

Visual Question Answering

A Thesis submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Technology
by

Vatsal Goel

160108018

Mohit Chandak

160108026

Under the guidance of
Dr. Prithwijit Guha

and

Dr. Ashish Anand



DEPARTMENT OF ELECTRONICS & ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

June 2020

© Copyright by Vatsal Goel and Mohit Chandak, June 2020
All Rights Reserved

Abstract

Artificial Intelligence has been a hotspot of research for the past couple of decades. After the advent of progress in Deep Learning, research in the fields of Computer Vision, Natural Language Processing, Reasoning, and Causality has increased dramatically. Multidisciplinary research is considered as a leap into the era of AGI, Artificial General Intelligence. In one such attempt, a novel Image Understanding task was proposed to combine progress in Computer Vision and Natural Language Processing, and hence stimulate further developments in both disciplines.

Visual Question Answering takes an image and a question about that image, and produces an answer. The involvement of both Computer Vision and Natural Language Processing makes this task even more exciting and challenging. Even though there has been tremendous progress in the field of Visual Question Answering, models today still tend to learn from language biases in the dataset leading to inconsistent performance. To this end, we propose a model-independent cyclic framework which increases consistency of any VQA architecture. We train our models to answer the original question, generate an implication based on the answer and then also learn to answer the generated implication correctly. As a part of the cyclic framework, we propose a novel implication generator which can generate implied questions from any question-answer pair. As a baseline for future works on consistency, we provide a new annotated VQA-Implications dataset. The dataset consists of ~30k questions containing implications of 3 types - Logical Equivalence, Necessary Condition and Mutual Exclusion - made from the VQA v2.0 validation dataset. We show that our framework improves consistency of VQA models by ~15% on the rule-based dataset and ~7% on the VQA-Implications dataset, without degrading their accuracy.

Contents

Abstract	i
List of Figures	iv
1 Introduction	1
2 Literature	5
2.1 Deep Learning based VQA methods	5
2.1.1 Stacked Attention Networks	6
2.1.2 Teney et. al [1]	6
2.1.3 Pythia v0.1 [2]	7
2.1.4 Bilinear Attention Networks [3]	7
2.1.5 LXMERT [4]	8
2.2 Variations in VQA	9
2.2.1 VQA with Explanations [5]	9
2.2.2 Captions to aid VQA [6]	10
2.2.3 VQG: Generating Natural Questions About an Image [7]	10
2.2.4 iVQA: Inverse Visual Question Answering [8]	12
2.3 Language Prior in VQA	13
2.3.1 Elevating Image Understanding in VQA [9]	13
2.3.2 Explicit Bias Discovery in VQA Models [10]	14
2.4 Consistency of VQA Models	14
2.4.1 Inconsistency in VQA Models [11]	14
2.4.2 SQuINTing at VQA Models [12]	16
2.4.3 VQA-LOL: VQA under the Lens of Logic [13]	17
2.5 Cyclic Training in VQA	18

2.5.1	Cycle-Consistency for Robust VQA [14]	19
2.6	Conclusion	21
3	Approach	23
3.1	Implications	24
3.2	Implication Generator Module	25
3.3	Knob Mechanism	26
3.4	Cyclic Framework	26
4	Experiments and Results	29
4.1	Consistency performance	29
4.2	Attention Map comparison	31
4.3	Data Augmentation	32
4.4	VQA Rephrasings	32
4.5	Implication Generator Performance	33
4.6	Implementation details	34
4.7	Examples of Attention Maps	34
4.8	Generated Implications by our module	36
5	Conclusion	38

List of Figures

1.1	Few Inputs in Visual Question Answering(VQA) task [15]	2
1.2	Example of inconsistency in VQA models	3
2.1	Model overview for Teney et. al [1]	7
2.2	Overview of two-glimpse BAN [3]	7
2.3	LXMERT Architecture [4]	8
2.4	Overview of VQA-E task [5]	9
2.5	An illustration of adjustment using caption attention [6]	10
2.6	Examples of generated question-relevant captions [6]	11
2.7	Example right and wrong questions for VQG [7]	11
2.8	Generative Model for VQG [7]	12
2.9	(a) Illustration of iVQA task (b) Architecture for iVQA model [8]	12
2.10	Few illustrations of balanced VQA dataset [9]	13
2.11	Existing Biases in VQA models [10]	14
2.12	Inconsistent VQA predictions [11]	15
2.13	Generated Implications [11]	16
2.14	Inconsistency in Reasoning questions [12]	17
2.15	SQuINT Architecture [12]	17
2.16	Inconsistency in logical composition of questions [13]	18
2.17	VQA-LOL Model Architecture [13]	19
2.18	Examples of brittleness of VQA Models [14]	19
2.19	Cyclic Consistency model-agnostic framework [14]	20
2.20	Examples of VQA-Rephrasings dataset [14]	21
3.1	Detailed architecture of our Implication generator	25

3.2	Proposed Model Architecture	27
4.1	Qualitative example showing improvement in attention maps for Pythia	31
4.2	Comparison in Attention maps. Top and bottom rows represent Pythia [2] and Pythia trained in our framework respectively.	35

Chapter 1

Introduction

Computer Vision seeks to develop systems to process and understand visual inputs i.e. images. On the other hand, NLP revolves around increasing interactions between humans and machines in a natural language. Historically, these areas of research have gone through separate development, which makes this marriage even more significant. Image Understanding is an integral part of Computer Vision. Being able to extract useful information by looking at an image has numerous applications such as examining CCTV footage, identifying art forgeries and so on. In this regard, a novel Image Understanding task was developed at Georgia Tech in 2015 - Visual Question Answering [15].

The Visual Question Answering(VQA) task consists of an image and a contextual natural language question as input and an answer to that question as output. The questions vary from different attributes of the objects in the image to actions and background details. As a result, VQA differs comprehensively from caption generating tasks. In contrast to most computer vision tasks, including image segmentation and object recognition, where a predetermined question is asked for different input images, VQA needs to answer questions determined during runtime. In this sense, VQA requires a more general understanding of an image and hence, a challenging task to learn.

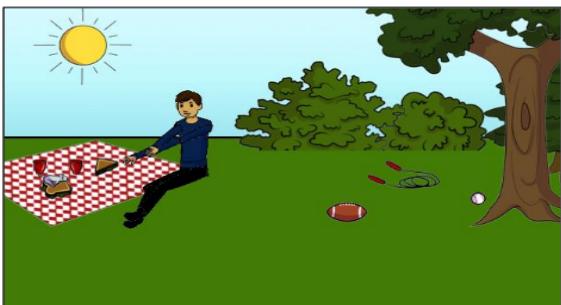
As evident from figure 1.1, open-ended questions require vast areas of AI expertise. Object detection (e.g., "Is there any bike ?"), object recognition (e.g., "What is the mustache made of?"), object localization (e.g., "What is just under the trees?"), inference derivation (e.g., "Is this person expecting company?") and general knowledge reasoning (e.g. "Does this person have 20/20 vision?"). Answers to these questions lie in a spectrum ranging from simple "yes/no" to "numbers" and even "colors." Most of the answers can be modeled as a multiple-



What color are her eyes?
What is the mustache made of?



How many slices of pizza are there?
Is this a vegetarian pizza?



Is this person expecting company?
What is just under the tree?



Does it appear to be rainy?
Does this person have 20/20 vision?

Figure 1.1: Few Inputs in Visual Question Answering(VQA) task [15]

choice task. Hence, it requires the AI to select between a predefined list of answers.

The authors of [15] also presented a large dataset along with a strong baseline model on the same. The dataset consists of 204,721 real images from the MS COCO dataset and 50000 scenes from a synthetic dataset. VQA Dataset contains at least three questions per image with ten answers per question, which sums to 760K questions and over 10M answers in total. The massive size of this dataset provides enough data for the task in hand.

Baselines for VQA include random selection, nearest neighbor, and deep learning-based models. For deep learning-based methods, they modeled this task as a classification over 1000 classes. Best performing models consist of 2 parallel channels for vision (image) and language (question) with few fully connected layers at the end. Image channel provides image embeddings in the latent dimension using VGGNet, a deep convolution-based architecture. Whereas LSTM based encoder is used for question embeddings. These two separately computed embeddings are merged via pointwise multiplication before passing on to fully connected layers. These models were able to outperform both the vision-alone and language-alone baselines with overall accuracies of 58.16% and 63.09% respectively for open-ended and multiple-choice questions. However, these baselines performed significantly worse than human-level understanding, thereby giving enormous growth potential to this area of research.



(a) Input image

Original	'What color is the frisbee?'	white
LogEq	'Is the frisbee white?'	no
Mutex	'Is the frisbee black?'	yes
Nec	'Is there anything white?'	no

(b) Implications answered incorrectly

Figure 1.2: Example of inconsistency in VQA models

Ideally, a VQA system should be equipped with the ability to extract useful information (with reference to the question) by looking at the image. To answer these questions correctly, the system should not only identify the color, size, or shape of objects, but may also require general knowledge and reasoning abilities.

Previous works [9, 10] have pointed out the strong language prior present in the VQA dataset. This could result in false impression of good performance by many state-of-the-art models, without them actually understanding the image. For instance, answering any question starting with "What sport is" by "tennis" results in 41% accuracy. Moreover, citing the 'visual priming bias' present in the VQA dataset, questions starting with "Do you see a .." result in "yes" 87% of the time.

Many recent works [11, 13, 14] have shown that despite having high accuracy on questions present in the dataset, these models perform poorly when similar questions are asked and hence are not robust enough to be deployed in the real world. Fig 1.2 shows the inconsistent nature of VQA models. Despite answering the original question correctly, the model fails to answer questions which are implied by the original question answer pair. This shows that models learn from language biases in the dataset rather than correctly understanding the context of the image. The inconsistency problem is shown in Fig 1.2. Even though the model [2] correctly answers the original question, it fails to answer any of the 3 generated implications correctly.

We believe that any model can be taught to unlearn these language priors and better understand the content of the image by enforcing consistency among the predicted answers. In this paper, we present and demonstrate a cyclic training scheme to solve the above mentioned problem of inconsistency. Our framework is model independent and can be integrated with any VQA architecture. The framework consists of a generic VQA module and our implication

generation module tailored especially for this task.

