.31	55.42	40.95	72.45	82.83	5.95
Unpublished							
DL-61	-	57.88	63.42	49.30	80.77	90.68	3.97
USTC-YTH	-	56.47	61.44	47.65	78.13	87.88	4.65
MS ConvAI	-	55.35	63.27	49.53	80.40	89.60	4.15

Network (LF) [7], Dual Attention Network (DAN) [21], CorefNMN [25], Factor Graph attention netowrk (FGA), [46], Graph Neural Network (GNN) [55], and other state-of-the-art unpublished work DL-61, USTC-YTH and MS ConvAI from last year challenge.

We evaluate our proposed method on the blind test-std v1.0 set, by submitting results to the online evaluation server. Table 5.1 shows the comparison between our model and state-of-the-art visual dialog models. Since 2018, **NDCG** is used as the major metric to compare the performance among different methods. Our model outperforms CorefNMN, FGA, GNN, LF-Att, MN-Att, HRE and LF models with a large margin of NDCG score. Moreover, compared with the latest published state-of-the-art methods including ReDAN and DAN, our proposed model yields very competitive NDCG score while our method uses the simple attention mechanism and network architecture.

5.4 Ablation Study

Table 5.2: Ablation study on each component to the model on the validation set of Visdial dataset version 1.0. The sign * indicates the module will be employed in the final model.

Component	**Details**	**NDCG**
Language modeling	2 LSTM layers	56.48
	2 Bi-LSTM layers*	59.64
	6 Transformer layers	58.42
The number of cross-attention layers	0	56.75
	1	58.96
	2*	59.64
	3	57.89
Decoder type	Discriminative	57.74
	Generative	57.38
	Both*	59.64

To analyze the contribution of each component in building the model for Visdial challenge, we performed the ablation study of the model on the validation set of Visdial dataset version 1.0. Table 5.2 shows the ablative results on three components: the language modeling for processing text, the number of cross-attention layers in a model, the choice of decoders. All the models in the ablation study were trained with the same settings and the same manner provided in section 5.2.

The choice of language modeling module. The first block of Table 5.2 is the results of using different types of language modeling including (i) a model containing a stack of two LSTM layers with 56.48 NDCG score, (ii) a model containing a stack of two bidirectional LSTM layers with 59.64 NDCG scores (the best), and (iii) a model containing a stack of 6 transformer layers. All the models have language modeling components with hidden size of 512. The reason why we chose the number of transformer layers is 6 because it has the equivalent parameters compared with 2 Bi-LSTM layers. Although faster than 2 Bi-LSTM in terms of inference speed, the model using transformer layers may need more time to converge during training. After 25 epochs epochs, we found that a model of 2 Bi-LSTM converged faster and reached a peak at the epoch 12.

The number of cross-attention layers. In the second block of Table 5.2, the results indicate the impact of cross-attention layers in the model. Obviously, using from one up to three layers boosts the NDCG at least by about 1.0 score. The model achieved the best NDCG score when the number of layers is set at 2. Two cross-attention layers may provide better reasoning than one layer while less overfitting than using three cross-attention layers.

The choice of decoder type. In the last block of Table 5.2, the results show the effect of training models with different kinds of decoder: (i) discriminative decoder, (ii) generative decoder, and (iii) both decoders. Although the generative models yield much lower R@1 scores than discriminative models, their NDCG score is on par with discriminative models'. When we train two tasks simultaneously, the multi-task model gained the overall NDCG by about 2.0 score. These results somehow confirm our hypothesis that the multi-task models enjoy the synergy of two tasks.

From the initial ablative results, we use the multi-task model with 2 Bi-LSTM layers and 2 cross-attention layers as our final model. We call it a Bi-LSTM version which appeared in section 5.1 and 5.2.

Chapter 6

Conclusion

Conclusion. In this thesis, we investigated visual dialog task, specifically Visdial challenge, and several existing methods for the problem. Previous works mostly focus on training a separate task with complicated attention mechanisms. We believe that there are many reasoning scenarios among question, image and dialog history, such as question $\rightarrow$ image $\rightarrow$ dialog history, or question $\rightarrow$ dialog history $\rightarrow$ image $\rightarrow$ dialog history and so on. The simple and effective way is to compute the attended features between each other simultaneously in one reasoning step. We adapted the dense co-attention mechanism to cross-attention mechanism to learn attended features among different kinds of features. This allows the model to run in parallel with faster runtime. We also proposed the end-to-end trainable framework learning two tasks simultaneously: (i) discriminative task to rank the answer options and (ii) generative task to generate the answer without the constraint of the input list of human-written answers. We observe that training a multi-modal framework allows both tasks to enjoy the synergy each other to boost the performance. Moreover, our model possesses the capacity of generating an answer rather ranking the given answer options. To evaluate the performance of the system, we performed several experiments for ablation study as well as comparing our models with several baselines and other state-of-the-art methods. The experimental results reported by standard evaluation protocol indicate that our model achieved very competitive performance on par with other state-of-the-art methods.