Our framework ensures consistent behaviour of VQA module while answering different questions on the same image. This is achieved in two steps: Implication generator module introduces linguistic variations in the original question based on the answer predicted by the VQA model. Then, the model is again asked to answer this on-the-fly generated question such that it remains consistent with the previously predicted answer. Thus, the VQA architecture is collectively trained to answer questions and their implications correctly. Using the rule-based approach in [11], we calculate the consistency of different state of the art models and show that our framework significantly improves consistency without harming the performance of the VQA model.

We observe that there is no benchmark for consistency, which perhaps is the reason for limited development in this area. Hence, to promote robust and consistent VQA models in the future we collect a human annotated dataset of around 30k questions on the original VQA v2.0 validation dataset.

In later chapters, we demonstrate the quality of these generated questions. We provide a baseline of our implication generator module for future works to compare with. We also perform a comparative study of the attention maps of models trained with our framework to those of baselines. We observe significant improvement in the quality of these attention maps. This proves that by learning on these variations, our framework not only improves the consistency of any generic VQA model but also achieves a stronger multi modal understanding of vision and language.

To summarize, our main contributions in this thesis are as follows -

- We propose a model independent cyclic framework which improves consistency of any given VQA architecture without degrading the architecture's original validation accuracy.
- We propose a novel implication generator module, which can generate implications $G : (Q, A) \rightarrow Q_{imp}$, for any given question answer pair.
- For future evaluation of consistency, we provide a new VQA-Implication dataset. The dataset consists of ~30k questions containing implications of 3 types - Logical Equivalence, Necessary Condition and Mutual Exclusion.

Chapter 2

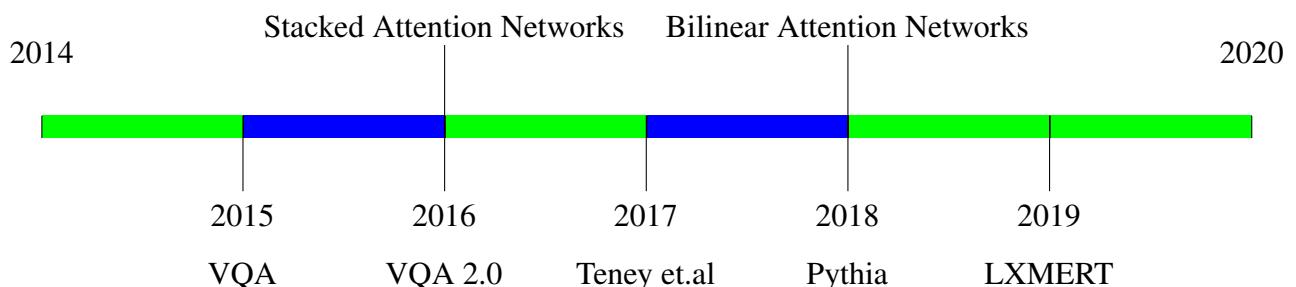
Literature

Like any other new challenge, when VQA was launched, there was a lot of excitement in the vision community. The developer team also started an annual VQA challenge where teams from all over the years could compete and achieve better results. During the first 2-3 years, everyone focused on beating the state-of-the-art results. People came up with very complex attention mechanisms that concentrated on a particular region of the image as asked in the question.

In this chapter, we shall discuss how the VQA task has progressed over the years. More specifically, we will discuss some of the techniques and mechanisms which have been employed for best results, the changes in the VQA dataset and the more recent findings and progressions of the vision community in this field.

2.1 Deep Learning based VQA methods

When VQA was born, the main challenge was combining the image and feature vectors. With high accuracy feature extraction models like ResNet50 already in place, combining the features accurately was of utmost importance. Some of the models which achieved the then state-of-the-art results include [16–20].



2.1.1 Stacked Attention Networks

Hierarchical Co-attention (HieCoAtt) [21]

In this attention-based VQA model, both the image and the question were co-attended to predict an answer. Specifically, it hierarchically separated the question: at the word-level, phrase-level, and entire question-level. Using image features, it then created image-question co-attention maps at all three levels. These features were then combined recursively to get the final output.

Multimodal Compact Bilinear Pooling (MCB) [16]

This model won the VQA challenge in 2016. It used multimodal compact bilinear pooling to combine the image and feature vectors and then passed it through fully connected layers to produce output. Also, this model employed ResNet to extract image features whereas the previous models used VGGNet.

2.1.2 Teney et. al [1]

This model employs a Joint embedding approach to achieve state-of-art results in the 2017 VQA Challenge. This relatively simple deep neural network architecture is carefully selected for performing on the VQA v2 benchmark [9]. While this approach is derivative of many general VQA methods, key technical innovations have greatly enhanced the performance. It implements joint RNN/R-CNN embedding of question/image with image-attention guided by the question.

The questions are tokenized and then vectorized to give 14x300-dimensional vectors. These vectors are initialized with *GloVe* word embeddings for better performance. These resulting embeddings are then passed through GRU. The image input is passed through ResNet CNN within a Faster R-CNN framework to obtain a Kx2048 sized vector for K image locations. *VisualGenome* dataset is used to pre-train and extract top-K objects in the images during the preprocessing step. Top-down attention is used as a question-guided attention mechanism. For each location in the image, attention weight is calculated by passing the concatenated question and image embeddings through a nonlinear layer. After applying attention to the image, the element-wise product is taken to fuse these two modalities. Treating VQA as a multi-label classification task, they used sigmoid for each class instead of softmax. Cross-entropy loss between soft ground-truth targets and these predicted scores provides richer training signals than binary

outputs. Smart shuffling of data during training and the use of larger mini-batches enhanced the model further.

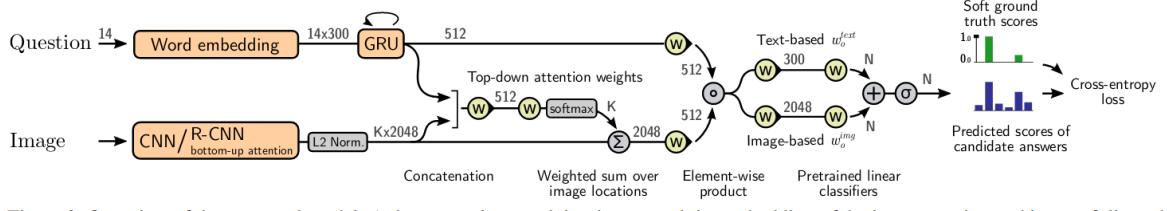


Figure 2.1: Model overview for Teney et. al [1]

2.1.3 Pythia v0.1 [2]

This model was the winning entry of the 2018 VQA challenge. The overall structure of the model was the same as that of Tenny et. al with minor changes in activation functions to fine-tune features. Moreover, they used ensemble learning over 30 models trained on different datasets.

2.1.4 Bilinear Attention Networks [3]

To reduce the computational cost of learning attention distributions, the authors of [3] proposed Bilinear Attention Networks, whereby different attention maps are built for each modality. Further, low-rank bilinear pooling extracts the joint representations for each pair of channels. Structure of a two-glimpse BAN is illustrated in fig 2.2

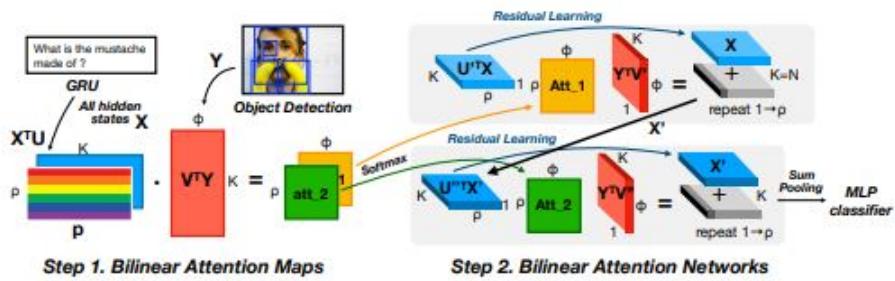


Figure 2.2: Overview of two-glimpse BAN [3]

2.1.5 LXMERT [4]

Based on recent developments in NLP, transformers are now becoming the standard for sequential data models overtaking LSTMs in accuracy. The LXMERT (Learning Cross-Modality Encoder Representations from Transformers) framework is built upon this idea. It uses three encoders - Object Relationship Encoder, Language Encoder and Cross Modality Encoder. LXMERT is the current state-of-the-art model in VQA with a test accuracy of 72.5%. The model architecture of LXMERT is shown in Fig [4].

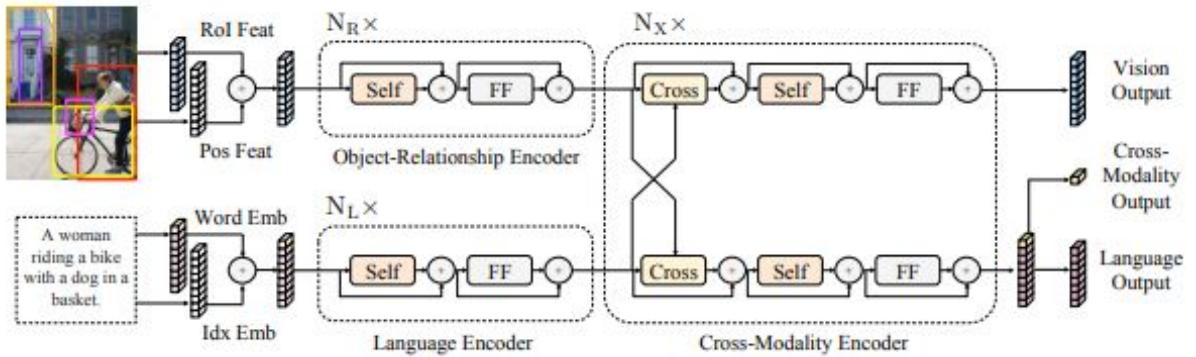


Figure 2.3: LXMERT Architecture [4]

The results of all the above-listed models are summarized and listed in the table 2.1.

Model	Method	Accuracy (in %)	Venue
VQA-baseline [15]	LSTM+CNN	57.75	ICCV 2015
HieCoAtt [21]	Hierarchical Attention	62.1	NIPS 2016
MCB [16]	Bilinear attention	64.2	CVPR 2016
Teney et. al [1]	FasterRCNN+GloVe	63.15(VQA-v2)	CVPR 2018
Pythia v0.1 [2]	[1]+ensemble	72.27(VQA-v2)	VQA challenge 2018
BAN [3]	Residual attention	70.04(VQA-v2)	NIPS 2018
LXMERT [4]	Transformers	72.50(VQA-v2)	EMNLP 2019

Table 2.1: Summary of deep learning models in VQA

2.2 Variations in VQA

Over the years, VQA models have become increasingly complex, employing convoluted attention mechanisms. Apart from a focus on attention maps, people have also used extra information such as generating image captions. Furthermore, derivatives of the original VQA task have emerged in recent years. In this section, we shall discuss a few such variations and techniques which played a crucial part in our problem formulation.

2.2.1 VQA with Explanations [5]

This paper introduced the task of generating relevant explanations for the given answers by the model. To this extent, they first introduced a new dataset VQA-E which contained explanations along with the answers. The dataset is derived from the VQA v2 dataset and uses image captions as explanations for the questions.

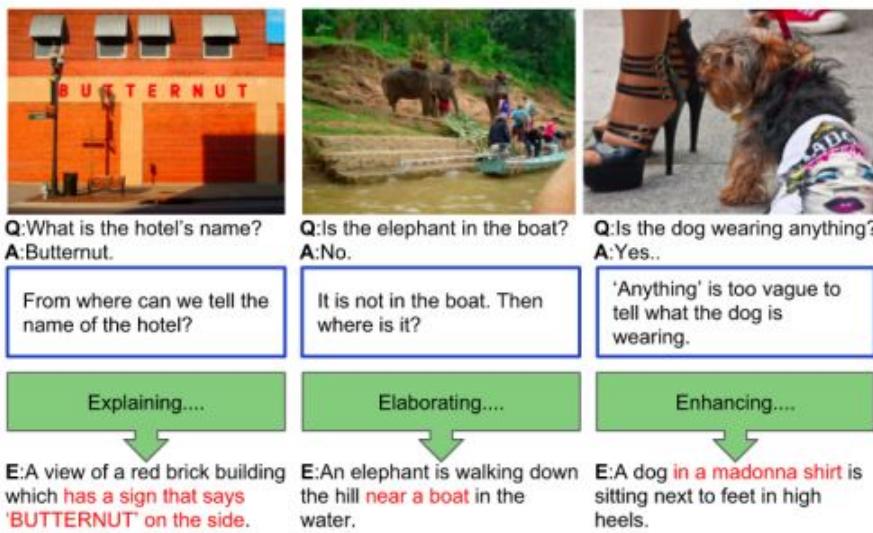


Figure 2.4: Overview of VQA-E task [5]

In the past, there was a lot of speculation that deep learning models learn based on statistical bias instead of looking at the image, and this affected accuracy adversely. Using explanations along with answers helped resolve this bias as irrelevant explanations to the question are awarded negatively in this model.

To generate the explanations for the VQA-E dataset, the authors first combined the question answer pair (Q, A) to form a statement S . They also used a caption generator to get a caption C . Then S and C were fused using constituency trees to get an explanation E . Due to

a diverse set of questions, generating good reasons for all questions was not possible. To tackle this problem, the authors removed the questions from the dataset for which good explanations weren't produced.

The authors also proposed a basic model which uses a simple attention mechanism and LSTM cells to generate captions. They trained their model on the VQA-E dataset and reported results that will be used as a baseline for future work.

2.2.2 Captions to aid VQA [6]

In this paper, the authors first generated captions for the image and then used those captions to fine-tune their VQA model to achieve better results. Caption embeddings were utilized to adjust the visual top-down attention weights for each object. The important distinguishing feature of this paper was that they used attention mechanisms on not only the image but also the caption, i.e. more weight was given to the words which helped in answering the question.

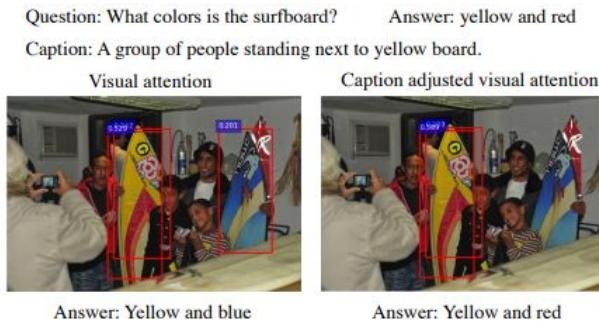


Figure 2.5: An illustration of adjustment using caption attention [6]

The grey-scale levels in fig 2.6 show the weights of the words in the captions. In fig 2.5 the question-relevant caption helps the VQA module to focus on the yellow board only.

2.2.3 VQG: Generating Natural Questions About an Image [7]

Moving beyond describing the content of images, this paper introduced the task of generating a natural and engaging question given any image. They believe that learning to ask questions is essential for any AI to master. Learning to ask the right question shows a deeper understanding of image and general reasoning over knowledge. Learning this ability has many applications in conversational systems to interactive environments.

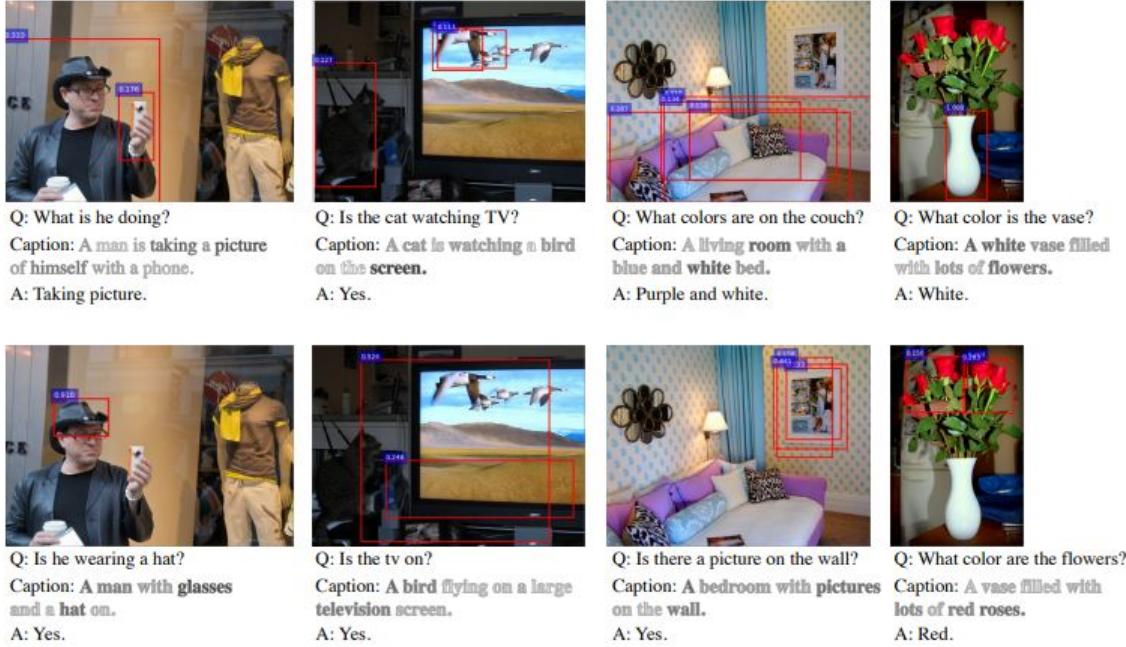


Figure 2.6: Examples of generated question-relevant captions [6]

Questions generated under this task should be verifiable visually. The key idea is to start a conversation with a human, So questions which can be answered by merely looking at the image are not of interest for this task. Fig 2.7 demonstrates the scope of questions in this task.



Figure 2.7: Example right and wrong questions for VQG [7]

To this end, they provided questions for three datasets: *MSCOCO*, *Flickr*, and *Bing*. In total, 15000 images with 75000 questions covering a wide range of visual events. 5000 images from each dataset are picked and five questions per image were collected by crowdsourcing on Amazon Mechanical Turk.

They proposed several generative and retrieval models to tackle this complex problem. For generative models, best results were obtained using an end-to-end deep learning based method previously used for image captioning. The model consists of an image encoder network using VGG architecture, followed by several fully connected layers. The transformed output serves as an initial state to a Gated Recurrent Unit (GRU). Overall, questions are produced one word

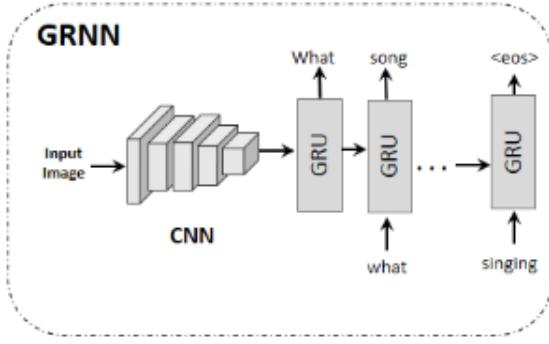


Figure 2.8: Generative Model for VQG [7]

at a time until the EOS token.

Retrieval models were customized for this task to make use of the caption of K-nearest neighbor in the training set. Questions with the highest semantic similarity among K selected neighbors are selected as the output. Although human evaluation is an ideal way to deal with such ill-posed problems, they also proposed metric-based methods to benchmark the progress.

2.2.4 iVQA: Inverse Visual Question Answering [8]

Deriving motivation from [7], this paper proposed the inverse problem for Visual Question Answering (iVQA), which is to infer a question Q for which a given answer A holds, in context of an image I . This work differs from VQG in the sense that the generated question is conditioned on the answer. By doing so, iVQA aims to produce more relevant questions.

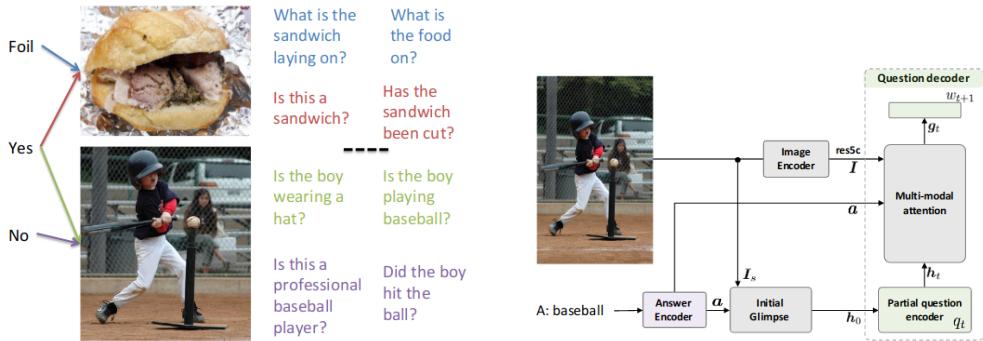


Figure 2.9: (a) Illustration of iVQA task (b) Architecture for iVQA model [8]

Fig 2.9a demonstrate a typical example of iVQA. Input to the model is an image (Sandwich and Baseball) and answers (foil, Yes, and No). This task expects to generate questions like "What is the sandwich laying on?" (foil), "Is the boy wearing a hat?" (Yes), which enhances image understanding and provide opportunities for much more complex tasks like counterfactual

reasoning.