Future Work. Besides the current use of Bi-LSTM layers, in our work, we provide the first step in the direction of using transformer blocks for processing text as well as multi-task model with cross-attention mechanism. In the future, we would like to continue our work with some ideas:

1. **Better language modeling for text representation.** In the future, we would like to examine more advanced contextual vector pretrained on language models BERT [10], OpenAI-GPT [42], ELMo [40], etc.

2. **Improve the generative decoder.** Although the NDCG score of generative decoder is on par with discriminative decoder, the rest scores of generative decoder are much lower with those of discriminative decoder. Therefore, improving the generative decoder may boost the overall performance of the model. There are several promising solutions for generative decoders such as using attention weights for all the context vectors from question, image and dialog history, using the negative example during the training. Using the transformer decoder in the generative decoder is also worth trying.

3. **More thorough ablative studies.** Since the limitation of computing resources, only a few experiments were done to provide short ablative study. Since with only one GPU Titan X of 12GB, it took about 8 to 9 days to complete the training procedure of one model. In the future, we would like to perform more experiments for the exhaustive ablation study.

Bibliography

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342. ACM, 2010.

[4] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016.

[5] Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. Evaluating visual conversational agents via cooperative human-ai games. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*, 2017.

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[7] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2017.

[8] Abhishek Das, Satwik Kottur, José MF 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*, pages 2951–2960, 2017.

[9] Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5503–5512, 2017.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Zhe Gan, Yu Cheng, Ahmed El Kholy, Linjie Li, Jingjing Liu, and Jianfeng Gao. Multi-step reasoning via recurrent dual attention for visual dialog. *ArXiv*, abs/1902.00579, 2019.

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[13] Dalu Guo, Chang Xu, and Dacheng Tao. Image-question-answer synergistic network for visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Shayan Hassantabar. Visual question answering: Datasets, methods, challenges and oppurtunities. 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] 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.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] MD Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):118, 2019.

[19] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813, 2017.

[20] Ilija Ilievski and Jiashi Feng. Multimodal learning and reasoning for visual question answering. In *Advances in Neural Information Processing Systems*, pages 551–562, 2017.

[21] Gi-Cheon Kang, Jaeseo Lim, and Byoung-Tak Zhang. Dual attention networks for visual reference resolution in visual dialog. *ArXiv*, abs/1902.09368, 2019.

[22] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

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

[24] Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *ECCV*, 2018.

[25] Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–169, 2018.

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[29] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

[30] Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *Advances in Neural Information Processing Systems*, pages 314–324, 2017.

[31] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[32] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9, 2015.

[33] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[34] Ishan Misra, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens van der Maaten. Learning by asking questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2018.

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[36] Medhini Narasimhan and Alexander G Schwing. Straight to the facts: Learning knowledge base retrieval for factual visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 451–468, 2018.

[37] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6087–6096, 2018.

[38] Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. Recursive visual attention in visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[40] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[41] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[42] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

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

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[45] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[46] Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G. Schwing. Factor graph attention. *CoRR*, abs/1904.05880, 2019.

[47] Kevin J Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4613–4621, 2016.

[48] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[50] Andeep S Toor, Harry Wechsler, and Michele Nappi. Biometric surveillance using visual question answering. *Pattern Recognition Letters*, 2018.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[52] Limin Wang, Sheng Guo, Weilin Huang, Yuanjun Xiong, and Yu Qiao. Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns. *IEEE Transactions on Image Processing*, 26(4):2055–2068, 2017.

[53] 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.

[54] Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton van den Hengel. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6106–6115, 2018.

[55] Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. Reasoning visual dialogs with structural and partial observations. *CoRR*, abs/1904.05548, 2019.

[56] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016.

[57] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian D. Reid, and Anton van den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. *CoRR*, abs/1711.06370, 2017.