The authors of [8] proposed a deep neural network for iVQA. The model architecture is shown in fig 2.9b. It consists of three sub-networks to tackle different modalities - an answer encoder to encode answer into a latent vector, an image encoder for image embeddings, and a question decoder to generate the output question, one word at a time. Novelties in this architecture include the proposed Dynamic multimodal attention for question decoding. At each time step, given the partially encoded question, answer and image embeddings, this attention model learns to focus on a region of the image, critical for this step. Based on these attended image and answer features, this model predicts the next word.

Moreover, this study proposed a new ranking-based metric for evaluating iVQA. The conditioning score $p(q|I, a; \theta)$ used for ranking is related to multiple-choice VQA. This metric could significantly contribute to diagnosing the strengths and weaknesses of Image understanding models. Ablation studies on this task show that if posed as a dual problem, iVQA can help improve significantly on the VQA task.

2.3 Language Prior in VQA

2.3.1 Elevating Image Understanding in VQA [9]



Figure 2.10: Few illustrations of balanced VQA dataset [9]

Analysis of the earlier models in VQA showed that there was a language bias in the original dataset, which led to models learning from those biases instead of looking at the image.

To tackle this problem, VQA 2.0 was released which created complementary pairs to decrease bias. More specifically, for every image, question and answer triplet (I, Q, A) , a complimentary image I' and answer A' were created. An example of such a complementary pair is illustrated in fig 2.10.

Supplementary datasets such as Visual7W, CLEVR have been used to fine-tune deep learning models but VQA 2.0 dataset remains the benchmark dataset for the VQA challenge.

2.3.2 Explicit Bias Discovery in VQA Models [10]

The authors of this paper perform a study to discover the statistical biases present in VQA v2 dataset, which VQA models end up learning from rather than really understanding the context of the image. In Fig 2.11, we can see how a particular set of words such as "what, time, day" in the question always result in the answer as "afternoon" irrespective of the image. Their work shows how VQA models are biased by precedents such as gender, or answers which are correct majority of the time. For eg, the answer to "What color is the grass?" would be "green" since that would be the most common general answer.

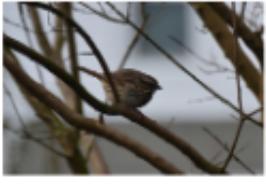
No.	antecedent words	antecedent visual words	consequents
1	what,time,day		afternoon*
2	what,time,day		night*
3	what,time,clock,show		11:30*
4	what,time,year		fall*

Figure 2.11: Existing Biases in VQA models [10]

2.4 Consistency of VQA Models

2.4.1 Inconsistency in VQA Models [11]

Recently, Biases have been observed in the VQA dataset, e.g. 87% of questions starting with "Do you see a ..." has an answer "yes" and answering "tennis" to questions "What sport is ..." results in 41% accuracy [9]. Many state-of-the-art models have found to be exploiting these biases instead of "higher-level reasoning".



How many birds?	A: 1
Is there 1 bird?	A: no
Are there 2 birds?	A: yes
Are there any birds?	A: no

Figure 2.12: Inconsistent VQA predictions [11]

The authors argue that the evaluation of such models should not be based entirely on prediction accuracy. Instead, the relation between several predictions should also be taken into account to measure complete understanding and generalisability i.e., any model answering "no" to "is the rose red?" should be penalized if it answers "red" to "What color is the rose?".

Consider the image shown in fig 2.12, the model answers "no" when asked about "Is there 1 bird?" despite counting the number of birds in original question as one. This inconsistency in QA models indicates that they fails to understand the context and are merely exploiting the datasets instead of looking into the image. This opens up new avenues along this line of coherence and consistency in models. Working along these lines, this paper further proposes to generate implied question-answer pairs from existing dataset and using this to measure the consistency of any model.

Generating Implications

Let any datapoint from VQA dataset be (I, Q, A) where I,Q and A denotes contextual image, question and correct answer to that question respectively. Then, logical implications are defined as $(I, Q, A) \rightarrow (I, Q', A')$, i.e. the answer A' to question Q' can be implied from question-answer pair Q,A given an image I. The authors presented with rule-based system to generate these implications. Mainly, three types of "yes/no" implications are derived :

Logical equivalence: Dependency parser were used to recognize root/subject/object and to detect auxiliary/copula in a question. To generate logically equivalent implications, the original question is formed into a proposition. Adding "do" auxiliaries or moving auxiliary/copula can then ask the appropriate "yes-no" equivalent of the original question. e.g. "Who painted the wall? Man" can be converted into "Did the man paint the wall? yes".

Necessary Condition: One way of deriving necessary conditions from QA pair is to use Heuristics such as converting numerical answer questions like "How many θ " to ask if there are any θ present in the picture. e.g. "How many birds? 2" can be molded into "Are there any

birds? yes". It can also be achieved by asking if picture contains the answer nouns. e.g. "What room is this? bathroom" implies "Is there a bathroom in the picture? Yes".

Mutual exclusion: Antonyms and other plausible answers can be found using Wordnet. Original noun and its antonym are mutually exclusive, this way the model can be asked if antonym is present or not. e.g. changing "Bathroom" to "Kitchen" in fig2.13. It is implied that if a room is bathroom it can't be a kitchen.



(a) Example input image. (b) Example (q, a) pair.

Type	Cov	Example
Logeq	56.8%	Is this a bathroom? Yes
Nec	50.2%	Is there a bathroom in the picture? Yes
Mutex	34.6%	Is this a kitchen? No

(c) Implication types, with coverage and examples.

Figure 2.13: Generated Implications [11]

To smoothen out these generated implications, 4-gram language model could be used to add "a", "the",etc in the questions. Finally, these implications could be used to check the consistency of any QA model. It is observed that models with high accuracy were often performing poorly in consistency. This proves the correctness of the hypothesis drawn by the authors.

They further used simple data augmentation techniques to improve the models. Training the models on the original dataset as well as on implied QA pairs substantially improved consistency while doing equally well on accuracy. However, data augmentation is limited by the kind of implications and could further create other undesirable biases in the system.

2.4.2 SQuINTing at VQA Models [12]

This paper focuses on a small subset of the VQA v2 dataset, the *Reasoning* split. These type of questions require some common knowledge of the world apart from the context of the image. For example, a question "Are the bananas ripe?" requires the model to look at the color of the bananas, and also have the knowledge that greenish-yellow means ripe whereas dark green would mean unripe and yellowish-black would mean stale. Fig 2.14 highlights this problem.

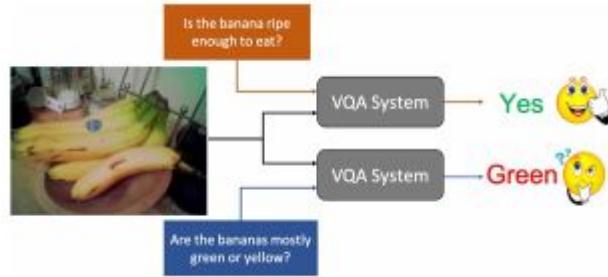


Figure 2.14: Inconsistency in Reasoning questions [12]

To tackle this problem, the authors created a new dataset which included ~3 sub-questions like "Are the bananas greenish-yellow?" for every main reasoning question. With this dataset and architecture shown in Fig 2.15, the authors of [12] improve consistency - if main question is answered correct, sub question should be answered correct - of VQA models on the Reasoning split by 7.8%.

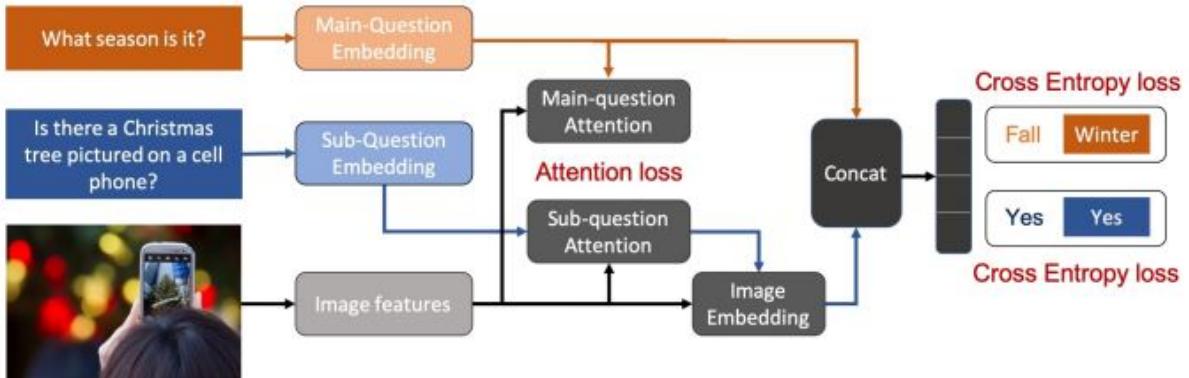


Figure 2.15: SQuINT Architecture [12]

2.4.3 VQA-LOL: VQA under the Lens of Logic [13]

This paper tackles inconsistency among binary i.e. "yes/no" questions. The authors argue that VQA models do not perform well on logical composition of questions, even if they answer the original question correctly. For 2 given questions Q_1 and Q_2 , possible composite question Q^* is defined as:

$$Q^* = Q'_1 \odot Q'_2$$

where, $Q'_1 \in \{Q_1, \neg Q_1\}$, $Q'_2 \in \{Q_2, \neg Q_2\}$

and connective $\odot \in \{\vee, \wedge\}$ (2.1)

For example, given a question "Is the man wearing shoes?", the model correctly answers "No". However, on passing the question "Is the man not wearing shoes?", the same model again answers "No". This problem is shown in Fig 2.16.

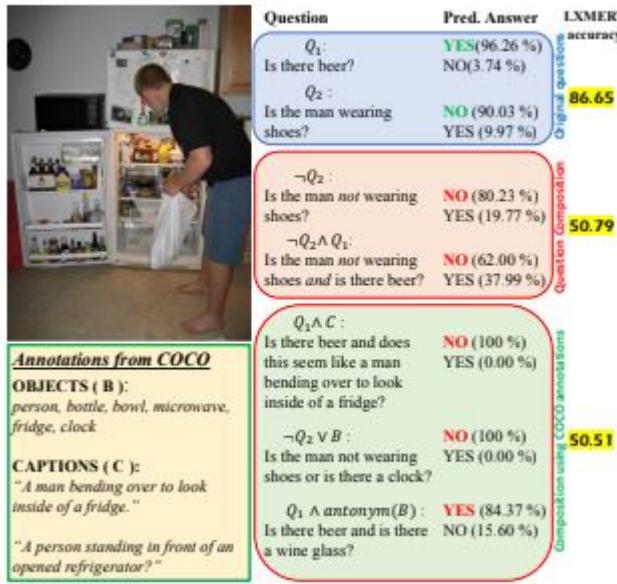


Figure 2.16: Inconsistency in logical composition of questions [13]

Similar to [12], the authors of [13] also provide their own dataset for tackling this problem. They form logical composition of binary questions *VQA-Compose* and *VQA-Supplement*. They also propose a method using this dataset which is dedicated to improve accuracy of logical composition of questions. The method is shown in Fig 2.17.

2.5 Cyclic Training in VQA

Cyclic training for singular modality has been used in the past for tasks such as motion tracking [22] and text-based question answering [23]. For multi-modal tasks such as VQA, cyclic training was first introduced by [14].

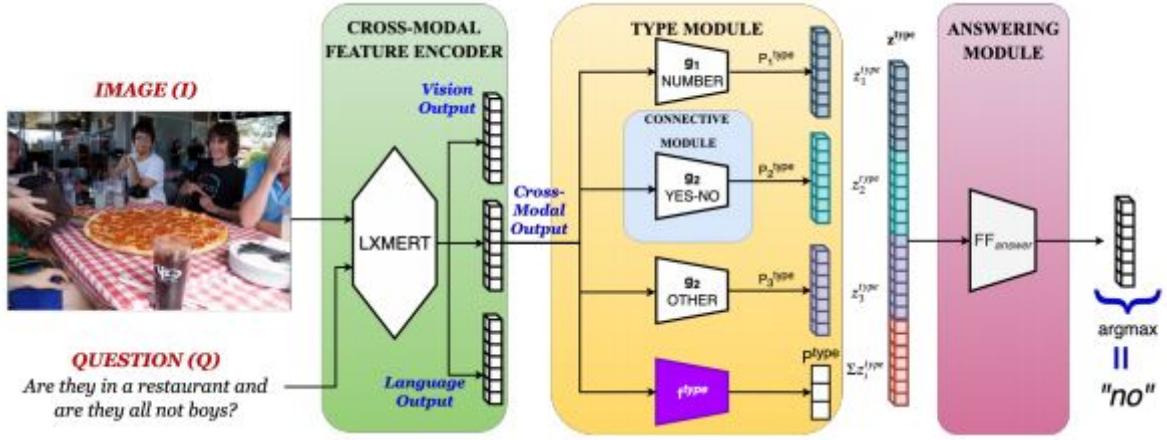


Figure 2.17: VQA-LOL Model Architecture [13]

2.5.1 Cycle-Consistency for Robust VQA [14]

The fundamental foundation of this paper is identifying that VQA models are brittle. Upon asking a rephrased question, the model answers differently even though the meaning of the question remains the same. To tackle this problem, the authors make the following significant contributions -

- A model-independent cycle-consistent training framework.
- New VQA-Rephrasings Dataset
- A consensus score for robustness
- model trained using a cyclic approach achieve state of the art results on VQA 2.0

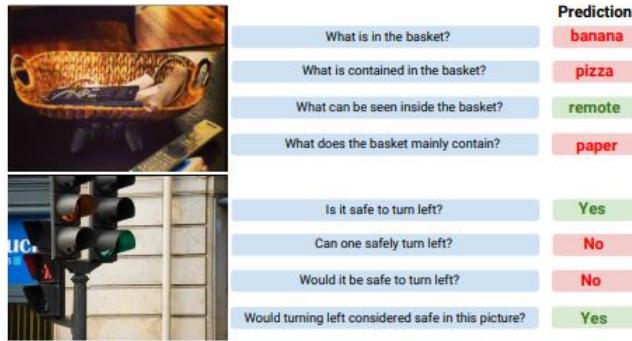


Figure 2.18: Examples of brittleness of VQA Models [14]

Cyclic Training Scheme

Under the cyclic training scheme, the model is trained to answer a question and also generate rephrased variations of questions conditioned on the answer. Then, the VQA model again

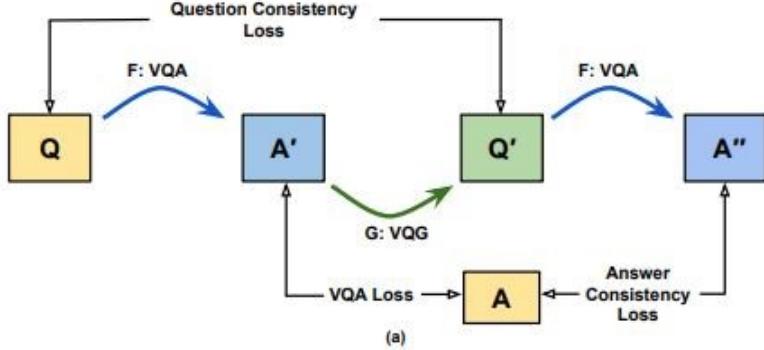


Figure 2.19: Cyclic Consistency model-agnostic framework [14]

answers the newly generated question Q' and the new answer A'' must match the first answer A' and the ground truth answer A . The advantage of using this scheme is that generating questions based on the input image, question and answer provides a stronger multimodal understanding of vision and language.

As seen in Fig 2.19, the VQA module first computes A' given (Q, I) pair. This predicted A' along with input (Q, I) pair when passed through the VQG module generates implied (Q', A'') . VQA is used again to compute A'' of this (Q', I) pair. This forms one iteration of the cycle. Losses on (Q, Q') will drive the learning process of VQG whereas VQA will be guided by losses through (A', A) and (A'', A'') pairs.

Gating Mechanism: Not all the questions generated by the VQG module were coherent with image, question and answer triplet (I, Q, A) . To resolve this, the authors used a similarity score based on cosine similarity with the original question and filtered the newly generated questions using a threshold.

Late Activation: The VQA and VQG models under the cyclic scheme are trained separately before combining them. If the models are cyclically trained from scratch, they can use statistical bias to fit on the training data instead of looking at relevant portions of the image for answering.

Loss Formulation: The overall loss consists of three components. First, the VQA Model loss L_F between the first answer A' and ground truth answer A . Second, the visual question generation loss L_G between original question Q and generated question Q' . Lastly, the cycle-consistency loss L_{cycle} between first answer A' and second answer A'' . The overall loss is given by -

$$Loss_{total} = L_F(A, A') + \lambda_G L_G(Q, Q') + \lambda_C L_{cycle}(A', A'') \quad (2.2)$$

$L_F(A, A')$ and $L_G(A', A'')$ are cross-entropy losses and $L_{cycle}(Q, Q')$ is sequence generation loss. λ_G , λ_C are tunable hyperparameters.

VQA Rephrasings Dataset

The authors used the validation part of the VQA 2.0 dataset and generated three rephrasings per question. Note that the generated rephrasings must have the same answer as the original question. The final dataset consists of 162,016 questions spanning 40,504 images.

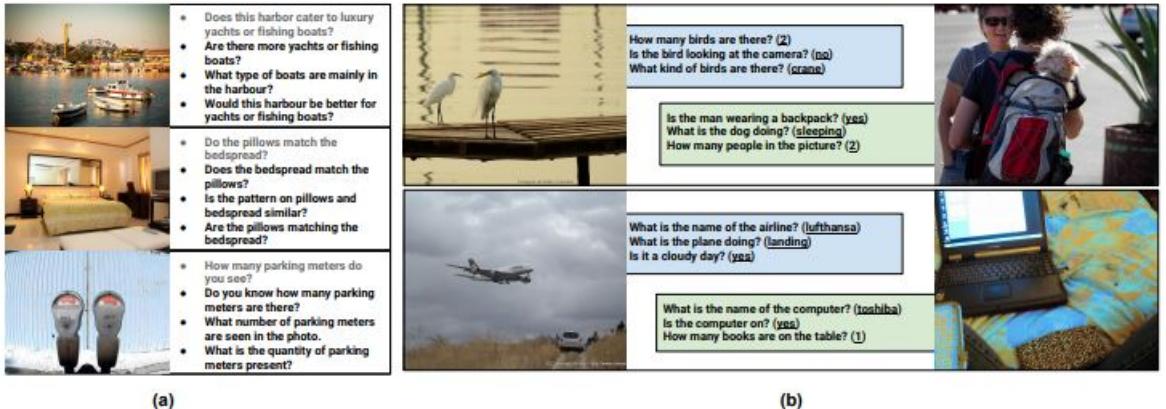


Figure 2.20: Examples of VQA-Rephrasings dataset [14]

Some examples are shown in fig 2.20. (a) Illustrations from the VQA-Rephrasings dataset. In each set, The first question - in gray, is the original question from VQA v2.0, the following questions are the rephrased ones. (b) Examples of questions generated by the VQG module based on the given answer.

Consensus Score: The authors proposed a consensus score $CS(k)$ to quantify robustness of models. For every group of questions Q_P containing m rephrasings, all subsets of size k are sampled. The consensus score $CS(k)$ is given by the ratio of the number of subsets with all correct answers to the total sampled subsets of size k .

2.6 Conclusion

In this chapter, we first discussed the progress in VQA over the years. We discussed how supplementary information has been used in VQA to improve accuracy and some variations of VQA. In addition, we also discussed about the existing language biases in VQA, their inconsistency and a few recent works in this area.

The authors of [11] proposed evaluating consistency of these models and a simple data augmenting technique. However, augmentation requires additional training dataset and it limits the scope of implied questions to this additional dataset. We in-turn propose a generative model based solution without these limitations. Model based solution for improving consistency has been proposed by [12, 13] but these target only a specific category of questions such as reasoning or binary questions. Unlike these, we show that our approach works better on the entire VQA v2.0 dataset rather than a small subset of it. Similar to [14], our framework is also model-independent and can be used for any VQA architecture. However, their aim was to make VQA models more robust to linguistic variations through rephrasings. Our aim, through our approach, is to make the models more consistent to not just rephrasings like in [14], but also on implications.

Chapter 3

Approach

During our literature survey, we figured some of the problems faced by Visual Question Answering systems. Over time, VQA models have evolved to become too complex and intricate. From stacking more and more attention layers to ensembling tens of models, there is a monotonous trend of throwing data into more and more complex models and fit it for that dataset making the model brittle.

As pointed out by [9, 10], like any other dataset, the VQA dataset is full of language prior. For instance, 39% of questions starting with "How many" have "2" as the answer. It was observed that models performing well were learning to identify these priors instead of looking into the images. Therefore, trained models were not able to generalize well on-the-wild.

These models were also observed to be inconsistent among predicted answers [11, 13]. It would answer "no" to "is there any bike?" despite answering "1" to "How many bikes are there?". These flaws and shortcomings would limit the scope of such systems to only the training distributions. Many approaches were tried in the past to tackle these problems through explanations, captions, and even augmenting the original VQA dataset with implications. However, we feel that this line of research still holds a lot of potential in Visual Question Answering.

In this chapter, we present a novel approach to solve the problem of inconsistencies in VQA models using implications. Firstly, we give a formal definition of implication, and throughout this thesis, we stick to this definition. Then, we design a deep-learning based module to automatically generate these implications. Later, we propose a novel cyclic framework to train any generic VQA model with this designed implication generator module.

3.1 Implications

Throughout this thesis, implications are defined as questions Q' which can be answered by knowing the original question Q and answer A without the knowledge of the context i.e. image I . For example, given the original QA pair ("What color are the flower pots?", "Brown"), one of the implication would be ("Are the flower pots brown ?", "Yes"). As defined by [11], we categorize these implications into 3 types - logical equivalence, necessary condition and mutual exclusion.

We use the rule-based approach in [11] to generate implications on entire VQA v2.0 dataset. We will refer to this as our implication dataset. This rule-based method is unable to create all 3 implications for every QA pair, especially on yes/no type questions. Due to these restrictions by the rule-based approach, implication dataset contains implications from about 60% of the original dataset. Moreover, all generated implications are of 'yes/no' type, this serves as a strong prior for our implication generator module.



Original	'What color is the hydrant?'	'red'
LogEq	'Is the hydrant red ?'	'yes'
Mutex	'Is the hydrant green?'	'no'
Nec	'Is there anything red in the picture?'	'yes'



Original	How many people are in the image'	'4'
LogEq	'Are 4 people in the image?'	'yes'
Mutex	'Are 5 people in the image?'	'no'
Nec	'Are any people in the image?'	'yes'

We show two examples of our implications dataset, generated by this rule-based approach. One thing to note is that given the answer to the original question, one doesn't need to look into the image for answering the implications. Additional details about the implication dataset can be found in Chapter 4.

3.2 Implication Generator Module

The role of this module is to generate implications of a given QA pair. This can be formulated as a transformation $G : (Q, A) \rightarrow Q_{imp}$ where Q_{imp} is the generated implication. In the VQA setting, this QA pair is provided by the VQA model. Any generic VQA model takes (Q, I) to predict A' where Q is the original question, I is the image and A' is the predicted answer. Our implication generator takes as input, the learned question encoding of the original question Q , the predicted answer scores A' and a knob (as one hot vector) to select between implication category.

There has been a thorough study of Natural Language generation in NLP, such as [24–27]. [25] extracts keywords from knowledge graphs and then formulate question generation from these keywords as Seq2Seq translation problem. [26] tackles the question generation problem from Reinforcement Learning point of view. They consider generator as an actor trying to maximise BLEU score as it’s reward function. [24] propose a Transformer based Seq2Seq pretraining model which beats the current state-of-art in many summarization and question generation tasks. To the best of our knowledge, we are the first ones to propose an implication generator module to improve consistency of any VQA architecture.

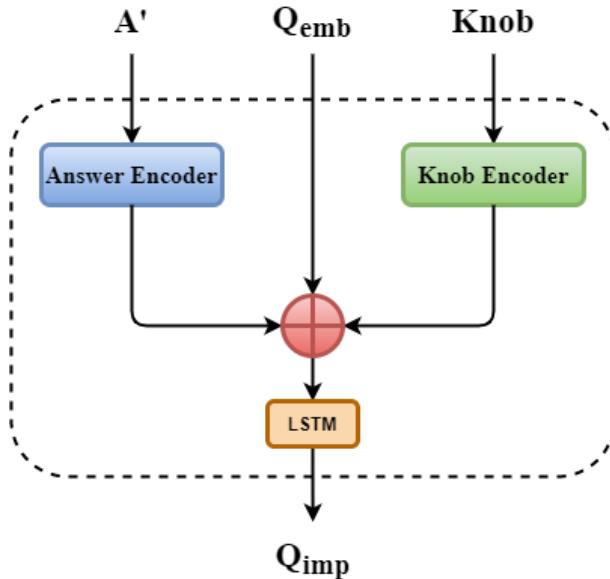


Figure 3.1: Detailed architecture of our Implication generator

The implication generation module consists of three linear encoders that transform question encoding obtained from VQA model, the predicted answer scores, and the knob to lower dimensional feature vectors. These three inputs are then added together, and passed through

a single layered LSTM with hidden size of 1024. This LSTM is trained to generate implication and optimized by minimizing the negative log likelihood with corresponding ground truth implication from the implication dataset.

$$L_Q(X, C) = -\log\left(\frac{\exp(X[C])}{\sum_j \exp(X[j])}\right) \quad (3.1)$$

where X is the prediction and C is the ground truth class. One thing to note is that we use answers scores instead of any particular answer label. This takes question with multiple correct answers into account. Also, this provides a distribution over the entire set of answers which is slightly rich and dense signal to learn from.

3.3 Knob Mechanism

Instead of using an implied answer selected randomly from (*yes, no*) as input to the implication generator module, we use a three way knob to switch between logical equivalence, necessary condition and mutual exclusion. This helps the model to have better control over the generated implications.

In our training dataset, implications from two categories - logical equivalence and necessary condition have 'yes' as the correct answer. While training the implication generator using implied answer, we noted that model tends to generate necessary implications when provided 'yes' as the implied answer. We believe that generating a necessary condition is easier as compared to logical equivalence and without having any control signal, model might collapse to generate necessary implications all the time. Hence, we provide this control signal in the form of a one hot vector between the three implication categories.

3.4 Cyclic Framework

To integrate our implication generator module with any VQA module, we use a cyclic framework. The confidence score over answers generated by the VQA module is used by the implication generator module. The implications are then passed as question to the VQA module, along with the image I to give implied answer A_{imp} . This enables the VQA module to learn on these implications and improve its consistency.

Training such cyclic framework could be tricky, so inspired by [14], We incorporate gating mechanism and late activation in our cyclic architecture. Instead of passing all implied questions, we filter out undesirable implications which have cosine similarity less than threshold T_{sim} with the ground truth implication. Also, as part of the late activation scheme, we disable cycle loss before A_{iter} .

$$\text{CosineSimilarity}(X_1, X_2) = \frac{X_1 \cdot X_2}{\|X_1\|_2 \|X_2\|_2} \quad (3.2)$$

We use three loss functions in our architecture, namely VQA loss L_{vqa} , question loss L_Q and implication loss L_{imp} . L_{vqa} and L_{imp} are the standard binary cross-entropy (BCE) loss, between predicted answer A' and ground truth A^{gt} , and A_{imp} and A_{imp}^{gt} respectively. L_Q , as defined above, is the log-likelihood loss between generated implication Q_{imp} and ground truth implication Q_{imp}^{gt} . Combining the three losses with their respective weights, we get total loss L_{tot} as:

$$L_{tot} = L_{vqa}(A', A^{gt}) + \lambda_Q L_Q(Q_{imp}, Q_{imp}^{gt}) + \lambda_{imp} L_{imp}(A_{imp}, A_{imp}^{gt}) \quad (3.3)$$

where λ_Q and λ_{imp} are weights for L_Q and L_{imp} respectively. Fig 3.2 shows an abstract representation of our cyclic framework. Given an input image I and question Q , a VQA model is used to predict the output answer A' . Then our proposed Implication generator transforms the original question Q to Q_{imp} with the help of A' and a control knob. This generated implication along with the input Image is passed to the VQA model to obtain answer A_{imp} to the implication. A' and A_{imp} are trained with their respective ground truth values.

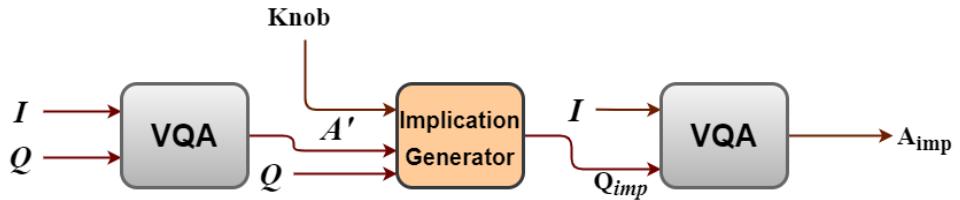


Figure 3.2: Proposed Model Architecture

Clearly, this proposed framework uses no prior information about the VQA model and hence can be applied on any generic VQA model. This makes our method model independent. With our implication generation module, we believe to introduce linguistic variations in the

original question. To encourage consistent behaviour of the VQA model, we enforce that the VQA model answers this on-the-fly generated implication correctly.

In the next chapter, we demonstrate several experiments to show the improvement in consistencies of 3 state-of-the-art VQA models after training with our approach. We also believe that our implication generator module introduces stronger linguistic variations than simple rephrasings of the original question which should enforce robustness along with consistency. Thus we discuss our models' performance on the VQA-Rephrasings dataset as well.

Chapter 4

Experiments and Results

In this section we report and compare the results of our model against some past state-of-the-art VQA baselines. We also show the importance of knob mechanism in our implication generator, quantitatively. We use the VQA v2.0 dataset for training and evaluating our model’s VQA performance. The VQA v2.0 training split consists of 443,757 questions on 82,783 images and the validation split contains 214,354 questions over 40,504 images.

To train and evaluate our implication generator module, we use the implication dataset made by the rule-based approach in [11]. This dataset consists of 531,091 implied questions in training split and 255,682 questions for the validation split.

We also evaluate our model’s consistency performance on human annotated VQA Implications dataset which consists of 30,963 questions. For this dataset, we randomly select 10,500 questions from the VQA v2.0 validation set and create 3 implications(logeq, nec and mutex) per question.

For robustness performance, we evaluate our models on the VQA Rephrasing dataset provided by [14]. The dataset consists of 121,512 questions by making 3 rephrasings from 40,504 questions on the VQA-v2 validation set.

4.1 Consistency performance

We define consistency of any VQA model as it’s ability to answer the implications of a question correctly, if it correctly answers the original question. Implications are generated on the correctly answered questions from validation VQA v2.0 dataset, and consistency score is calculated as the fraction of correct predictions to total implications. These generated implications are bi-

Method	Val acc	Consistency(rule-based)				Consistency(VQA-Implications)			
		Log Eq	Necc	Mut Ex	Overall	Log Eq	Necc	Mut Ex	Overall
BUTD [28]	63.62	64.3	71.1	59.8	65.3	67.45	72.67	61.31	67.14
BUTD + IC (ours)	62.57	88.5	96.7	77.0	88.1	84.56	83.56	49.01	74.38
BAN [3]	65.37	67.1	77.6	61.1	69.0	65.76	74.27	59.68	66.57
BAN + IC (ours)	64.28	89.3	97.9	79.8	89.6	84.76	84.85	54.23	74.61
Pythia [2]	64.70	69.7	76.4	67.7	70.0	70.66	77.57	64.42	70.89
Pythia + IC (ours)	65.60	88.7	97.6	79.0	88.7	85.66	87.20	56.80	76.55

Table 4.1: **Consistency performance on rule-based validation and VQA-Implications dataset.** Consistency is defined as percentage of correctly answered implications, generated only on correctly answered original questions. All the models trained with our approach outperform their respective baselines in both categories, keeping the validation accuracy almost same.

nary yes/no questions, and hence randomly answering them would give about 50% consistency score. In order to show the model independent behaviour of our proposed method, we evaluate consistency of 3 VQA models: BUTD, BAN, Pythia. We use the open-source implementation of these models for training and evaluation. These models are trained with hyperparameters proposed in respective papers.

BUTD [28] uses bottom up attention mechanism from pretrained Faster-RCNN features on the images. Visual Genome [29] dataset is used to pretrain and extract top-K objects in the images during the preprocessing step. This model won the annual VQA challenge in 2017. For training BUTD, we used the fixed top-36 objects RCNN features for every image. Their model achieves 63.62% accuracy on the VQA 2.0 validation split.

BAN [3] uses bilinear model to reduce the computational cost of learning attention distributions, whereby different attention maps are built for each modality. Further, low-rank bilinear pooling extracts the joint representations for each pair of channels. BAN achieves 65.37% accuracy on the VQA 2.0 validation split.

Pythia [2] extracts image features from detectron also pretrained over visual genome. It also uses Resnet-152 features and ensembling over 30 models, but we didn’t use these techniques in our study. Glove embeddings are used for question and its implications. Pythia was

the winning entry of 2018 VQA challenge and achieves 65.59% accuracy on the VQA 2.0 validation split.

As seen in Table 4.1 All the 3 models achieve an average consistency score of ~70%. i.e. they fail 30% of the times on implications of correctly predicted questions. Intuitively, Nec-implication serves as the necessary condition which the models should know in order to answer the question. For eg: In order to answer "How many birds are there?", they should understand if "Are there any birds in the picture ?" Consistency score of ~75% Nec-implication shows the lack of image understanding in these models. Using our approach, the 3 models achieve ~97% on Nec-implication.

4.2 Attention Map comparison

As a qualitative analysis, we also compare the attention maps of [2] with our approach. As we can see in Fig 4.1, the attention maps generated by our approach are significantly better than those of [2]. Pythia model answers 'black' for 'what color are the skis?' without actually looking at the skis. This is further highlighted when the model fails to answer any implication correctly. After training pythia in our cyclic framework, which encourages it to remain consistent while answering, it's attention is focused on the ski and hence it is able to correctly answer all the implications. This shows that multi-modal understanding of vision and language is enhanced using our approach. Some more examples are shown in Section 4.7.

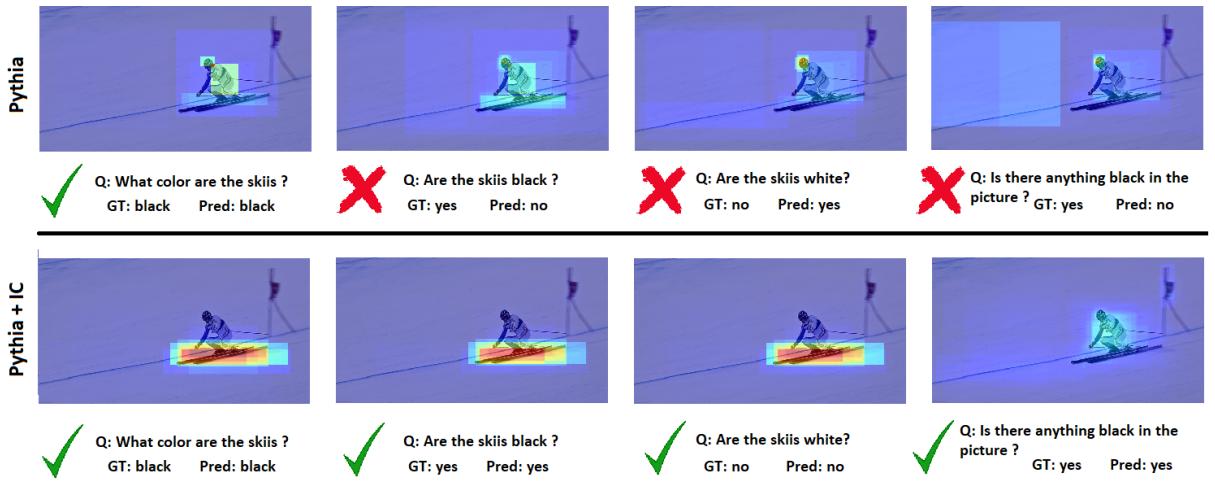


Figure 4.1: Qualitative example showing improvement in attention maps for Pythia

Method	Consistency (rule-based)	Consistency (VQA-Imp)
BUTD + DA	93.1	74.24
BUTD + IC (ours)	88.1	74.38
BAN + DA	87.6	74.33
BAN + IC (ours)	89.6	74.61
Pythia + DA	89.7	76.19
Pythia + IC (ours)	88.7	76.55

Table 4.2: **Consistency comparison of data augmentation vs our approach.** VQA-Imp denotes our VQA-Implications dataset and DA stands for models finetuned on rule-based training implications. Even though our models lack on rule-based dataset, they consistently outperform their respective baselines on the VQA-Implication dataset.

4.3 Data Augmentation

Since we are using an extra dataset (Rule-based implications) apart from VQA-v2 to train our models, we also compare our models’ consistency with models finetuned with data augmentation. Table 4.2 summarizes the results. Better performance of our models on the human annotated VQA-Implications dataset shows that models trained with our approach generalize better and hence would do better than data augmentation in the outside world.

4.4 VQA Rephrasings

We also evaluate our models’ robustness performance on the VQA-Rephrasings dataset introduced in [14]. A rephrasing is defined as a variation of the original question keeping the answer exactly same. Note that just like the models in [14], we also do not train our models on the VQA-Rephrasings dataset. The results in Table 4.3 show that training models with our approach also improves robustness of models. This is consistent with the hypotheses that our models learn to improve on a stronger linguistic variation than rephrasings by learning on implications and hence improvement in robustness is expected.

Method	Val Acc	VQA-Rep Acc
BUTD	63.62	53.76
BUTD + IC (ours)	62.57	54.54
BAN	65.37	54.60
BAN + IC (ours)	64.28	55.56
Pythia	64.70	56.49
Pythia + IC (ours)	65.60	57.03

Table 4.3: **Robustness performance on VQA-Rephrasing dataset.** VQA-Rep denotes VQA-Rephrasing dataset. All models trained with our approach consistently outperform their respective baselines.

4.5 Implication Generator Performance

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	CIDEr
Pythia + IC	0.627	0.520	0.443	0.381	0.632	0.288	3.343
Pythia + IC + Knob	0.785	0.715	0.647	0.581	0.795	0.409	5.263

Table 4.4: **Implication generation performance on rule-based Implication validation dataset.** Note that using the knob mechanism instead of an implied answer gives significant improvement.

We train our implication generator on the rule-based training dataset and evaluate our module on rule-based validation split. We use common question generator metrics such as BLEU [30], ROUGE-L [31], METEOR [32] and CIDEr [33] scores for evaluation. We also demonstrate the importance of using the Knob mechanism instead of an implied answer as input to the module. Table 4.4 shows the results of the implication generator module. Some examples of generated implications by our module are shown in Section 4.8.

4.6 Implementation details

For the gating mechanism and late activation, $T_{sim} = 0.9$ and $A_{iter} = 5500$ for Pythia and $A_{iter} = 10,000$ for BAN and BUTD. The LSTM hidden state size for implication generator module is 1024 and Glove embeddings are used of $dim = 300$. The weights for the losses are kept as $\lambda_Q = 0.5$ and $\lambda_{imp} = 1.5$. All models are trained on training split and evaluated on validation split of VQA v2.0 dataset.

4.7 Examples of Attention Maps

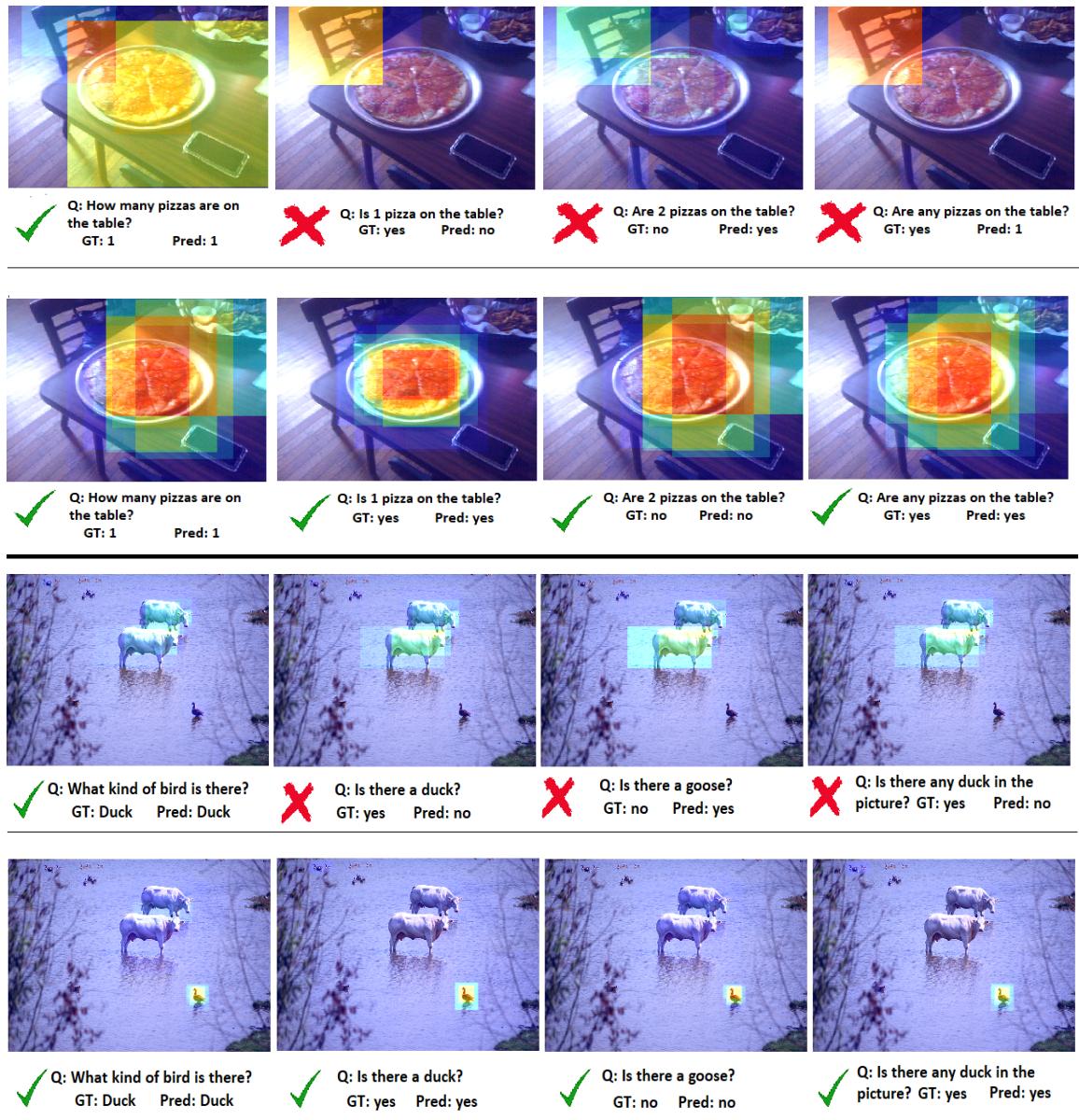




Figure 4.2: Comparison in Attention maps. Top and bottom rows represent Pythia [2] and Pythia trained in our framework respectively.

4.8 Generated Implications by our module



Original	What color is the stop light? A: red
LogEq	Is the stop light red?
Mutex	Is the stop light green?
Nec	Is there anything red in the picture?



Original	How many chairs can be seen? A: 2
LogEq	Can 2 chairs be seen?
Mutex	Can 3 chairs be seen?
Nec	Can any chairs be seen?



Original	What sport are they playing? A: tennis
LogEq	Are they playing tennis?
Mutex	Are they playing basketball?
Nec	Is there a tennis in the picture?



Original	What's on the ground? A: snow
LogEq	's on the ground snow?
Mutex	's on the ground rain?
Nec	Is there snow in the picture?



Original What is the man holding in his hand? **A:** phone

LogEq Is the man holding in his hand phone?

Mutex Is the man holding in his hand set?

Nec Is there a phone in the picture?



Original How many people do you see in this scene? **A:** 0

LogEq Do you see in this photo any people?

Mutex Do you see in this photo 1 person?

Nec Do you see in this photo any people?

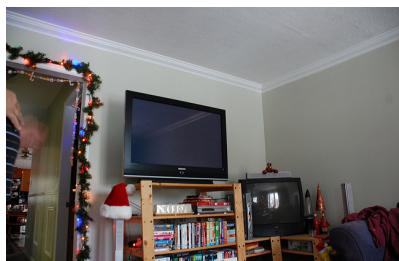


Original How many devices are in the picture? **A:** 5

LogEq Are 3 devices in the picture?

Mutex Are 4 devices in the picture?

Nec Are any devices in this picture?



Original Is this a flat screen TV? **A:** yes

LogEq Is this a natural screen tv?

Mutex Is this a tv screen?

Nec Is this a natural screen tv?

Table 4.5: **Implications generated by our module.** As seen in the examples, the module can replace the answer value in Logical Equivalence type sometimes. Also, for numbered questions having answer '0' and 'yes/no' questions, the module fails to generate correct implications due to limitations of the rule-based dataset.

Chapter 5

Conclusion

In this thesis report, we started off by exploring different state-of-the-art models on Visual Question Answering. During our survey, we realised the monotonous increase in complexity of models. In many cases, VQA models were using language priors to perform well on the training dataset and hence it tends to perform poorly in the wild. We further discussed different approaches in the literature to detect and tackle these problems through new evaluations schemes such as consistency and robustness. Working along similar lines, we designed a system dedicated to improving consistency of VQA models.

Our contributions in this thesis are three fold. First, we propose a model-independent cyclic training scheme for improving consistency of VQA models without degrading their performance. Second, a novel implication generator module for making implications using the question answer pair and a knob mechanism. Third, a new annotated VQA-Implications dataset as an evaluation baseline for future works in consistency.

Our implication generator being trained on rule-based implications dataset, has its own limitations. Firstly, the implications are restricted to 3 types - Logical Equivalence, Necessary Condition and Mutual Exclusion and all implications are limited to 'yes/no' type. We believe that learning on implications not restricted to these limitations should lead to better performance. Furthermore, the rule-based implications come from a fixed distribution and are not as diverse as human annotated implications would be. This limitation can be quantitatively seen by observing the difference between models' performance on rule-based and human annotated implications.

Bibliography

- [1] D. Teney, P. Anderson, X. He, and A. van den Hengel, “Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge,” 2017.
- [2] Yu Jiang, Vivek Natarajan, Xinlei Chen, M. Rohrbach, D. Batra, and D. Parikh, “Pythia v0.1: the Winning Entry to the VQA Challenge 2018,” *arXiv preprint arXiv:1807.09956*, 2018.
- [3] J.-H. Kim, J. Jun, and B.-T. Zhang, “Bilinear Attention Networks,” in *Advances in Neural Information Processing Systems 31*, pp. 1571–1581, 2018.
- [4] H. Tan and M. Bansal, “LXMERT: Learning Cross-Modality Encoder Representations from Transformers,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [5] Q. Li, Q. Tao, S. Joty, J. Cai, and J. Luo, “VQA-E: Explaining, Elaborating, and Enhancing Your Answers for Visual Questions,” 2018.
- [6] J. Wu, Z. Hu, and R. J. Mooney, “Generating Question Relevant Captions to Aid Visual Question Answering,” 2019.
- [7] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende, “Generating Natural Questions About an Image,” 2016.
- [8] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun, “iVQA: Inverse Visual Question Answering,” 2017.
- [9] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [10] V. Manjunatha, N. Saini, and L. S. Davis, “Explicit Bias Discovery in Visual Question Answering Models,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [11] M. T. Ribeiro, C. Guestrin, and S. Singh, “Are Red Roses Red? Evaluating Consistency of Question-Answering Models,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 6174–6184, Association for Computational Linguistics, July 2019.
- [12] R. R. Selvaraju, P. Tendulkar, D. Parikh, E. Horvitz, M. Ribeiro, B. Nushi, and E. Kamar, “SQuINTing at VQA Models: Interrogating VQA Models with Sub-Questions,” 2020.
- [13] T. Gokhale, P. Banerjee, C. Baral, and Y. Yang, “VQA-LOL: Visual Question Answering under the Lens of Logic,” 2020.
- [14] M. Shah, X. Chen, M. Rohrbach, and D. Parikh, “Cycle-Consistency for Robust Visual Question Answering,” in *2019 Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE.
- [15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “VQA: Visual Question Answering,” in *International Conference on Computer Vision (ICCV)*, 2015.
- [16] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding,” 2016.
- [17] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask Your Neurons: A Neural-based Approach to Answering Questions about Images,” 2015.
- [18] M. Ren, R. Kiros, and R. Zemel, “Exploring Models and Data for Image Question Answering,” 2015.
- [19] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, “Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering,” 2015.
- [20] H. Noh, P. H. Seo, and B. Han, “Image Question Answering using Convolutional Neural Network with Dynamic Parameter Prediction,” 2015.

- [21] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical Question-Image Co-Attention for Visual Question Answering,” 2016.
- [22] N. Sundaram, T. Brox, and K. Keutzer, “Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow,” in *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV’10, (Berlin, Heidelberg), p. 438–451, Springer-Verlag, 2010.
- [23] D. Tang, N. Duan, Z. Yan, Z. Zhang, Y. Sun, S. Liu, Y. Lv, and M. Zhou, “Learning to Collaborate for Question Answering and Asking,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 1564–1574, Association for Computational Linguistics, June 2018.
- [24] Y. Yan, W. Qi, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, “ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training,” 2020.
- [25] S. Reddy, D. Raghu, M. M. Khapra, and S. Joshi, “Generating Natural Language Question-Answer Pairs from a Knowledge Graph Using a RNN Based Question Generation Model,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 376–385, Association for Computational Linguistics, Apr. 2017.
- [26] V. Kumar, G. Ramakrishnan, and Y.-F. Li, “Putting the Horse Before the Cart: A Generator-Evaluator Framework for Question Generation from Text,” 2018.
- [27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations,” in *International Conference on Learning Representations*, 2020.
- [28] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [29] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations,” 2016.

- [30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL ’02, (USA), p. 311–318, Association for Computational Linguistics, 2002.
- [31] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.
- [32] M. Denkowski and A. Lavie, “Meteor Universal: Language Specific Translation Evaluation for Any Target Language,” in *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.
- [33] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “CIDEr: Consensus-Based Image Description Evaluation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.